UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Università degli Studi di Modena e Reggio Emilia

Research Doctorate in
Information and Communication Technologies (ICT)

XXXVIII Cycle

Bridging Vision and Language for Open-Vocabulary Segmentation

From Self-Supervised Backbones to Multimodal Reasoning

Luca Barsellotti

Supervisor: Prof. Rita Cucchiara
PhD Programme Coordinator: Prof. Luigi Rovati

Modena, 2026

Review committee composed of:

Fabrizio Falchi, *Consiglio Nazionale delle Ricerche (ISTI-CNR)*
Pascal Mettes, *University of Amsterdam (UvA)*

To Sara

Abstract

Localizing and categorizing objects within an image is one of the fundamental and long-standing challenges in Computer Vision, extensively investigated since the early days of the field in the 1970s. Semantic segmentation tackles this problem by assigning a label to every pixel in the image, thereby partitioning it into coherent regions that correspond to the underlying object categories. However, fully supervised deep-learning approaches rely on costly and time-consuming pixel-level annotations drawn by human annotators. The high effort required to produce such data limits their availability across diverse datasets and domains, restricting the scalability of supervised methods and confining them to a finite set of available categories, thus motivating the need for weaker forms of supervision.

In recent years, the availability of large-scale image–caption datasets and the advent of vision–language models have transformed the field, enabling the joint learning of visual and textual representations. This progress has motivated researchers to extend multimodal understanding to a fine-grained, pixel-level form, giving rise to open-vocabulary semantic segmentation, a paradigm that allows models to segment arbitrary categories described by free-form text queries, thus overcoming the constraints of fixed taxonomies. This capability opens new opportunities in applications such as robotic navigation and manipulation, autonomous driving, augmented and mixed reality, and image editing. Nevertheless, transferring the global supervision provided by image captions into consistent pixel-level multimodal representations remains an open challenge.

In parallel, self-supervised learning has emerged as a paradigm in which models learn rich visual representations from unlabeled data by using intrinsic supervisory signals from the images themselves. This approach has produced general-purpose visual features that can be transferred across tasks and domains without explicit supervision. In particular, Vision Transformer–based models have demonstrated the ability to encode semantics not only globally but also locally, at the level of small image patches.

This dissertation investigates methods to combine the complementary strengths of vision–language and self-supervised models to achieve open-vocabulary segmentation. We propose techniques that leverage localized semantics from self-supervised representations and their integration with text through visual prototypes and contrastive learning, demonstrating that self-supervised features naturally align with linguistic concepts when properly extracted. Building upon these insights, we extend open-vocabulary perception to embodied settings, designing robotic agents capable of object-driven navigation in photorealistic environments through

instance matching and localization guided by both visual and textual cues. Finally, we explore how Multimodal Large Language Models can reason about visual grounding and referring tasks, analyzing existing approaches, tasks, and datasets that connect high-level reasoning with pixel-level localization.

Sommario

La localizzazione e la categorizzazione degli oggetti presenti in un'immagine rappresentano una delle sfide fondamentali e di lunga data della Visione Artificiale, oggetto di studio sin dagli albori della disciplina negli anni Settanta. La segmentazione semantica affronta questo problema assegnando una categoria a ciascun pixel dell'immagine, suddividendola così in regioni coerenti che corrispondono alle categorie degli oggetti coperti. Tuttavia, gli approcci di deep learning completamente supervisionati richiedono annotazioni a livello di pixel, tracciate manualmente da annotatori umani. L'elevato costo e il notevole dispendio di tempo necessari per produrre tali dati ne limitano la disponibilità per domini e dataset diversi, riducendo la scalabilità dei metodi supervisionati, confinandoli a un insieme finito di categorie note e motivando la ricerca di forme di supervisione più deboli.

Negli ultimi anni, la disponibilità di dataset su larga scala costituiti da coppie immagine–didascalia e l'avvento dei modelli linguistici-visivi hanno trasformato il campo, consentendo l'apprendimento congiunto di rappresentazioni visive e testuali. Questo ha portato la comunità scientifica a estendere la comprensione multimodale verso una forma più localizzata, a livello di pixel, dando origine al paradigma della segmentazione semantica open-vocabulary. Tale paradigma permette ai modelli di segmentare categorie arbitrarie descritte testualmente, superando così i vincoli imposti dalle tassonomie predefinite. Questo nuovo approccio apre la strada a numerose applicazioni, come la navigazione e la manipolazione robotica, la guida autonoma, la realtà aumentata e mista e l'editing di immagini. Nonostante ciò, la trasformazione della supervisione globale fornita dalle didascalie in rappresentazioni multimodali coerenti a livello di pixel rimane tuttora un problema irrisolto.

In parallelo, l'apprendimento auto-supervisionato è emerso come paradigma in cui i modelli apprendono ricche rappresentazioni visive a partire da dati non annotati, sfruttando segnali di supervisione intrinseci alle immagini stesse. Questo approccio ha portato allo sviluppo di rappresentazioni visuali universali, che possono essere trasferite per affrontare problemi e domini differenti anche in assenza di una supervisione esplicita. In particolare, i modelli basati su Vision Transformer hanno dimostrato la capacità di codificare informazioni semantiche non solo a livello globale, ma anche localmente, a livello di piccole porzioni dell'immagine chiamate patch.

Questa tesi di dottorato indaga metodi per combinare i punti di forza complementari dei modelli linguistici-visivi e di quelli auto-supervisionati per la

segmentazione open-vocabulary. Vengono proposte tecniche che sfruttano la semantica locale appresa dai modelli auto-supervisionati e la loro integrazione con il linguaggio mediante prototipi visuali e apprendimento contrastivo, dimostrando che tali rappresentazioni sono naturalmente allineate ai concetti linguistici quando estratte opportunamente. Su queste basi, estendiamo la percezione open-vocabulary a sistemi di robotica intelligente, progettando agenti in grado di eseguire attività di navigazione orientata al trovare oggetti in ambienti fotorealistici tramite riconoscimento e localizzazione di istanze guidata da referenze visive e testuali. Infine, esploriamo come i Multimodal Large Language Models possano ragionare su compiti di visual grounding e referring, analizzando approcci, problemi e dataset che collegano il ragionamento ad alto livello con la localizzazione a livello di pixel.

Acknowledgments

First of all, I would like to thank Prof. Marcella Cornia for her invaluable support and mentorship throughout my PhD journey. I am grateful to Prof. Rita Cucchiara and Prof. Lorenzo Baraldi for giving me the opportunity to join such an ambitious, stimulating, and inspiring research lab, filled with talented and outstanding people.

I would like to express my sincere gratitude to my reviewers, Prof. Fabrizio Falchi and Prof. Pascal Mettes, for their time and valuable feedback.

I would like to thank all the colleagues with whom I have had the pleasure of collaborating over these years. First of all, a special thanks goes to Roberto, who guided me through my first steps in the research world. Learning from someone so talented and thoughtful has been invaluable. A special thanks to the *kindergarten*, Davide, Nicholas, Beppe, and Fabio, amazing friends who turned the PhD journey into a continuous celebration of jokes and laughter, brightening every single day. Thanks also to Nello, Roberto, Monica, Silvia, Vitto, Fede Cocchi, and Alle, dear friends and companions of many unforgettable memories; to Davide Morelli, Lollo, Jack, Fede Biagi, Gianluca, and Lele, for the many adventures around the world; and to the new generation, Leo, Davide Bucciarelli, and Evelyn, who inherited our spirit and to whom I wish all the best.

I also thank the entire ISTI-CNR team, Lorenzo, Fabio, Nicola, and Fabrizio. Your contagious passion and friendliness made me feel at home from the very beginning. It has been a true pleasure working together.

I am grateful to the CSSE team at Amazon, my host Luitpold, my supervisor Jochen, and Waleed, Pol, Dominik, Matthias, Satya, Dmitrii, Martijn, Kevin, Tim, Sergey, Yasin, Íñigo, and Manh Long. You welcomed me like family, and those six months were truly wonderful. I learned a great deal from working in such an environment and alongside people with strong values.

I also thank the entire Semantic Perception team at Google, my host, Maxim, and my co-authors, Martin, Mattia, Ferjad, Yongqin, and Nikita, as well as the other interns, Luca, Nando, Yuru, Theo, Haofei, Selim, Orest, and Zhaochong. Being part of such a talented group, constantly pushing the boundaries of current research, has been extremely inspiring.

I would like to thank my closest group of friends. Riccardo, the lifelong friend, we have known each other since we were newborns and have shared everything. Our friendship is one of the most precious things I have. Matteo, whose passion, lightheartedness, and way of seeing the world always make me feel as if we were still classmates, and who continuously inspires the most idealistic part of me.

Leonardo, whose deep sense of friendship makes him someone I can always rely on, and whose passion for his work is truly inspiring. Massimiliano, whose strong ideals and principles make him a point of reference in our lives. And Gianluca, for the countless hours spent talking about our lives, and for always being our *newbie*.

I am deeply grateful to my parents, Ombretta and Claudio, for always supporting my choices, for always listening to me, and for their genuine curiosity and interest in what I was doing, making every effort to understand my world and offer me the best possible advice. Thank you to my brother Paolo for always trying to make me laugh and for the unconditional affection he has always shown me.

And finally, my most important thanks go to my girlfriend, Sara, who has always believed in me and supported me, even in the most difficult moments. Your love and your presence have been the foundation of this journey, and the memories we have built together make looking back on these years truly magical.

Contents

Chapter 1

Introduction

In the last decades, the industry has witnessed a profound transformation, often referred to as the *Fourth Industrial Revolution*. This new era is characterized by advanced forms of automation in industrial processes, enabled by smart devices and robotic agents empowered with Artificial Intelligence technologies. The main driving forces behind this revolution are the exponential growth in the availability of data and the continuous increase in computational power. Technological progress has therefore focused on the development of models capable of learning complex patterns directly from raw information, processing massive amounts of data through high-performance hardware such as GPUs and TPUs.

Among the various types of data that have fueled this transformation, visual data, as images and videos, have played a particularly central role. The widespread presence of cameras in everyday life, now even present in all our pockets through smartphones, along with their decreasing cost and high quality, has made visual information extremely abundant and easy to collect. This unprecedented accessibility, combined with the inherent richness of visual content in describing the world around us, has established vision as one of the most influential and challenging domains within Artificial Intelligence research.

Computer Vision. The branch of Artificial Intelligence dedicated to the interpretation of visual information is Computer Vision. Its goal is to develop systems capable of perceiving, understanding, and reasoning about the visual world in a manner comparable to human perception. By extracting meaningful representations from pixels, Computer Vision algorithms aim to recognize and localize objects, interpret scenes, and ultimately connect visual perception with higher-

level reasoning and decision-making. These capabilities pave the way to a wide range of applications, from autonomous driving and robotics to medical imaging, augmented reality, and industrial automation.

Image Segmentation. Among the various problems studied in this field, localizing and categorizing objects within an image represents one of its most fundamental and long-standing challenges. The ability to identify *what* is present in a scene and *where* it is located lies at the core of visual understanding. Historically, this challenge has been addressed through image segmentation, the process of partitioning an image into regions that are internally coherent and distinct from their surroundings. In its early stages, dating back to the 1970s and 1980s, segmentation research primarily focused on low-level vision, relying on geometric and photometric cues such as edges, color, and texture to separate image regions. These approaches, while capable of delineating boundaries, lacked semantic understanding: they could group pixels based on similarity but could not determine what those regions represented.

The advent of deep learning radically transformed this paradigm. The availability of large annotated datasets and the introduction of convolutional neural networks, as well as more recent Transformer-based architectures, have enabled the learning of hierarchical representations that connect visual structure with semantic meaning. In this context, semantic segmentation emerged as a key task, extending traditional segmentation to the assignment of a class label to each pixel in the image. This shift from structural to semantic interpretation marked a turning point in Computer Vision, allowing models to parse complex scenes and reason about objects and their relationships.

Supervised Learning. Fully supervised learning has represented the dominant paradigm in Computer Vision for several years, particularly in tasks such as image classification, object detection, and semantic segmentation. In this setting, a model is trained on a large dataset of images paired with human-provided annotations that explicitly define the desired output for each input example. For semantic segmentation, this supervision takes the form of pixel-level masks, where each pixel is assigned to a label corresponding to one of a predefined set of object categories. Through this process, the model learns to associate visual patterns in the input image with their corresponding semantic meanings. A model is said to *generalize* if it maintains its ability to recognize the same categories when evaluated on images from domains or distributions that differ from those used for training.

While this approach has achieved remarkable success, it suffers from fun-

damental limitations related to its dependence on extensive manual annotation. Creating pixel-accurate segmentation masks is an extremely time-consuming and costly process, as it requires human annotators to delineate object boundaries and assign labels consistently across large image collections. As a result, annotated datasets tend to cover only a limited number of domains and object categories, typically those that are most common or easiest to label. Consequently, fully supervised models are constrained to recognize only the categories present in their training data, lacking the ability to be extended to novel or fine-grained concepts that were not explicitly annotated. This dependency on exhaustive human supervision poses significant challenges for scalability and adaptability. The visual world is inherently open-ended, containing a vast, long-tailed collection of concepts that vary in appearance, scale, and attributes. Extending supervised models to handle these cases would require continuously expanding and relabeling datasets, which is impractical at a large scale.

Vision-Language Models. The availability of large-scale web-crawled datasets containing images paired with natural language captions has paved the way for a new generation of vision–language models, such as CLIP. These models learn joint visual and textual representations by aligning features extracted from images and their corresponding captions within a shared embedding space. This multimodal alignment allows the model to associate visual concepts with their linguistic descriptions, effectively bridging the gap between perception and language. A key consequence of this learning paradigm is the emergence of the zero-shot classification capability. In this setting, a model trained with supervision on a set of labeled categories, also referred to as *seen* categories, can be extended to *unseen* ones by simply providing the new category names as textual prompts. The model maps visual to textual embeddings and selects the prompt that best matches the image content, enabling recognition beyond the fixed label space defined during training. This approach eliminates the need for explicit retraining on new categories and has demonstrated impressive generalization abilities across diverse datasets.

Open-Vocabulary Segmentation. Building upon its success, the open-vocabulary paradigm extends the idea of zero-shot recognition by removing the distinction between seen and unseen categories altogether. Instead of operating within a closed set of predefined labels, open-vocabulary models aim to understand and classify any arbitrary textual concept provided by the user. This paradigm shift has naturally inspired researchers to investigate whether such flexibility could be transferred from image-level recognition to pixel-level understanding for the

semantic segmentation task, where the dependence on dense annotations makes supervised learning far more constrained than in classification. Open-vocabulary methods thus represent a promising direction to relax these constraints and enable segmentation models to generalize beyond the fixed set of annotated categories. Achieving open-vocabulary segmentation requires establishing multimodal correspondences between pixels and text. However, while large-scale image–caption datasets provide alignment between global image representations and language, they do not offer supervision at the pixel level. The central challenge, therefore, lies in transferring the multimodal knowledge learned from image-level pairs to fine-grained pixel-level representations.

Self-Supervised Learning. In parallel with the rise of vision–language models, self-supervised learning has emerged as a powerful paradigm for visual representation learning. In this setting, models are trained on large collections of unlabeled images by exploiting intrinsic supervisory signals derived from the data itself, rather than relying on human annotations. By solving automatically defined pretraining objectives, self-supervised models learn semantically rich and transferable visual representations. These features can be effectively reused across a wide range of downstream tasks with minimal additional supervision.

The advent of Vision Transformer (ViT) architectures has further expanded the potential of self-supervised learning. By representing images as sequences of patch tokens processed through self-attention, Vision Transformers enable the modeling of long-range dependencies and interactions between image regions. Recent self-supervised approaches built upon this architecture have shown that such representations capture semantic structure not only at the global image level but also locally, at the level of individual patches. Among these approaches, the DINO series of models has demonstrated remarkable semantic richness at the patch level. Through a self-distillation strategy that aligns representations across multiple augmented views of the same image, DINO learns highly structured visual embeddings without requiring labeled data. These representations have proven to be strong general-purpose features, forming an effective foundation for dense prediction tasks such as semantic segmentation and depth estimation.

1.1 Thesis Overview

The central objective of this dissertation is to investigate how to bridge self-supervised visual representations with language understanding, and to explore their integration and application in open-vocabulary perception and related multimodal

tasks. The underlying intuition motivating this research is that self-supervised models, trained without human annotations, learn representations that encode a rich and structured understanding of visual scenes, which are potentially alignable with linguistic semantics. However, this alignment is not explicitly established during training and therefore remains hidden within the learned feature space. The core hypothesis of this work is that this latent connection between visual and textual semantics can be revealed and formalized through appropriate mechanisms that associate the visual embeddings produced by self-supervised backbones with representations derived from language.

This perspective envisions the development of universal visual models that preserve the generality, scalability, and transferability of self-supervised features while also integrating the capacity to interpret and reason about visual content through natural language. Such models could serve as a foundation for a broad spectrum of downstream tasks, spanning geometric and structural tasks, such as depth estimation and 3D reconstruction, and semantically grounded tasks, such as segmentation or object recognition. The combination of these two dimensions, geometric understanding and semantic reasoning, is fundamental for real-world applications in autonomous driving, robotic navigation, and industrial automation, where systems must simultaneously comprehend the physical structure of the environment and the meaning of the objects within it. By embedding textual understanding into general-purpose visual representations, this dissertation aims to move toward perceptual systems that are both semantically aware and universally applicable across domains and modalities.

Chapter 2 provides the foundational background for the rest of the thesis. It begins with an overview of traditional image segmentation methods, and evolves toward deep learning approaches such as convolutional neural networks and attention-based architectures. The Chapter then introduces the open-vocabulary paradigm and the motivation for integrating vision-language models into pixel-level perception. Finally, it discusses the growing field of object-oriented navigation and embodied AI, positioning these tasks as real-world benchmarks for evaluating the interplay between visual understanding and language grounding.

Chapter 3 investigates how to connect self-supervised visual representations with textual semantics by introducing the concept of *visual prototypes*. A visual prototype represents a category or concept through the average of patch-level embeddings produced by a self-supervised model over localized regions of an image. If such localized regions can be associated with textual descriptions, these prototypes naturally provide a bridge between visual and linguistic spaces. We investigate different strategies to obtain this association between visual localiza-

tion and textual concepts. One direction leverages image–caption datasets, where captions provide weak supervision about the objects and entities present in the scene, and existing open-vocabulary models can be used to estimate the spatial regions corresponding to the mentioned terms. A complementary approach exploits the properties of diffusion-based image generation models, which are capable of producing synthetic images conditioned on textual prompts while revealing the image regions linked to each word in the conditioning caption. These two sources of supervision enable the construction of text-aligned visual prototypes that bridge the gap between self-supervised features and linguistic representations.

Chapter 4 explores a contrastive approach to align self-supervised visual backbones with text encoders for open-vocabulary segmentation. The core idea is to learn a lightweight projection layer that connects patch-level embeddings from a self-supervised model with the corresponding textual representations, using training on large-scale image–caption datasets. Unlike traditional contrastive methods that align a global image representation with its associated caption, this work exploits an inherent property of self-supervised models that allows them to identify coherent and semantically consistent regions within the image. By leveraging this property, the alignment focuses only on the image regions that are relevant to the textual description, thereby establishing a fine-grained correspondence between visual and linguistic features.

In Chapter 5 we introduce a novel navigation task in which a robotic agent must locate a specific object within an unexplored environment, given either textual descriptions or visual references of the object. To support this task, we present a new dataset specifically designed for these conditions, in which target objects may appear among distractors or within multiple locations. We evaluate how state-of-the-art open-vocabulary detectors and self-supervised visual backbones can be leveraged to identify the target object, integrate visual and textual cues, and guide the navigation policy.

Chapter 6 focuses on the recent evolution of Multimodal Large Language Models (MLLMs) and how these models can be equipped with fine-grained visual understanding. It provides a structured analysis of the techniques that enable MLLMs to reason about images at the pixel level. In particular, we review and categorize existing approaches that integrate visual grounding capabilities, where the model predicts the image region corresponding to a textual reference, and referring capabilities, where the user specifies the region as input for description or reasoning, focusing on how the integration of self-supervised models enhances the local understanding. We further examine the training data required for such grounding abilities and discuss the evaluation protocols and datasets commonly adopted

in the literature. By analyzing these methods and their associated benchmarks, this chapter outlines the emerging directions for extending the reasoning abilities of MLLMs toward pixel-level perception, bridging the gap between high-level language understanding and detailed visual localization.

1.1.1 Activities carried out during the Ph.D.

In addition to the research activities presented in this thesis, contributions to academic service and participation to internships, conferences, scientific schools, and seminars are listed below. The complete list of publications is instead reported in Appendix 8.

Internships

- Google, Zurich, Switzerland (Nov. 2025 - March. 2026): Conducted research on real-time open-vocabulary instance segmentation on edge devices in the Semantic Perception Team, under the supervision of Dr. Maxim Berman
- Amazon, Berlin, Germany (Apr. 2025 - Sep. 2025): Conducted research on cross-model alignment for product image retrieval in the Content Systems Science and Engineering Team, under the supervision of Dr. Jochen Gast and Dr. Luitpold Staudigl

Conferences and schools attended

- International Summer School on Machine Vision (VISMAC) Padova, Italy, 2023
- ELLIS Summer School on Large-Scale AI for Research and Industry, Modena, Italy, 2023
- IEEE/CVF Computer Vision and Pattern Recognition (CVPR), Seattle, Washington (USA), 2024
- International Computer Vision Summer School (ICVSS), Scicli, Italy, 2024
- European Conference on Computer Vision (ECCV), Milan, Italy, 2024
- Conference on Neural Information Processing Systems (NeurIPS), Vancouver, Canada, 2024

- IEEE/CVF International Conference on Pattern Recognition (ICCV) Honolulu, Hawaii (USA), 2025

Seminars and Workshops

- "*Digital Humanities and Artificial Intelligence for humans in today society*", Prof. Rita Cucchiara, 2022
- "*Graph Signal Processing for Machine Learning: Challenges and Usecases*", Prof. Laura Toni, 2022
- "*From Handcrafted to End-to-End Learning, and Back: a Journey far Multi-Object Tracking*", Prof. Laura Leal-Taixé. 2022
- "*3D Computer Vision for animals*", Prof. Silvia Zuffi, 2022
- "*The Unreasonable Effectiveness of Large Language-Vision Models for Video Domain Adaptation*", Prof. Elisa Ricci, 2023
- Terzo Meeting Annuale del progetto Fit4MedRob, 2025
- ICCV Doctoral Consortium, mentored by Prof. Cordelia Schmid, 2025

Participation to Research Projects

- ELIAS - *European Lighthouse of AI for Sustainability*, Horizon Europe Research & Innovation Programme
- PERSEO - *Personalized Robotics as Service Oriented Applications*, Marie Sklodowska-Curie Action Horizon
- Fit4MedRob - *Fit for Medical Robotics*, PNRR
- FAIR - *Future Artificial Intelligence Research*, PNRR

Teaching Activities

- Lecturer for the "*Intensive Master: AI and ML for Smart Factory*", Experis Srl

Journals Reviewing

- IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)

- Pattern Recognition
- ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)
- AI Communications

Conferences Reviewing

- IEEE/CVF Computer Vision and Pattern Recognition (CVPR, Outstanding Reviewer in 2025)
- IEEE/CVF International Conference on Computer Vision (ICCV)
- European Conference on Computer Vision (ECCV)
- The British Machine Vision Conference (BMVC)
- IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
- International Conference on Image Analysis and Processing (ICIAP)

Chapter 2

Background

2.1 Traditional Image Segmentation

Before the advent of deep learning, image segmentation was studied as a low-level vision task grounded in the principles of human perception and mathematical modeling. Early research aimed to formalize the organization of visual elements into coherent structures, drawing inspiration from Gestalt psychology and modeling this concept through region-based, boundary-based, and graph-theoretical formulations. These classical approaches treated segmentation as the problem of partitioning an image into internally consistent regions using handcrafted criteria such as intensity, texture, or edge continuity. Over time, they evolved into more structured formulations through graph partitioning, probabilistic graphical models, and superpixel algorithms. Although these methods lacked semantic understanding, they laid the groundwork for modern segmentation pipelines by defining the core principles of spatial coherence, region homogeneity, and perceptual grouping that continue to underpin current approaches. This Section reviews the foundational methods that preceded the learning-based era, tracing the development of segmentation from early perceptual models to structured optimization techniques.

2.1.1 From Gestalt Psychology to Image Segmentation

In the early years of Computer Vision, the formulation of image segmentation was strongly influenced by Gestalt psychology, a school of thought that studies how the human visual system organizes visual stimuli into coherent perceptual

units named *Gestalten*. According to Gestalt principles, visual perception is governed by grouping rules such as similarity, proximity, continuity, and closure, which determine how individual elements are naturally aggregated into meaningful wholes [155, 156, 294, 347, 348]. Early segmentation methods can be understood as computational attempts to formalize these perceptual principles, translating them into an algorithmic low-level perceptual grouping problem, defined as the task of partitioning an image into a set of disjoint regions such that pixels within the same regions are homogeneous according to low-level visual criteria, such as intensity, color, or texture, while pixels in different regions are dissimilar. Image segmentation was often explicitly framed as a clustering problem, where the goal was to identify coherent groups of elements based on similarity relationships [238, 268]. As noted by Haralick and Shapiro [120], the key distinction between clustering and image segmentation lies in the fact that clustering operates in an abstract measurement space, while segmentation must simultaneously respect the spatial organization of the image, enforcing mutually exclusive regions in the image domain alongside similarity in feature space.

A seminal contribution in this direction was introduced by Zahn [395], who proposed one of the first graph-theoretical formulations of perceptual grouping inspired directly by Gestalt psychology. In this framework, image elements are modeled as vertices in a graph with edge weights encoding pairwise proximity based on inter-point distances. Gestalt clusters are then detected by constructing the minimum spanning tree (MST) of the graph and cutting edges with high weights, corresponding to dissimilar connections between elements, with the remaining connected components of the MST defining the segmentation.

Another influential line of early work approached image segmentation through region-based recursive decomposition and merging. In particular, Horowitz and Pavlidis [127, 240] proposed a segmentation method based on a split-and-merge strategy, in which the image is initially considered as a single region and recursively subdivided into smaller regions according to predefined homogeneity criteria, such as intensity variance or texture measures. Once splitting is complete, adjacent regions that satisfy similarity constraints are iteratively merged, yielding a final partition of the image. A closely related approach was later introduced by Ohlander *et al.* [232], who proposed a segmentation method based on recursive region splitting within a multiscale framework. In their formulation, the image is progressively decomposed into regions of increasing granularity, yielding a hierarchical representation in which large, coarse regions capture global image structure, while smaller regions encode finer local details.

Together, graph-theoretical clustering methods and region-based approaches

established the foundational paradigms of image segmentation, influencing the research over subsequent decades. In particular, they laid the conceptual and methodological groundwork for later developments in graph-based optimization, spectral clustering, superpixel algorithms, and, ultimately, learning-based segmentation methods, including those in the deep learning era.

2.1.2 Boundary-Based and Thresholding Approaches

In parallel to region-based and graph-theoretical formulations, early research on image segmentation also explored boundary-based approaches, which framed segmentation as the problem of identifying discontinuities in image intensity. Classical edge detectors, such as the Roberts [265], Prewitt [244], and Sobel [293] operators, estimate local image gradients by convolving the image with small predefined kernels that approximate first-order spatial derivatives. Pixels exhibiting high gradient magnitude are then interpreted as candidate boundaries, under the assumption that object contours correspond to abrupt intensity changes.

Building upon these early gradient-based methods, Canny [33] introduced a more principled approach to edge detection by formulating it as a multi-stage optimization problem. The Canny edge detector was designed to satisfy three criteria: high detection accuracy, precise localization of edges, and robustness to noise. To this end, the method integrates image smoothing, gradient computation, non-maximum suppression, and hysteresis thresholding within a unified framework, producing thin and well-localized edge maps.

Complementary to boundary-based techniques, thresholding approaches addressed segmentation by partitioning images according to pixel intensity values. In these methods, pixels are assigned to different regions based on whether they fall above or below a selected threshold [144, 264, 269, 272]. However, determining an appropriate threshold is a critical challenge, particularly in the presence of noise or varying illumination [120, 238, 278]. Otsu [235] proposed a widely used solution by introducing an automatic threshold selection method that maximizes the separation between foreground and background classes, measured in terms of inter-class variance. While effective in constrained settings, thresholding-based methods cannot generally capture complex spatial structures, motivating the development of more global and context-aware segmentation techniques.

2.1.3 Graph-based Segmentation

Building upon the graph-theoretical perspective on image segmentation originally introduced by Zahn [395], Wu and Leahy [355] revisited segmentation as a graph partitioning problem, where the image is represented as a weighted undirected graph $G = (V, E)$ whose vertices correspond to pixels and whose edges connect neighboring pixels and encode pairwise affinities derived from a given measure of similarity. The objective of the problem is to partition the vertices into the set $V_1, V_2, \ldots, V_m$ such that the measure of similarity is high between the vertices of the set V_i, while it is low among the vertices across the sets V_i and V_j. The graph G can be partitioned into two disjoint sets A and B such that $A \cup B = V$ and $A \cap B = \varnothing$ by removing the edges that connect the two sets. The degree of dissimilarity between the two sets can be computed as the total weight of the removed edges. This value, in graph theory, is called a cut and is defined as:

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v). \tag{2.1}$$

The optimal bipartitioning is the one that minimizes the value of the cut. When partitioning the graph into k-subgraphs, the objective is to minimize the total cut cost between partitions, and this can be solved by recursively computing the minimum cut that bisects each partition. This formulation, with respect to the local grouping rules introduced by Zahn, exploits a global measurement of the quality of the segmentation, thereby enforcing the overall coherence by discouraging the separation of strongly connected image elements. However, the minimum cut objective favors the formation of small sets of vertices because the cut grows with the number of edges in such partitions.

To avoid the bias toward partitioning small sets of vertices, Shi and Malik [283, 284] introduced a novel formulation of the degree of dissimilarity between partitions named normalized cut (NCut) and defined as

$$\text{NCut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \tag{2.2}$$

where $\text{assoc}(A, V) = \sum_{u \in A, v \in V} w(u, v)$ is the total number of edges connecting vertices from A to all vertices in the graph. The normalized cut can be reformulated as the Rayleigh quotient [109], and, assuming that the optimal cut can assume real values, it can be minimized by solving a generalized eigenvalue system. Starting from the second smallest eigenvector, each next eigenvector corresponds to the subsequent real-valued partitioning, which can be approximated to a discrete cut.

From this perspective, this approach can also be interpreted as a form of spectral clustering, where the eigenvectors of the graph Laplacian provide an embedding that reveals the intrinsic grouping structure of the image graph [45, 64, 89, 103]. While the real values relaxation yields multiple eigenvectors corresponding to low-cost cuts, in practice, the discretization of high-dimensional embeddings is non-trivial and often unstable. Hence, the normalized cut algorithm is typically applied through recursive bipartitioning, recomputing the generalized eigenvalue problem at each stage while considering the second smallest eigenvector as the current cut, to finally obtain k-subgraphs.

2.1.4 Probabilistic Graphical Models

Alongside graph-based and region-based segmentation methods, probabilistic graphical models played a significant role in the evolution of image segmentation. Markov Random Fields (MRFs) [26, 82, 83] and Conditional Random Fields (CRFs) [158] model segmentation as a structured prediction problem, where pixels or regions are treated as random variables whose labels are jointly inferred by minimizing an energy function that balances data fidelity with spatial coherence. In this formulation, unary terms encode local appearance information, while pairwise terms enforce smoothness and contextual consistency between neighboring elements. CRFs, in particular, enabled the incorporation of rich data-dependent potentials and were widely adopted to refine segmentation outputs, including as post-processing stages in early deep learning pipelines.

2.1.5 Superpixel Algorithms

The modern notion of superpixels originates from the observation that individual pixels are not natural or meaningful entities for representing visual scenes, but rather artifacts of the discrete sampling process used in image representation. A seminal formulation of this idea was introduced by Ren and Malik [262], who coined the term superpixel to indicate a visual unit composed of multiple pixels that maintains locality, coherence, and the necessary structure for segmentation at the desired scale. Importantly, superpixels are class-agnostic and do not carry semantic meaning; instead, they serve as perceptual primitives that reduce the complexity of the segmentation problem [24, 34]. In their proposed approach, Ren and Malik adopted superpixels as an oversegmentation preprocessing stage by applying the Normalized Cuts algorithm using contour and texture cues as measures of similarity. Then, the goodness of the current segments, initialized

with the superpixels and updated with a random search, is predicted with a trainable classifier on top of Gestalt cues, such as texture, brightness, contour, and curvilinear continuity.

Superpixel algorithms have been categorized according to their high-level formulation by Achanta *et al.* [1, 2] and Stutz *et al.* [297]. Graph-based approaches can be seen as a natural evolution of the Normalized Cuts framework and are commonly divided into optimal-cutting [262, 325], bottom-up merging [102, 132, 195], and elimination-based [413] methods. In particular, Felzenswalb and Hutterlocher [102] proposed a pioneering bottom-up merging approach in which an agglomerative clustering of pixels is performed such that each superpixel is the minimum spanning tree of the constituent pixels.

Watershed-based approaches interpret an image as a topographic surface, where pixel intensities or gradient magnitudes define elevation. In this analogy, segmentation is obtained by identifying catchment basins associated with local minima and the watershed lines that separate them [219, 326]. Variants of this paradigm primarily differ in how the image is preprocessed and how markers are defined to guide the flooding process, enabling control over the number, size, and compactness of the resulting superpixels [18, 129, 210, 211, 230].

Clustering-based superpixel algorithms are inspired by classical clustering techniques and generate superpixels by grouping pixels according to a measure of similarity [185, 230, 239, 331, 346]. Among them, the Simple Linear Iterative Clustering (SLIC) algorithm [1, 2] has emerged as a widely adopted baseline, which adapts k-means for superpixel generation by restricting the clustering to local neighborhoods, and employs color similarity and spatial proximity as measures.

Energy optimization–based superpixel algorithms formulate superpixel generation as the iterative minimization of an objective function that encodes desired region properties, such as color homogeneity and spatial regularity [69, 218, 306, 307, 380]. A representative example is the SEEDS algorithm [319], which starts with a regular grid of initial superpixels and iteratively refines their boundaries through a hill-climbing optimization process. The energy function is defined as the sum of the likelihood of the colors and a prior of the boundary shapes of the superpixels.

Beyond the main families discussed above, a variety of additional strategies for superpixel generation have been explored. These include density-based methods based on mode-seeking in feature space [68, 323], contour evolution approaches that represent superpixels as deformable regions [29, 30, 172], path-based formulations that connect seed points through pixel paths [75, 93], as well as other

signal-processing–driven strategies [295].

2.2 Segmentation in the Era of Deep Learning

The advent of deep learning marked a paradigm shift in image segmentation, moving from handcrafted heuristics to learned representations driven by data. Fueled by large annotated datasets, powerful hardware, and scalable optimization, convolutional neural networks demonstrated unprecedented performance in extracting semantic meaning from raw pixels, enabling dense prediction tasks such as semantic, instance, and panoptic segmentation. This evolution continued with the introduction of attention-based architectures, culminating in Vision Transformers that brought long-range contextual reasoning to pixel-level understanding. In parallel, object-centric and query-based models reframed segmentation as a set prediction problem, unifying semantic and instance segmentation under a common architectural framework. This chapter surveys the foundational deep learning architectures that have shaped modern segmentation pipelines, ranging from fully convolutional networks to Transformer-based models, and highlights the transition toward object-aware methods.

2.2.1 Convolutional Neural Networks

The field of Computer Vision underwent a profound transformation with the advent of deep learning, driven by the availability of large-scale labeled datasets [81, 190], advances in optimization techniques [229, 134, 151], and the increasing accessibility of high-performance computational resources [67, 251]. A pivotal moment in this transition was the introduction of the ImageNet dataset [81], which enabled training and evaluating models on millions of labeled images spanning thousands of object categories. Leveraging this resource, AlexNet [162] demonstrated that deep convolutional neural networks could dramatically outperform traditional hand-engineered pipelines in large-scale image classification, marking the beginning of the modern deep learning era in vision. Subsequent architectures [125, 292, 302] further refined this paradigm, showing that deep networks are capable of learning hierarchical and increasingly abstract feature representations that encode high-level semantic information directly from raw pixel data [19, 168, 385, 397].

Building upon the success of deep learning in image classification, early attempts [66, 111] to apply neural networks to fine-grained scene understanding

tasks soon emerged. Pioneering works [100, 101, 223] demonstrated that deep architectures could be extended beyond image-level prediction to dense labeling tasks, highlighting the potential of learned hierarchical features for capturing semantic information at the pixel level. Moreover, several human-annotated datasets were introduced across diverse domains, including urban driving scenes, indoor environments, natural images, and medical imagery, providing pixel-level annotations for fixed sets of categories. These datasets established a common evaluation setting and enabled the training of models to recognize and localize predefined categories at the pixel level. With these advancements, the focus of image segmentation shifted from purely perceptual grouping toward tasks that explicitly aim to assign semantic meaning to image regions, giving rise to three closely related yet distinct problem formulations: semantic segmentation, instance segmentation [123], and panoptic segmentation [152]. Semantic segmentation assigns a semantic class label to every pixel in the image, grouping all pixels that belong to the same object category. Instance segmentation further refines this formulation by distinguishing between different instances of the same category, producing separate masks for each object occurrence. Panoptic segmentation unifies these two perspectives, assigning every pixel both a semantic label and, when applicable, an instance identifier. In this dissertation, we focus primarily on semantic segmentation, as it provides a natural foundation for pixel-level semantic understanding and serves as the basis for the open-vocabulary setting explored in the following chapters.

A fundamental challenge in adapting deep learning models to semantic segmentation lies in combining high-level semantic abstraction with the preservation of fine-grained spatial detail required for pixel-level prediction [121, 204]. Convolutional neural networks designed for image classification progressively reduce spatial resolution through pooling and strided convolutions to increase receptive field size and semantic expressiveness [125, 292, 302]. While effective for global recognition, this design produces coarse feature maps that lack precise localization [52, 388], making it challenging to model long-range contextual relationships across distant image regions without reducing the spatial resolution [200, 415].

To address these challenges, a range of architectural strategies emerged to adapt deep networks to dense prediction tasks. Fully Convolutional Networks (FCNs) reformulated classification models into fully convolutional architectures capable of producing pixel-wise predictions in an end-to-end manner [204]. Building upon this paradigm, encoder–decoder architectures introduced symmetric decoding stages that progressively recover spatial resolution [10, 44, 231, 267]. These methods leverage lateral skip connections to fuse low-level spatial details from the

encoder with high-level semantic features in the decoder, enabling precise boundary localization. This design is best exemplified by the U-Net architecture [267], which concatenates feature maps from the contracting path directly to the expanding path, and SegNet [10], which reuses pooling indices. In parallel, pyramidal and multi-scale designs explored the aggregation of contextual information across multiple resolutions to improve robustness to scale variations [124, 189, 188, 415], while dilated (or atrous) convolutions enabled the enlargement of the effective receptive field without additional downsampling [52, 389].

2.2.2 Vision Transformer

Another major paradigm shift in Artificial Intelligence was driven by the introduction of the Transformer architecture and the attention mechanism [322], which initially revolutionized the field of Natural Language Processing by enabling models to capture long-range dependencies and global contextual relationships in a flexible and scalable manner. The success of Transformers in language modeling and sequence-to-sequence tasks soon motivated their adoption beyond NLP, leading to a broader impact across multiple AI domains, including Computer Vision. Early attempts to transfer the Transformer paradigm to visual tasks focused on hybrid architectures that combined convolutional neural networks with attention mechanisms, leveraging convolutions for local feature extraction while using self-attention to model global context [131, 182, 338, 394]. These approaches aimed to mitigate the limitations of purely convolutional designs, particularly their difficulty in capturing long-range interactions without aggressive spatial downsampling [408, 407].

A decisive step toward fully attention-based visual models was taken with the introduction of the Vision Transformer (ViT) [90], which demonstrated that a pure Transformer architecture could be successfully applied to image classification when trained on sufficiently large datasets. Let $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ denote an input image with height H, width W, and C channels. The input image is first divided into a sequence of non-overlapping patches of size P, yielding $N = \frac{HW}{P^2}$ patches. Each patch is flattened into a vector $\mathbf{x}_i \in \mathbb{R}^{P^2C},\ i = 1, \ldots, N$ and linearly projected into a D-dimensional embedding space $\mathbf{z}_i = \mathbf{x}_i\mathbf{E},\ \mathbf{E} \in \mathbb{R}^{(P^2C) \times D}$. These patch embeddings serve as the visual tokens processed by the Transformer, analogous to word embeddings in NLP models. Since Transformers are inherently permutation-invariant, positional information is injected by adding a positional embedding as $\mathbf{Z}_0 \leftarrow \mathbf{Z}_0 + \mathbf{P}$, where $\mathbf{P} \in \mathbb{R}^{(N+1) \times D}$ is a learnable or fixed positional encoding, allowing the model to maintain spatial awareness. Additionally, a spe-

cial learnable token $\mathbf{z}_{\text{cls}} \in \mathbb{R}^D$, commonly referred to as the classification token, is prepended to the sequence and is designed to aggregate global information from all patches using the attention mechanism. The output representation corresponding to this token is then used for image-level prediction tasks. The token sequence is processed by a stack of L Transformer encoder layers. Each layer consists of a multi-head self-attention (MSA) block followed by a position-wise feed-forward network (FFN), both equipped with residual connections and layer normalization:

$$\mathbf{Z}'_\ell = \text{MSA}(\text{LN}(\mathbf{Z}_{\ell-1})) + \mathbf{Z}_{\ell-1},$$
$$\mathbf{Z}_\ell = \text{FFN}(\text{LN}(\mathbf{Z}'_\ell)) + \mathbf{Z}'_\ell,$$

for $\ell = 1, \ldots, L$. Given an input $\mathbf{Z} \in \mathbb{R}^{(N+1)\times D}$, self-attention is computed by projecting the tokens into queries, keys, and values:

$$\mathbf{Q} = \mathbf{Z}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{Z}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{Z}\mathbf{W}_V,$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D\times d}$. The attention operation is defined as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}.$$

In the multi-head formulation, this operation is performed independently across h heads, whose outputs are concatenated and linearly projected:

$$\text{MSA}(\mathbf{Z}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_O.$$

Each Transformer layer includes a position-wise feed-forward network defined as

$$\text{FFN}(\mathbf{z}) = \sigma(\mathbf{z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2,$$

where $\sigma(\cdot)$ denotes a non-linear activation function.

Following the release of the ViT, a research direction started investigating how to utilize its architecture for segmentation focusing on decoding the patch embeddings into semantic masks. SETR [418] pioneered this direction by treating the ViT purely as an encoder, reshaping back the patch embeddings into a 2D feature map and feeding them into a standard convolutional decoder to predict pixel-wise class labels. To better recover multi-scale information, DPT [256] introduced a mechanism to reassemble tokens from intermediate layers into a feature pyramid, while ViT-Adapter [57] proposed an external adapter module to explicitly inject local spatial priors into the Transformer backbone. Despite these advancements, the quadratic complexity and fixed resolution of standard ViTs remained a bottleneck, motivating the subsequent shift toward hierarchical Transformer designs and query-based paradigms.

2.2.3 Object-Centric Segmentation

In parallel to the introduction of the ViT, Carion *et al.* [36] introduced a fundamental attention-based paradigm shift toward object-centric methods for object detection, which lately has been adopted in semantic segmentation and is still nowadays representing the reference paradigm [58, 59, 135, 175, 296, 330]. In their proposed method, DETR, the feature map computed from a convolutional network is provided to a Transformer encoder-decoder architecture. The encoder performs self-attention on the flattened convolutional features to capture long-range contextual information, while the decoder employs cross-attention between a set of learnable object queries and the output of the encoder. Finally, the bounding boxes are directly regressed from the output object embeddings using a FFN.

Segmenter [296] adapts the framework of DETR to semantic segmentation in a fully attention-based architecture. It employs a ViT as backbone and replaces the decoder with a Mask Transformer, in which a set of learnable class embeddings is concatenated to the output patch embeddings of the ViT before performing self-attention. Finally, the dot product between the class embeddings and the patch embeddings is computed and upsampled to obtain the resulting masks.

To overcome the structural limitations of the standard ViT, specifically its quadratic complexity and single-scale output, a parallel research direction focused on reintroducing the hierarchical feature pyramid characteristic of CNNs into Transformer-based backbones [110, 203, 350, 361, 369]. These approaches aimed to reconcile the global modeling capabilities of self-attention with the multi-scale inductive biases required for dense prediction, while simultaneously improving computational efficiency. Swin Transformer [203] represents the most prominent advancement in this direction. It constructs a hierarchical representation by merging image patches at deeper layers, effectively building a feature pyramid. To reduce computational cost, the self-attention is restricted to non-overlapping local windows and a *shifted window* mechanism is introduced to alternates the partitioning between consecutive layers. This design enables cross-window communication and global context modeling with linear complexity $O(N)$, establishing Swin as a standard backbone for high-performance segmentation.

The convergence of these hierarchical backbones with the object-centric paradigm eventually led to the unification of segmentation tasks under a single framework. The MaskFormer series of models [58, 59] challenged the conventional per-pixel classification formulation dominant since FCNs, proposing instead that semantic segmentation be solved as a mask classification problem. In this architecture, a hierarchical backbone, such as the Swin Transformer, extracts

multi-scale features, which are then processed by a pixel decoder to generate high-resolution per-pixel embeddings. A Transformer decoder then employs a set of learnable object queries to attend to these embeddings and predict a set of binary masks and their corresponding class labels. This formulation has since become a reference paradigm for fully supervised semantic segmentation and provides the architectural foundation upon which many supervised open-vocabulary segmentation methods are built.

2.3 Self-Supervised Learning

Despite the remarkable successes achieved by fully supervised learning, several real-world applications in Computer Vision face a common challenge where there is a large abundance of unsupervised data with respect to supervised. Thereby, researchers started to investigate paradigms in machine learning that exploit unlabeled data for representation learning, such as self-supervised, active, and semi-supervised learning. Self-supervised learning aims to intrinsically generate labels from unlabeled data, leveraging supervisory signals coming from data themselves, such as relationships between components and different views. The key idea is to design tasks that encourage the model in learning meaningful representations by solving automatically defined objectives, without relying on human-annotated data. This paradigm has led to the development of models capable of learning semantically rich and transferable visual features, which can be effectively reused for a wide range of downstream tasks, such as image classification and retrieval, with only a minimal amount of supervised data required to train lightweight prediction heads on top of them.

Gui *et al.* [114] organize the self-supervised learning techniques into three main categories: (i) context-based methods, which are based on inherent contextual relationships among the training images, such as rotation, colorization, and jigsaw; (ii) contrastive learning methods, which aim to force the representations of diverse views of the same image to be similar, while moving them away from negative samples; (iii) generative algorithms, mainly represented by the masked image modeling models such as BEiT [15] and MAE [122], where the model is trained to reconstruct masked o corrupted portions of the image. In this section, we focus on the DINO family of models, highlighting their training strategy and analyzing how their learned representations can be leveraged for unsupervised object localization.

2.3.1 Distillation Without Labels

In this dissertation, we explore the properties of the DINO (DIstillation with NO labels) family of models [37, 77, 234, 291], belonging to the contrastive learning-based methods, intending to obtain a universal vision model that is capable of performing multiple tasks with the same weights, including open-vocabulary segmentation. The original formulation of DINO is inspired by knowledge distillation, in which a student network is trained to produce the results of a given teacher network. In DINO, the two networks share the same architecture, and the teacher network is dynamically built during training as an exponential moving average of the student network. Given an image, two global views and several local views are obtained by randomly cropping the image. The student network processes all the views, while the teacher network processes only the global views. Then, the loss forces the student to predict the global representations of the teacher from the local crops of the image. The DINO pre-training has been tested on both the ResNet [125] and ViT [91]. On the latter, it has been observed that the pre-training leads to strong localization properties. Indeed, by observing the attention between the classification token and the patch tokens, it can be seen that the largest attention scores focus on the foreground objects. Moreover, DINO embeds a property named semantic coherence, for which semantically similar regions in an image should produce similar patch representations in the vision encoder [227].

DINOv2 [234] has been introduced as a scalable evolution of the framework, leveraging a large curated dataset of 142 million images and combining the image-level distillation loss of DINO with the patch-level masked image modeling of iBOT [423]. This combination allows the model to learn feature maps in which the patch embeddings semantically describe the portions of the image that are covered by the patches themselves. However, qualitative analysis revealed the presence of artifact tokens in the feature maps of DINOv2. These artifacts are patch tokens with a norm higher than the others, and acquire global information about the image rather than describing the corresponding portion of the image. To mitigate their presence, Darcet *et al.* [77] proposed to explicitly add a set of register tokens to the input sequence, which join the classification token in absorbing global information, freeing the patch tokens.

Most recently, DINOv3 [291] further increased the scale of the framework, training models with up to 7 billion parameters on a dataset of 1.7 billion images. A key insight was that extended training often degraded the quality of patch-level features even if performance on global tasks improved. To counter this, the authors introduced the Gram Anchoring, a regularization objective that preserves the cor-

relation structure of patch features throughout training. This ensures that the model maintains the fine-grained localization capabilities required for segmentation, even at a massive scale.

2.3.2 Object Localization with Self-Supervised Features

The strong localization properties exhibited by DINO have led several works [146, 217, 289, 339, 341, 321] to investigate how to leverage such features to effectively detect and segment the relevant objects in the scene without training and labels. These methods recall the graph theory [283, 284, 355, 395], traditionally used for image segmentation, interpreting image units, such as pixels, as nodes, and the dissimilarity between them as the weight for the edges connecting the nodes. An undirected graph G on the feature maps of DINO can be built employing the N patches as nodes and the cosine similarity between them for the edges. Most of the methods use the keys from the last attention layer of DINO as patch embeddings due to their improved correlation properties [289, 339, 341]. The graph G, given the similarity matrix between patches $W = (W_{ij})_{1 \leq i,j \leq N} \in \mathcal{R}^{N \times N}$, can be represented through the adjacency matrix $\bar{W} = (\bar{W}_{ij})_{1 \leq i,j \leq N} \in \{0,1\}^{N \times N}$, such that

$$W_{ij} = \begin{cases} 1 & \text{if } W_{i,j} \geq \delta \\ \epsilon & \text{otherwise.} \end{cases} \tag{2.3}$$

with $\delta \in [0,1]$ as a constant threshold and $\epsilon \geq 0$ as a small value used to obtain a fully connected graph if required by the method.

A first line of research, starting from the assumption that the objects in the scene occupy less space than the background, investigates the detection of seed patches from which to start building object clusters, considering the patches connected to the seeds [289, 321]. In particular, LOST [289] employs the patches with the lowest amount of connections as seeds, while MaskDistill [321] uses the attention maps between the classification token and the patches from the last layer to identify the seeds. Another research direction adapts the normalized cut algorithm to perform spectral clustering on the graph built from the feature map of DINO [146, 217, 339, 341]. Such methods involve computing the second smallest eigenvalue and the corresponding eigenvector from the generalized eigensystem, as discussed in 2.1.3, to get the optimal cut that partitions the image into two clusters. In TokenCut [341], the salient object segment is selected as the cluster containing the patch with the largest norm, while Melas-Kyriazi *et al.* [217] select the smallest component. To produce multiple object masks, MaskCut [339] uses iteratively

TokenCut while disconnecting the detected object partitions and computing the optimal cut on the remaining patches. The resulting approach is then used as an annotator to train a detector, such as Mask R-CNN, and create the model named CutLER. Differently, Kara *et al.* [146] select the N smallest eigenvectors from the generalized eigensystem and represent each patch according to the feature space defined by these eigenvectors, to finally perform a k-means in the new feature space and obtain the segmentation masks.

2.4 Open-Vocabulary Semantic Segmentation

The traditional paradigm of semantic segmentation relies on fully supervised learning with densely annotated datasets and a fixed set of predefined categories. While this approach has led to significant advances, it remains fundamentally limited in scalability due to the cost of manual labeling and the rigidity of closed-set classification. In contrast, the real-world visual world is inherently open-ended, with an ever-growing collection of categories and concepts that models may encounter. To address this, the research community has increasingly shifted toward open-vocabulary segmentation, a setting in which the model is expected to recognize and localize any category described by the user through free-form text, without requiring pixel-level annotations for that category at training time.

This section provides a comprehensive overview of the developments in open-vocabulary semantic segmentation. We begin by discussing vision–language pre-training, which enables zero-shot recognition by aligning image and text features in a shared space. We then explore how these representations have been extended from image-level classification to dense prediction tasks, distinguishing between fully supervised methods that leverage existing segmentation models, weakly supervised approaches that rely on noisy image–text pairs, and training-free strategies that exploit the internal representations of pre-trained models. We further examine how recent generative architectures, such as Stable Diffusion [266], contribute to dense multimodal alignment, and how the Segment Anything [154] family of models has redefined prompt-based segmentation. Together, these research directions represent the diverse strategies being explored to bridge vision and language for scalable and flexible segmentation.

2.4.1 Vision-Language Pre-Training

The past few years have seen an increasing interest in vision-language pre-training, encouraged by the success of Transformer models in modeling natural language [322] and the availability of web-crawled image–text data containing billions of samples [104, 276, 275, 281]. In this paradigm, a model learns joint visual and textual representations by aligning images with their accompanying captions in a shared embedding space [99, 98, 136, 183, 177, 176, 249, 300, 393, 398]. The most popular and adopted model is CLIP [249], which has been trained on hundreds of millions of image–text pairs to associate visual content with natural language descriptions. The training objective of CLIP is based on a dual-stream contrastive learning approach. The image encoder and text encoder are trained to produce embeddings that are close together for a matching image–caption pair and far apart for mismatched pairs. This is achieved via the InfoNCE loss, which treats the paired caption of each image as the sole positive and all other captions in the batch as negatives. Formally, for a batch of B image–text pairs with image features v_i and text features t_i, the contrastive loss can be written as:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{2B}\sum_{i=1}^{B}\log\frac{\exp(\text{sim}(v_i,t_i)/\tau)}{\sum_{j=1}^{B}\exp(\text{sim}(v_j,t_i)/\tau)} -\frac{1}{2B}\sum_{i=1}^{B}\log\frac{\exp(\text{sim}(v_i,t_i)/\tau)}{\sum_{j=1}^{B}\exp(\text{sim}(v_i,t_j)/\tau)}.$$

where $\text{sim}(v,t)$ is a similarity (*e.g.* cosine) between image and text embeddings, and τ is a temperature parameter. Intuitively, this loss pushes v_i close to its correct caption t_i while pushing it away from other non-matching captions, and vice versa. Through this training, the model learns a multimodal embedding space where images are positioned near the text that best describes them. A key outcome of vision–language pre-training is the emergence of zero-shot recognition capabilities. Because the visual features of the model are learned in alignment with natural language, it can perform classification by simply comparing the embedding of an image to the embeddings of candidate category names, provided as text prompts. In this way, a model trained on certain labeled categories, referred to as *seen* classes, can recognize *unseen* classes at inference time by using their names or descriptions, without any further fine-tuning.

Following the success of CLIP, several works have explored improvements in vision–language pre-training. Since image–text pairs from the web can be noisy, one line of research has focused on data quality, proposing approaches

for cleaning or enriching the captions [177]. Another direction is to incorporate auxiliary training objectives alongside the basic contrastive loss, including vision-to-language generation like captioning [176, 390] and self-supervised learning objectives [99, 98, 226, 337]. In terms of architectures and training strategies, notable progress has been made by leveraging pretrained components or modifying the loss function [317, 393, 398]. In particular, LiT [399] demonstrated that one can take a high-quality pre-trained vision backbone, like DINO [37], freeze its weights, and then train a text encoder to align with it on image–text data. This simple recipe effectively "transfers" the strong visual features into a multimodal model, achieving superior zero-shot classification accuracy with much less training cost than training both towers from scratch.

2.4.2 Fully-Supervised Open-Vocabulary Segmentation

The remarkable success of vision-language pre-training in global-level tasks, such as classification and retrieval, has motivated a shift toward extending these capabilities to dense prediction tasks, including detection and segmentation [173]. Unlike image-level labels, acquiring pixel-level masks or bounding boxes is prohibitively expensive and time-consuming [190], requiring human annotators to precisely draw the regions corresponding to the categories. Thereby, such data is restricted to closed sets of categories, and, hence, the zero-shot approach, which investigates how to transfer the learning from the seen available categories to unseen categories, has gained increasing interest. More recently, the zero-shot paradigm has been generalized into the open-vocabulary paradigm, in which the model has to be able to recognize and localize any object category represented by the user through free-form text at inference time [179, 221, 419]. However, the data annotation bottleneck in dense predictions makes transferring the vision-language understanding from the global image level to a fine-grained level a still open challenge. Indeed, the vision-language pre-training is made on large sets of images paired with textual descriptions that describe the whole image, lacking localized supervision.

To overcome this, a dominant line of research exploits the decoupling of localization and classification. These approaches rely on the observation that object-centric detection and segmentation models, trained on a closed dataset, learn the concept of "*objectness*", namely what is an entity in the scene, and hence the proposed class-agnostic boxes and masks are suitable also for categories which do not belong to the closed set. This idea has been directly adopted by works, such as ZegFormer [88], ZSSeg [368], and OVSeg [187], that employ a

two-stage approach, explicitly separating the pipeline into mask generation and classification steps. These methods utilize an object-centric segmentation model (*e.g.*, Mask2Former [58]) trained on seen classes to generate a set of binary masks. In the second stage, the image regions corresponding to these masks are cropped, masked, and augmented via prompt engineering before being fed into a CLIP encoder. By treating each mask crop as an independent image, these methods leverage the native zero-shot classification power of CLIP to assign open-vocabulary labels to the proposed segments. However, along with the computational overhead given by running CLIP on hundreds of mask proposals, these methods suffer from the domain shift given by running CLIP on small masked crops and do not exploit the context of the whole image, which may help in recognizing the objects in the scene [118, 108, 392, 368]. Hence, another line of work accompanies the class-agnostic mask proposals generated by an object-centric segmentation model by pooling the mask features directly from a feature map computed on the whole image. In particular, FC-CLIP [392] directly adopts the convolutional CLIP model as the backbone of Mask2Former to use it to both generate class-agnostic masks and provide the feature map for mask pooling. Similarly, other works have explored using the supervision of segmentation data to enable localization capabilities from CLIP while leveraging its multimodal understanding. SAN [367] introduces a lightweight side network that runs in parallel to the frozen CLIP model, fusing its strong semantic features with the high-resolution geometric details learned by the side branch, while CAT-Seg [61] improves the feature alignment by explicitly computing a dense cost volume between the image and text embeddings, which is then refined by a learned aggregation module to resolve spatial ambiguities.

While proposal-based methods effectively leverage pre-trained segmentation models, they fundamentally rely on a *late fusion* paradigm: the visual model generates class-agnostic masks in isolation, and textual alignment occurs only at the final classification stage. This disconnect prevents the textual prompt from influencing the proposal generation process, often leading to missed objects if they were not part of the ones proposed by the class-agnostic regions. To address this, a parallel line of research explored *early fusion* architectures, where textual descriptions condition the visual feature extraction and decoding process itself [142, 179, 198, 404]. The success of this paradigm paved the way for universal segmentation frameworks that jointly tackle multiple tasks to maximize performance, demonstrating that training across these tasks leads to mutual benefits and robust open-world perception [248, 352, 375, 430, 431].

2.4.3 Weakly-Supervised Open-Vocabulary Segmentation

Instead of employing external architectures trained on densely annotated data to combine localization capabilities with the multimodal understanding of CLIP, a parallel research direction has investigated whether segmentation capabilities could directly emerge from extensive training on noisy image-text pairs alone. This paradigm, often referred to as weakly-supervised open-vocabulary segmentation and evaluated along with training-free open-vocabulary methods in the unsupervised setting defined by Cha *et al.* [39], eliminates the need for expensive pixel-level annotations. However, since the supervision of the captions is global, the core challenge lies in learning how to assign specific semantic concepts to local regions without explicit spatial labels. A first line of work investigates novel Transformer architectures designed to facilitate the emergence of semantic regions by explicitly grouping image patches during the forward pass through group tokens [362, 363], learnable center [209], slot attentions [366], online clustering [196], and prototype-based grouping [402]. This forces the model to aggregate semantically similar patches into coherent regions that also correspond to the textual concepts.

While grouping methods modify the architecture, another stream of research retains standard backbones but modifies the training objective to enforce alignment at the patch level [39, 227, 255, 351, 384]. The standard CLIP loss aligns the global classification token with the caption, lacking direct supervision on the patch embeddings. On the contrary, these methods involve the patch embeddings in the contrastive loss, aligning the output of a pooling of the patches [255], treating the patch-text alignment as a multiple-instance learning problem [227], introducing grounding mechanisms that maximize the similarity between local visual features and the corresponding nouns in the caption [39, 351, 384], and exploiting multi-view consistency learning [260]. These approaches are not only used as open-vocabulary segmentation models, but also lead to building multimodal backbones with the semantic coherence property introduced in 2.3.1 for DINO.

A more recent direction focuses on this semantic coherence property and acknowledges that text supervision alone is often too sparse to learn precise object boundaries [228, 353, 357]. Hence, it introduces the guidance of self-supervision to improve the localization properties of CLIP. SILC [228] pre-trains the model, combining the sigmoid contrastive loss of SigLIP [398] and the self-distillation of DINO to establish the multimodal understanding on top of semantically coherent patches. CLIP-DINOiser [357] teaches CLIP to predict the affinity matrix of DINO (*i.e.*, self-similarities among patches) and uses it to recombine the last

layer patch embeddings. CLIPSelf [353] uses a distillation paradigm in which the teacher is a frozen CLIP that processes image crops whose classification tokens are matched with the corresponding pooled patch embeddings of the student.

2.4.4 Training-Free Open-Vocabulary Segmentation

Researchers have begun to question whether open-vocabulary semantic segmentation capabilities can be directly extracted from the pre-trained CLIP model or by combining it with other models, such as DINO, which exhibit strong localization properties, to uncover such capabilities directly [13, 25, 184, 191, 280, 328, 379, 422]. This setting is also referred to as training-free and does not involve any other training process, but investigates the properties of the currently available models. A series of works have started investigating whether CLIP truly does not embed localization capabilities or whether this is a consequence of directly adopting the output patch embeddings for segmentation. Indeed, it has been observed that the patch embeddings in the last layer of CLIP are spatial invariant, meaning that the content of the patch embeddings is not related to the spatial positions that they occupy in the image. This phenomenon can be observed from the query-key attention scores between the patches themselves, which result in being similar across patches independently of their location in the image, and the attention score with the classification token, which results in being larger than the others [184, 328, 379, 422]. The impact of the classification token regards only the last layer, which is directly provided to the contrastive loss, but the intermediate layers preserve local coherence in the attention scores [191, 379]. The pioneering approach MaskCLIP [422] proposed to directly adopt the value from the last attention block as output patch embeddings, bypassing the query-key attention, while following studies replaced this attention with value-value attention [184], query-query and key-key attention [328], ensembling on multiple self-self attentions [25, 280], query-key attentions from intermediate layers of CLIP [379], or attention scores from the last layer of another image encoder such as DINO [166]. However, these approaches have been demonstrated to be effective when applied to the ViT-B model, but encounter a significant drop with the ViT-L. Lan *et al.* [165] attribute this effect to the residuals in the last layer, which harm the segmentation results by adding noise and present a larger norm in the deeper layers of the ViT-L. Thereby, removing the residuals and the last feed-forward layer has been shown to improve the segmentation quality [379]. Other than the classification token, large attention scores harming the semantic coherence of the patch embeddings have been observed between patches and artifact tokens [13, 280]. In SC-CLIP [13],

the authors propose to detect and replace these tokens to enhance the performance of the model.

2.4.5 Stable Diffusion

In recent years, diffusion models have revolutionized image generation with their ability to produce high-quality images. In particular, Stable Diffusion [266] has gained enormous interest as a text-to-image model that performs denoising in a latent space, encoded by a VAE, instead of directly on pixels. This model employs a U-Net [267] as the denoising network, injecting textual conditioning through cross-attention layers interleaved with self-attention layers at multiple resolutions. In each denoising step, the cross-attention mechanism links word embeddings from the caption to specific regions in the latent image, while self-attention captures relationships among visual features. This design enables Stable Diffusion to learn fine-grained correspondences between text and image content during its iterative training, which involves gradually adding noise to an image and tasking the model with a conditioning caption to remove it.

Along with outstanding image synthesis capabilities, researchers have investigated whether the internal attention maps of Stable Diffusion can be used to localize concepts mentioned in the prompt. Tang *et al.* [304] conduct a deep analysis of cross-attention distributions in Stable Diffusion, known as DAAM, showing how different types of words (*e.g.*, nouns and verbs) are attributed to generated image regions. Based on this, DiffuMask [354] uses the aggregated cross-attention scores for each token to automatically obtain semantic masks on a set of synthetic images, effectively generating ground truth masks without human annotation and enabling the training of traditional segmentation models on them. On the contrary, GroundedDiffusion [186] exploits the supervision of annotated segmentation data to learn a grounding mechanism to locate any category during the Stable Diffusion generation process. These studies confirm that, unlike the global image-text alignment of CLIP, the fine-grained multimodal training objective of Stable Diffusion augments the model with the inherent ability to associate words with specific visual regions. As discussed in Sec. 2.4.4, zero-shot segmentation limitations of CLIP stem from its image-level training objective, which provides only weak spatial supervision at the pixel level. Stable Diffusion, in contrast, was trained with a localized text-conditioning objective, suggesting its features may be more suitable for dense predictions. One line of research has thus explored incorporating Stable Diffusion into segmentation architectures. For instance, ODISE [364] uses the frozen Stable Diffusion model as a backbone within a Mask2Former-like [58]

framework, combining its rich latent visual features with a learned mask decoder to achieve open-vocabulary segmentation.

In parallel, a number of training-free approaches directly exploit internal representations of Stable Diffusion to perform segmentation without any additional fine-tuning. CLIPper [298] refines the noisy patch-level predictions of CLIP with the high-resolution details extracted from the self-attention maps of Stable Diffusion. DiffSegmenter [332] and iSeg [299] both exploit the attention maps to delineate objects for arbitrary categories at inference time. In particular, they use the diffusion cross-attention maps to identify coarse object regions corresponding to a given text label, and then refine these regions using the self-attention maps to improve boundary accuracy. MaskDiffusion [148] further pushes this idea by employing spectral clustering to propose class-agnostic regions from the self-attention of Stable Diffusion while using the cross-attentions to localize the specific categories. OVDiff [147] generates a batch of images for a given category via the diffusion model and then encodes these images into a feature space, using a pre-trained vision model like CLIP, DINO, and Stable Diffusion itself. Segmentation is achieved by comparing the features of the target image to these category prototypes, effectively matching regions of the target image to the generated examples.

Overall, the integration of Stable Diffusion into open-vocabulary segmentation, whether by utilizing its attention maps or by generating reference samples, has opened up promising directions. Diffusion models bring fine-grained vision–language alignment and a virtually unlimited vocabulary learned from web-scale data, addressing some of the key challenges left by CLIP-based methods.

2.4.6 Segment Anything

The Segment Anything Model (SAM) [154] was introduced as a foundation model for promptable image segmentation, trained on an unprecedented dataset of over 1 billion masks (SA-1B) to enable strong zero-shot performance. Its architecture follows a simple design: a powerful image encoder, a ViT pretrained via MAE, produces a one-time embedding for the image, a prompt encoder encodes user prompts, such as points, boxes, and masks, and a lightweight mask decoder combines these to output the segmentation mask. This design allows SAM to generalize to any object in an image, given the appropriate prompt, returning a reasonable mask even from ambiguous cues. SAM2 [258] extends this idea to video segmentation by introducing hierarchical features and temporal memory. It replaces the ViT backbone with the hierarchical transformer Hiera [271] to provide multi-scale feature maps for finer mask detail. More critically, SAM2 adds

a memory mechanism to propagate masks across frames by including a memory encoder and memory bank that store key embeddings from previous frames, and a memory attention module that lets the features of the current frame attend to past frames. This streaming memory design enables temporally consistent object masks even through occlusions or when objects reappear.

Researchers have also begun adapting SAM to specialized segmentation tasks beyond its original scope, such as personalized segmentation [202, 409], where the model has to segment the same object indicated in a reference provided by the user, camouflaged segmentation [49], where objects share similar textures and patterns with the background, few-shot segmentation [72], where the model is tasked with learning to segment a category from a given set of references, and medical image segmentation [424]. A major line of research augments SAM and SAM 2 with open-vocabulary segmentation abilities, enabling them to segment by category names rather than just spatial prompts [73, 119, 169, 174, 261, 285, 301, 327, 360]. A straightforward approach is to combine SAM with an open-vocabulary detector. For instance, Grounded-SAM [261] uses Grounding DINO [199] to generate bounding boxes for a given text prompt, which SAM then takes as input to produce the corresponding mask. ESC-Net [169] proposes to fuse features from CLIP with the SAM decoder by generating pseudo-text prompts from image–text correlations and feeding them into the mask decoder, so that the output masks are inherently tied to open-vocabulary classes.

Most recently, SAM3 [35] has been released, which builds multimodal capabilities directly into the model rather than via external adapters. SAM 3 targets the new task of Promptable Concept Segmentation (PCS), where the goal is to segment all instances of a given concept in an image or video, based on a prompt that can be a text phrase, an example image, or both. To achieve this, SAM 3 exploits a Perception Encoder [23] backbone that is shared between image and text inputs, followed by an early fusion module. The fused features are then passed to a DETR-like [36] decoder with a set of learned object queries, which produces a set of object masks along with a novel *presence* token that indicates whether the queried concept is present.

2.5 Object-based Embodied AI

While much of the progress in visual representation learning, open-vocabulary segmentation, and self-supervision has been demonstrated in traditional image datasets, their advances have been fundamental in enabling intelligent behavior in

interactive environments. Embodied AI provides a natural and challenging testbed for this vision: it places an agent with visual perception, language understanding, and action capabilities inside a simulated or real environment, where it must ground its decisions in real-time sensor input and respond to complex open-ended goals. In particular, object-based Embodied AI focuses on tasks where the agent must reason about specific objects, either through category names, visual references, or instance-level goals, mirroring the kind of object-centric understanding explored throughout this dissertation. This setting makes it possible to study how vision models trained on large-scale, weakly supervised, or self-supervised objectives transfer to goal-directed navigation and interaction.

In this section, we review two foundational aspects of object-based Embodied AI: the environments and datasets that define the simulation spaces and task protocols, and the architectural approaches used to build navigation agents capable of completing object-centric tasks, from modular pipelines to end-to-end learning systems.

2.5.1 Object-based Embodied Datasets

A foundational component of Embodied AI research is the availability of high-fidelity simulation environments and annotated datasets that support reproducible and scalable experimentation. These platforms provide 3D scenes, realistic physics, and programmable task definitions, enabling the development and benchmarking of agents across a range of navigation and manipulation tasks.

Recent years have witnessed the release of several versatile simulators such as Habitat [245, 274, 303], AI2-THOR [157], RoboTHOR [78], and ProcTHOR [80], as well as datasets of scenes for robotic navigation like Gibson [282, 358], Matterport3D [40], and Habitat-Matterport3D (HM3D) [253]. The evaluation of the capabilities of such agents can be performed on multiple embodied tasks [7, 259, 288] mimicking different real-world requirements. PointGoal Navigation (PointNav) [6] requires the agent to reach specific relative coordinates to its starting position. In object-oriented navigation, the agent is tasked to find any instance of an object category (ObjectNav) [6, 17], multiple objects in sequence (MultiON) [344], or a specific instance of a category (ION) [180]. Other embodied navigation tasks are ImageGoal navigation (ImageNav) [62, 429] that requires the agent to reach the position where the goal image has been taken, and a more object-oriented formulation of ImageNav called Instance-Specific Image Goal Navigation (InstanceImageNav) [160] that sets as the navigation goal a precise object instance given a photo of it. Recently, the GOAT-Bench benchmark has been introduced, which

requires finding sequences of target objects using multimodal references [150].

2.5.2 Object-based Navigation Agents

With benchmarks and datasets in place, a central challenge becomes designing agents that can complete these tasks: navigating, exploring, and interacting with the environment based on object-centric goals. Broadly, object-based navigation agents can be categorized into modular pipelines and end-to-end models, each offering trade-offs between interpretability, flexibility, and learning capacity.

Modular approaches decompose the problem into structured components, often including a semantic mapper, an exploration policy, and an object detection module. Some approaches adapted the architecture proposed by ANS [43] for object goal navigation by building semantic maps to locate the target [42, 167, 252, 427]. Following, Stubborn [208] proposed a strong baseline using a heuristic exploration method. In Mod-IIN and GOAT, to tackle the InstanceImageNav task, the keypoint matching method SuperGlue is employed to match the goal object image with the current observations of the agent. SuperGlue [273] leverages an attention-based graph neural network on local descriptors, as the ones extracted with the SuperPoint model [85]. IEVE [170], instead, proposes an Exploration-Verification-Exploitation framework that combines a segmentation model and a keypoint matcher to recognize distant objects and confirm them when the agent is closer. Among end-to-end methods, Mousavian *et al.* [225] and Yang *et al.* [378] worked on improving visual representations, Mayo *et al.* [216] used spatial attention maps, and Ye *et al.* [381] used auxiliary tasks. Other related work leveraged object relation graphs [94, 95, 237]. THDA [213], instead, used 3D scans of objects from YCB dataset [32] to augment the training dataset. Recently, PIRLNav [254] used a two-stage learning strategy, Chen *et al.* [53] used a method based on recursive implicit maps, and OVRL [371, 372] exploited self-supervised visual pretraining to boost agent capabilities. Additionally, zero-shot object goal navigation has been recently explored by ZER [4], ZSON [212], and ORION [76].

Chapter 3

Open-Vocabulary Segmentation via Prototypes

In Chapter 2, we retraced the evolution of the paradigms and methods of image segmentation, starting from the localization of the *Gestalten*, visual homogeneous units based on low-level cues, to the powerful open-vocabulary semantic segmentation techniques, capable of segmenting any semantic category provided by the user in the form of free-form textual prompts. The most straightforward approaches to obtain such methods fall into revisiting the vision-language pre-trained models, as support of traditional architectures commonly used for fully-supervised and closed-set segmentation [88, 187, 352, 368, 392], the employment of diverse objective functions that enforce learning a multimodal alignment at a fine-grained level of the image [39, 227, 255, 351, 384], and architectural modifications that uncover the latent capabilities already embedded in multimodal models trained with global objectives [13, 25, 184, 191, 280, 328, 379, 422]. However, another research direction introduced the concept of avoiding the employment of a direct alignment between the vision encoder and the textual embeddings in favor of creating visual reference, also referred to as prototypes, associated with the textual concepts provided by the user [38, 147, 286]. A single textual embedding is often insufficient to represent the intra-class variability of the semantic categories, while the visual references enable catching diverse characteristics such as shape, pose, color, and illumination.

In this Chapter, we introduce a series of open-vocabulary semantic segmenta-

tion methods based on the retrieval of visual prototypes. In Sec. 3.1, we present VOCSeg, a method that leverages the weak supervision of the captions regarding which textual elements can be localized in the corresponding images. Thereby, such elements can be segmented using another open-vocabulary segmenter to build a *visual vocabulary*, a sort of atlas that shows a set of visual references for each category. A given segment can be classified through its CLIP [249] embedding, not only employing a visual-textual similarity with the category embeddings, but also using a visual-visual similarity with the visual references.

The prototype-based paradigm enables equipping the open-vocabulary semantic segmentation functionality in any visual backbone that exhibits semantic coherence properties. Thereby, in Sec. 3.2 we introduce FOSSIL, a training-free method that aims at transforming DINOv2 [234] into a vision encoder able to segment textual concepts through the retrieval of visual prototypes. Such prototypes are built exploiting the property of Stable Diffusion of localizing the regions of the generated images that correspond to the nouns in the conditioning captions. Employing DINOv2 as a multimodal vision encoder is a first step towards the creation of a universal model capable of performing several vision tasks in parallel. Indeed, we propose OpenCut, a graph-based iterative approach that extends the previous studies on DINO [37] to generate class-agnostic masks, which are subsequently assigned to the semantic categories.

Finally, in Sec. 3.3 we present FreeDA, an enhancement of the training-free framework based on building visual prototypes from the synthetic images generated by Stable Diffusion. Through the employment of superpixel algorithms to convert the subdivision into patches of DINOv2 into a more fine-grained pixel-aware representation, and with the injection of the multimodal contextual understanding of CLIP, FreeDA demonstrates that, with the use of prototypes, self-supervised backbones can achieve outstanding performance in open-vocabulary segmentation while retaining explainability properties.

3.1 VOCSeg

In this Section, we propose VOCSeg, a prototype-based open-vocabulary semantic segmentation architecture, depicted in Figure 3.9. We augment the two-stage methods [88, 187, 368], which reformulate the task into dividing the image into coherent regions and classifying each region, with retrieving references for the categories from a pre-built visual vocabulary.

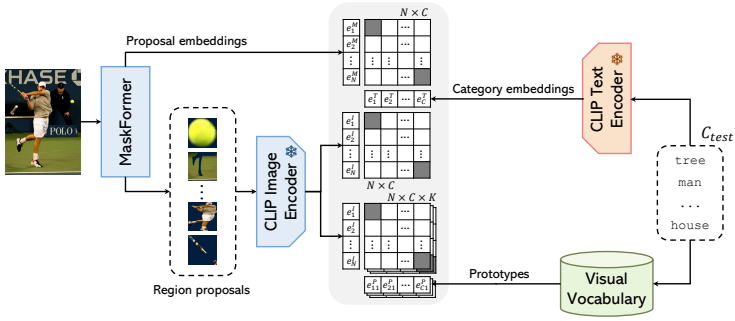

Figure 3.1: Overview of our proposed method, VOCSeg, for two-stage open-vocabulary semantic segmentation enhanced by visual prototype retrieval.

3.1.1 Method

The architecture of VOCSeg comprises three main components: a mask proposer, an enhanced CLIP model with retrieval capabilities, and a visual vocabulary. The mask proposer generates region proposals within the image, while the CLIP model extracts embeddings for these proposed regions. These embeddings serve as representations for independent open vocabulary classification of each region. However, it is essential to consider the domain shift introduced by cropping and masking regions, as it deviates from the training images of CLIP. To mitigate this domain shift, we employ the concept of visual prototypes. Firstly, we employ a two-stage segmentation method on a dataset consisting of image-text pairs to obtain region proposals for a diverse range of words. These proposals collectively form the visual vocabulary, which encapsulates the domain shift resulting from the cropping and masking process. Subsequently, we generate visual prototypes for each word by clustering the corresponding set of collected regions. These prototypes serve as representative embeddings within the feature space.

At inference time, we leverage textual category embeddings and retrieved prototypes for each category. These prototypes reside in the same feature space as the embeddings and allow the framework to incorporate both textual and visual similarities using only the CLIP model, avoiding an increase in computational effort.

3.1.1.1 Prototype Extraction from Image-Caption Pairs

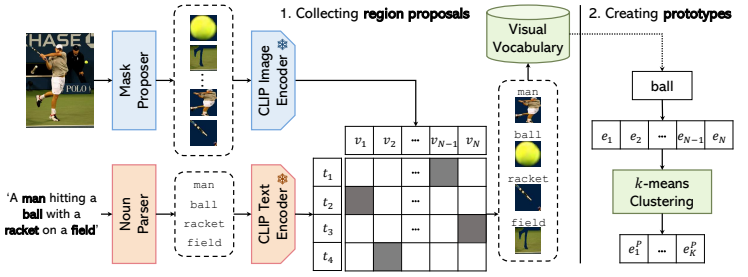

Figure 3.2: Overview of the approach for collecting region proposals starting from image-caption pairs and of the clustering process used to generate prototypes.

Collecting a visual vocabulary. In our approach to open-vocabulary segmentation, it is crucial to utilize prototypes that capture both the distinctive features of each category and the domain shift resulting from masking the regions. These prototypes play a pivotal role in classifying the proposed regions by identifying visually similar correspondences. However, collecting regions for a large vocabulary represents a challenge, making the use of pre-annotated segmentation datasets infeasible due to their limited category coverage. To tackle this challenge, we adopt a *self-labeling* strategy for constructing an open-vocabulary collection of regions. This strategy involves extracting regions from a dataset of image-caption pairs, associating them with a vocabulary based on their corresponding captions, and subsequently generating prototypes through the clustering of similar embeddings, as shown in Figure 3.2.

Specifically, we extract nouns from each caption, incorporate them into a text prompt, and provide them as input to the Text Encoder of a CLIP model. Subsequently, we obtain mask proposal embeddings using the Image Encoder of the same CLIP model and match the mask proposals with each noun using their respective computed embeddings. Although this matching process may introduce some noise, the presence of the noun in the caption ensures that one of the masks must be related to the corresponding object. Finally, we singularize the extracted nouns and store the CLIP embeddings of each match in a visual vocabulary.

Generating prototypes. Finally, we perform a k-means clustering on the set of collected region embeddings for each noun in the vocabulary to generate a set of prototypes, represented by the cluster centroids. The k-means algorithm groups similar features, forming representative prototypes for each noun category. In this way, we ensure that our prototypes capture a wide range of visual characteristics.

Handling rare nouns. There are cases where the number of collected embeddings may not be sufficient to perform k-means clustering effectively, either due to a limited correspondence in the captions or arbitrary test categories that do not match entries in the visual vocabulary. For these rare nouns, we employ a k-nearest neighbors algorithm. This algorithm matches the textual embeddings extracted using CLIP with the most similar words present in the vocabulary. Subsequently, we perform k-means clustering on the embeddings of the N neighbors to generate prototypes. We increment the value of N until we have an adequate number of embeddings to perform the k-means clustering effectively.

3.1.1.2 Two-Stage Open-Vocabulary with Prototype Retrieval

The objective of two-stage open-vocabulary semantic segmentation is to identify a pair of mappings $(\mathcal{S}, \mathcal{L})$ for an input image $I \in \mathbb{R}^{H \times W \times 3}$ across C_{test} arbitrary categories. In this task, $\mathcal{S}$ partitions I into a set P of T regions, defined as follows:

$$P = \{P_i, \}_{i=1}^{T} \quad \text{with} \quad P_i \subseteq I, \cup_{i=1}^{T} P_i = I, \forall i, j : i \neq j, P_i \cap P_j = \emptyset\,, \quad (3.1)$$

whereas $\mathcal{L}$ assigns a category $c \in C_{\text{test}}$ to each region $P_i \subseteq I$, where $i = 1, \ldots, T$.

Extracting Mask Proposals Embeddings. To obtain class-agnostic mask proposals, we utilize MaskFormer [59]. This model is trained on a set of classes C_{train}, nevertheless, as reported by Xu *et al.* [368], it can generate T high-quality mask proposals $\{M_i\}_{i=1}^{T}$ and their corresponding mask embeddings, even for unseen classes. Each mask proposal $M_i \in \mathbb{R}^{H \times W}$ is converted into a binary mask $M_i^B \in 0, 1^{H \times W}$ by applying a sigmoid function followed by thresholding. The binary mask indicates the location of the object in the input image.

In the original MaskFormer [59] architecture, the mask embedding is a C_{train}-dimensional distribution that represents the probability of each training class. To extend the model to an open-vocabulary setting, inspired by [187, 368], we modify MaskFormer in such a way that each mask generates an F-dimensional embedding, where F is the embedding dimension of a CLIP model. This adaptation ensures compatibility between the mask embeddings and the CLIP textual embeddings, which are extracted from the nouns of various semantic classes, thus enabling open-vocabulary capabilities. We include an additional F-dimensional learnable embedding for `no-object`.

Further, we also employ the CLIP image encoder to extract an additional set of embeddings from the proposed regions, which complements the ones generated for each region by MaskFormer. In particular, for each binary mask M_i^B, we erase

the unused background, crop around a bounding box that entirely incorporates the foreground area, and resize to the input resolution of CLIP. Then, the region is fed to CLIP to produce an embedding that can be used to compute similarity against the textual category embeddings.

Assigning proposals to classes. For each category in C_{test}, we retrieve a set of K reference prototype embeddings from a visual vocabulary. To compute the final similarities between region proposals and categories, we combine two terms: one that exploits textual category labels and one that exploits the reference prototype embeddings. In particular, for each category $c_j \in C_{\text{test}}$ we extract an embedding e_j^T with CLIP using the Textual Encoder, we retrieve a set of prototypes $\{e_{jk}^P\}_{k=1\dots K}$, and for each region P_i we extract an embedding e_i^I with the Image Encoder of CLIP and an embedding e_i^M with MaskFormer. First, we aggregate the prototype similarities by considering the average of the maximum similarity with the K prototypes assigned to c_j and the mean similarity with all of them. This is a trade-off between considering the nearest reference embedding, which is the most significant for the current region, and the robustness offered by a single average embedding representative for the whole concept:

$$s_{i,j}^P = \frac{1}{2} \max_k \text{sim}(e_i^I, e_{jk}^P) + \frac{1}{2K} \sum_{k=1}^{K} \text{sim}(e_i^I, e_{jk}^P), \tag{3.2}$$

where $i = 1 \dots T$, $j = 1 \dots |C_{\text{test}}|$, $k = 1 \dots K$ and $\text{sim}(\cdot, \cdot)$ is the cosine similarity.

Then, since both the prototype similarities and the textual similarities are computed in the same feature space, we fuse them using a linear combination with weights α and $(1 - \alpha)$. This ensembling strategy rewards the situations in which the textual and prototype similarities agree, while it penalizes cases of disagreement. Formally, the resulting aggregated similarity is defined as

$$\tilde{s}_{i,j} = \alpha s_{i,j}^P + (1 - \alpha) \cdot \text{sim}(e_i^I, e_j^T). \tag{3.3}$$

The probability vector over classes $\tilde{p}$ is computed through the softmax function with a temperature τ.

Fusing with MaskFormer predictions. Since MaskFormer is trained on C_{train}, its performance is biased towards categories belonging to this set. When the object contained in the region P_i is not recognized as a category of C_{train}, MaskFormer produces an embedding similar to the `no-object` embedding. Hence, when the softmax is applied to its similarities, all the resulting probabilities corresponding

to the categories of C_{test} are small, and the one corresponding to `no-object` is large, which is removed after the softmax. So, the final prediction of P_i and c_j is obtained through the weighted geometric mean, with weights β and $(1-\beta)$, between the probability $\tilde{p}$ of the visual-text branch and the probability $\hat{p}$ resulting from MaskFormer, in such a way that the prediction of MaskFormer is enhanced only when it is confident about it (*i.e.*, when c_j belongs to C_{train} too):

$$p_{i,j} = \tilde{p}_{i,j}^{\beta} \cdot \hat{p}_{i,j}^{(1-\beta)}. \tag{3.4}$$

Computing Semantic Segmentation. Finally, mask predictions and probabilities are aggregated to compute the semantic segmentation. Specifically, the score $z_j(q)$ of a category $c_j \in C_{\text{test}}$ in a pixel q is computed as the sum of each mask activation M_i multiplied for the corresponding probability $p_{i,j}$:

$$z_j(q) = \sum_{i=1}^{T} M_i(q) p_{i,j}. \tag{3.5}$$

3.1.2 Experiments

3.1.2.1 Experimental Setup

We train the modified MaskFormer model on the COCO-Stuff dataset, according to [187], with the Swin-B [203] backbone. We follow the original training settings of MaskFormer [59]. We use the OpenCLIP [133] implementation of CLIP with ViT-L/14 backbone trained on LAION2B [275]. To embed the category names with CLIP, we surround them with the text prompts proposed in the original CLIP [249] and in ViLD [113]. To obtain a diverse set of prototypes, we utilize COCO Captions [55]. We collect 15,000 unique nouns from the dataset. To extract binary masks, we apply a threshold of 0.4 after the sigmoid.

3.1.2.2 Ablation Studies

Masking Strategy. We investigate the impact of three different masking strategies for extracting the regions detected by the mask proposer. In particular, MaskFormer generates N mask proposals denoted as $M_i \in \mathbb{R}^{H \times W}$. These proposals indicate the activation level of each position in the image with respect to the detected region. In our main pipeline, referred to as *binary* strategy, we consider the binarized masks $\{M_i^B\}_{i=1}^N$. In order to isolate the foreground object and eliminate the potential interference of surrounding context noise on the open-vocabulary classification

Table 3.1: Ablation on different masking strategies, in terms of mIoU score.

Dataset	Masking Strategy		
	None	Heatmap	Binary
ADE-150	17.7	17.7	**22.5**
PAS-20	82.51	85.0	**93.4**

Table 3.2: Ablation on similarity ensembling, in terms of mIoU score.

Dataset	Similarity		
	Text	Visual	Ensembling
ADE-150	21.0	20.1	**22.5**
PAS-20	92.6	93.2	**93.4**

of the region through CLIP, we erase the background information, keeping solely the foreground object. However, we also acknowledge that in certain cases, the background can provide crucial information for accurately recognizing the object. To address this, we explore two alternative strategies: one in which we crop the region without erasing the background (which we name *none*), and one, instead, in which we attenuate the background by multiplying the image pixels with a normalized heatmap derived from the originally proposed mask (termed *heatmap*). This allows us to retain some contextual information while still emphasizing the foreground object of interest.

Our experimental results, as reported in Table 3.1, demonstrate that the *binary strategy* provides the best mIoU scores. We argue that the noise introduced by the background overwhelms any potential advantage gained from the contextual information when it comes to clarifying the foreground object.

Ensembling. In our method, we introduce the usage of CLIP for both image-to-text and image-to-image similarities to leverage their benefits concurrently. In Table 3.2, we present a comparison between the individual usage of these similarities, as well as their ensembling. The results show a significant improvement of +1.5 mIoU on the ADE-150 dataset and +0.2 on the PAS-20 dataset compared to the baseline that considers only visual similarity. We argue that the reason behind this observed improvement is the complementary nature of the two types of similarities provided by CLIP. Image-to-text similarity captures the semantic understanding of the textual information associated with the images, while image-to-image similarity focuses on the shared visual content between images.

In Figure 3.3, we present the trend of the mIoU as a function of the ensemble weight, for both ADE-150 and PAS-20 datasets. Notably, we observe that the performance trends differ between the two datasets, with ADE-150 performing better when assigning a larger weight to the text similarity, while PAS-20 performs better with a larger weight assigned to the visual similarity. We hypothesize that this discrepancy is influenced by the number of arbitrary categories in (C_{test}) and

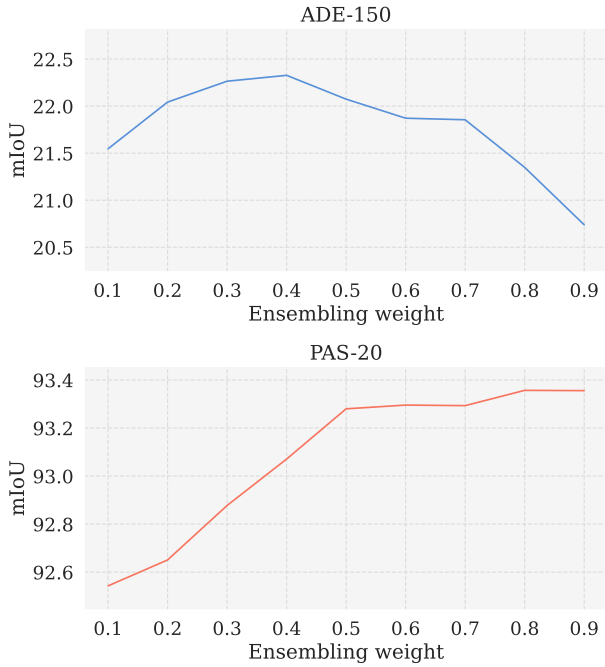

Figure 3.3: Ablation on different values of the ensembling weight α.

the quality of the vocabulary employed. Factors such as the number of samples collected for a specific word, the accuracy of matching regions with words, the distribution of the embeddings in the feature space, and their representativeness of the semantic concept all play significant roles. These observations emphasize the need for an adaptation phase specific to the set of arbitrary classes, by tuning the value of the ensemble weight to obtain the best performance.

Number of Reference Prototypes. Figure 3.4 illustrates the trend of the mIoU as the number of clusters k in the k-means algorithm increases. We observe that the mIoU reaches its peak at $k = 10$ for both datasets and shows a tendency to stabilize as k further increases. The variation in mIoU can be attributed to the frequency of word occurrences in the captions. We theorize that as k increases, the noise incorporated in the reference embeddings also increases. On the other hand, when using a small value of k, the variety of representations offered by

[1]OpenSeg uses ALIGN as the pre-trained vision-language model instead of CLIP.

Table 3.3: Comparison with other state-of-the-art two-stage models.

Method	Training Dataset	Frozen CLIP	Similarity Text	Similarity Visual	PAS 20	ADE 150	ADE 847	PC 59	PC 459
GroupViT [363]	GCC+YFCC	✓	✓	✗	52.3	-	-	22.4	-
ZegFormer [88]	COCO-Stuff-156	✓	✓	✗	80.7	16.4	-	-	-
OpenSeg [136] (R-101) [1]	COCO Panoptic	✗	✓	✗	60.0	15.3	4.0	36.9	6.5
ZSSeg [368] (R-101)	COCO-Stuff-171	✗	✓	✗	88.4	20.5	7.0	47.7	-
OVSeg [187] (R-101)	COCO-Stuff-171	✗	✓	✗	89.2	24.8	7.1	53.3	11.0
OVSeg [187] (Swin-B)	COCO-Stuff-171	✗	✓	✗	94.5	29.6	9.0	55.7	12.4
VOCSeg	COCO-Stuff-171	✓	✓	✓	93.4	22.5	8.1	47.3	10.8

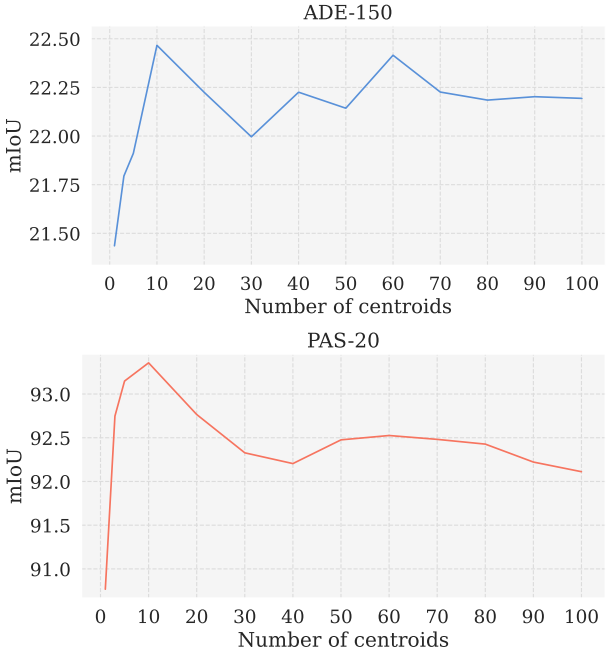

Figure 3.4: Ablation on the number of clusters used in the k-means algorithm.

the vocabulary becomes limited. This limitation hampers the ability to embed different visual concepts under the same word, leading to decreased performance in capturing the multitude of nuances in the objects.

3.1.2.3 Comparison with the State of the Art

We conduct a comparison with the following open-vocabulary architectures: GroupViT [363], ZegFormer [88], OpenSeg [136], ZSSeg [368], and OVSeg [187]. The results can be observed in Table 4.1. The *Similarity* column highlights the uniqueness of our approach in leveraging the similarities between image embeddings to bridge the gap between the images used to train CLIP and the regions extracted in two-stage approaches. Despite introducing a pre-processing step without additional parameters or fine-tuning CLIP, our method outperforms ZSSeg, which utilizes learnable tokens in the textual prompts, on both the ADE-150 and ADE-847 settings by +2 and +1.1 mIoU, respectively, and on PAS-20 by 5 mIoU. It also surpasses OpenSeg on all benchmark datasets, obtaining a +7.2 on ADE-150, +4.1 on ADE-847, +23.4 on PAS-20, +10.4 on P-59, and +4.3 on P-459. Furthermore, it outperforms OVSeg with a ResNet-101 backbone on ADE-150 by +4.2 and ADE-847 by +1.0. These architectures achieve high performance through fine-tuning or learnable tokens on a limited set of annotated segmentation data, which limits their generalization ability. In contrast, our method provides comparable results while allowing the extension of the visual vocabulary without compromising the quality of previously collected prototypes. Moreover, our VOCSeg largely outperforms ZegFormer and GroupViT, which operate in the same setting (*i.e.*, without fine-tuning CLIP). Our best performance is achieved using $k = 10$ in the k-means algorithm, $N = 10$ in the k-nearest neighbors algorithm, α equal to 0.8, 0.35, 0.2, 0.9, and 0.1 on, respectively, PAS-20, ADE-150, ADE-847, PAS-59 and PAS-459, and β equal to 0.7 on ADE-150 and ADE-847, and 0.6 on PAS-20, PAS-59, and PAS-459.

3.2 FOSSIL

In this Section, we introduce a training-free pipeline for open-vocabulary semantic segmentation, named FOSSIL, that creates a synthetic collection of visual references from a large set of captions using diffusion models and retrieves the reference corresponding to the input text to perform prototype-based segmentation. Along with FOSSIL, we present OpenCut, a mask proposer approach that iteratively bipartitions the features obtained with a self-supervised visual backbone to produce high-quality masks for both foreground and background regions.

3.2.1 Method

3.2.1.1 Preliminaries

Task definition. Open-Vocabulary Semantic Segmentation aims at segmenting an image $I \in \mathbb{R}^{H \times W \times 3}$ according to a set of arbitrary concepts $c_i \in \mathcal{C}$ described by text. The majority of the other weakly-supervised and training-free approaches [39, 165, 166, 227, 328, 379, 422] tackle the task by associating features extracted from a visual encoder $\Phi_v(I) : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H' \times W' \times D}$ to those extracted from a reference encoder $\Phi_r(c_i) : \mathbb{R}^{d_t} \rightarrow \mathbb{R}^D$, exploiting a similarity function (*e.g.* cosine similarity), so that visual and textual features are treated as lying in a shared multi-modal space. However, a single textual embedding is not sufficient to represent the intra-class variability in the visual appearances of a given concept. Moreover, individually classifying pixel-level features produces noisy semantic regions, especially along borders, whose coherence with the underlying visual object is not guaranteed.

Overview of our approach. To address these weaknesses, we decouple the task into two phases: grouping pixels into visually coherent regions and associating a concept from the set $\mathcal{C}$ to each region. A region proposer in an open-vocabulary setting should be able to detect regions based mostly on visual appearance to maintain a good quality across a large range of concepts. To tackle this challenge, we introduce OpenCut, which aims to iteratively apply MaskCut [339] by varying the threshold τ to accurately detect foreground objects and then the background.

To assign an arbitrary concept to each region, we propose FOSSIL, an architecture that exploits a collection composed of pairs of synthetic Visual Reference Embeddings and Retrieval Textual Embeddings that can be retrieved through an arbitrary textual query and be used to compute similarity against regions in the unimodal visual space. The synthetic visual references are created by providing a large set of captions to Stable Diffusion [266] and, for each noun in the captions, by extracting the corresponding heatmap on the generated image I_g. These heatmaps are binarized to obtain a mask for each noun. The generated image is processed with the visual encoder Φ_v to obtain its dense features, and, for each word, a region pooling on the corresponding binary mask is performed to produce a representative Visual Reference Embedding. At the same time, for each noun, a text encoder Φ_t is applied on a pre-determined prompt template in which the noun is inserted, and to the caption. Then, the two resulting feature vectors are linearly combined to produce the Retrieval Textual Embedding.

At inference time, the set of textual arbitrary concepts is embedded with

the text encoder Φ_t and is used to retrieve the N most similar Textual Retrieval Embeddings. The corresponding Visual Reference Embeddings are clustered to obtain a set of K prototypes. The input image is encoded with the visual encoder Φ_v to obtain its dense features, and a set of region proposals is produced using OpenCut. Then, for each region, we perform region pooling on the dense features to create a unique feature vector for that region. Thus, we compute the similarity between the feature vector and the prototypes to assign the most similar concept to the pixel covered by that region.

3.2.1.2 Reference Collection Generation

Our objective is to enable a self-supervised visual backbone to perform open-vocabulary semantic segmentation on an image, given a set of free-form arbitrary texts. To achieve so, we want to create a collection of pairs composed of a Visual Reference Embedding, in the backbone space, and a Textual Retrieval Embedding, in the space of a text encoder, for a vast vocabulary from segmentation data. These pairs would couple the visual aspect described by the dense features of the backbone to the corresponding label. However, manually annotated datasets do not cover a vast set of terms and expressions due to the expensive costs of annotating. Hence, we propose to exploit a large web-crawled set of captions that we provide to Stable Diffusion to generate a collection of synthetic images. Thus, we parse nouns from the captions to extract their corresponding heatmap on the generated image through the cross-attention mechanism proposed in DAAM [304]. Since these heatmaps often present peaks on particularly significant portions of the image (*e.g.*, eyes for animals, faces for humans), we apply the sigmoid function on the values of a heatmap followed by a threshold to flatten it. The resulting binary mask can be used to compute the average of the dense features covered by the mask. This produces a Visual Reference Embedding, namely a representative feature vector for the region corresponding to the parsed noun in the caption.

A straightforward approach to building the Textual Retrieval Embedding would be to insert the parsed word in some pre-determined prompt templates, encode them with the text encoder, and compute their average. Nevertheless, this would allow us to only build Textual Retrieval Embeddings for single words, whereas the input text can correspond to any textual description. Hence, we propose to combine the average feature vector of the pre-determined templates with the feature vector of the entire caption. This method moves the vector corresponding to that word towards the real context in which it has been found. Finally, we create an efficient retrieval index on the whole set of collected Textual Retrieval Embeddings.

3.2.1.3 Prototype Creation

Given an arbitrary text, we embed the same pre-determined templates used when creating the Textual Retrieval Embeddings using the text encoder to retrieve the N most similar Textual Retrieval Embeddings using the cosine similarity. At inference time, we could interpret the corresponding N Visual Reference Embeddings as prototypes. However, a low value of N risks to be not sufficient to represent the concept, whereas a large N would add outliers that diminish the robustness of the segmentation. So, we cluster the N Visual Reference Embedding through K-Means and we consider the resulting K centroids as prototypes to obtain a trade-off between robustness and representativeness.

3.2.1.4 OpenCut

Open-vocabulary segmentation approaches that only leverage pixel-level similarities without considering more high-level perspectives can produce noisy segmented regions, especially along borders. When multiple objects in a scene come close, indeed, the features along their border embed clues related to multiple semantic elements. On the other hand, implementing region proposal methods in an open-vocabulary setting presents significant threats, as generating high-quality masks for a wide range of concepts requires dense supervision. Following this insight, we propose an extension of the normalize cut algorithm [284] to provide training-free mask proposals based on the extraction of dense features from a self-supervised backbone.

Preliminaries. Given a dense feature map $\Phi_v(I) \in \mathbb{R}^{H' \times W' \times D}$, normalized cut builds a fully-connected undirected graph in which each feature vector corresponds to a node, and adds an edge between each pair of features with a weight corresponding to their cosine similarity. A threshold τ is applied on the resulting similarity matrix W to obtain $\bar{W}$ such as

$$\bar{W}_{ij} = \begin{cases} 1e^{-5} & \text{where } W_{ij} < \tau \\ 1 & \text{where } W_{ij} \geq \tau, \end{cases} \tag{3.6}$$

to binarize and enhance similarities scores. Then, the graph is split into two disjoint graphs by minimizing the energy of the resulting sub-graphs. This operation corresponds to solving the following generalized eigenvalue system

$$(D - \bar{W})x = \lambda x D, \tag{3.7}$$

and considering the eigenvector x relative to the second smallest eigenvalue λ. In the above equation, D is a diagonal matrix with size $N \times N$ and with $D_{ii} = \sum_j \bar{W}_{ij}$, while $\bar{W}$ is a symmetric matrix with size $N \times N$.

As the resulting eigenvector can be interpreted as a heatmap, we obtain two complementary binary masks by thresholding the eigenvector on its mean value, as

$$M_{ij}^t = \begin{cases} 1 & x_{ij} > \text{mean}(x) \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

Following MaskCut [339], we first heuristically label the binary mask that contains the patch corresponding to the maximum absolute value in the eigenvector as the foreground mask. Then, in order to obtain the mask relative to the next object, we update the weight matrix by setting to zero the features vector of the nodes corresponding to the current foreground mask and recomputing the weight matrix. This process is repeated until a maximum of t times to detect multiple objects. Also, the procedure is stopped when the thresholded weight matrix is composed of either all 1s or $1e^{-5}$s.

OpenCut. The value of the threshold τ on the weight matrix is determinant in selecting masks with the norrmalized cut algorithm, and it is strongly correlated with the structure of the feature space of the visual backbone. Indeed, when considering DINO [37] as backbone, negative or close to 1 values of τ tend to produce masks on background regions rather than foreground objects. In our proposed OpenCut, we leverage this behavior to extract a set of masks so that the majority of pixels in the image are covered. To accomplish this, we iteratively apply the MaskCut algorithm for a set of values of τ, and for each τ we extract a maximum of t masks. The set of chosen τ serves to first identify masks corresponding to foreground objects, refine these masks during iterations, and finally identify the background masks.

Mask refinement. Since each resulting mask is associated with a bipartition of the graph, it is not guaranteed that the mask corresponds to a single region of the image. Hence, we split each binary mask into a mask for each of its connected components. Components that are composed of a number of pixels under a threshold η are discarded to remove noisy regions. During iterations, for each new mask, we check whether it does not present an overlap, measured as Intersection over Union, that is larger than a hyper-parameter μ with a previous mask. If so, it is likely that the two masks correspond to the same object but focus on different parts, and hence we merge them. Moreover, for each new mask, we check whether its surface is not covered for more than a value ρ by the union of

the previously extracted masks. If so, we discard the new mask. Then, for each accepted new mask we remove the pixels that are already covered by previously extracted masks. These mechanisms allow us to keep only significant masks and that each pixel is covered by at most a unique mask (*i.e.*, masks are mutually exclusive). There might be uncovered pixels that require to be handled by the segmentation model.

3.2.1.5 Inference protocol

Given an image I and a set of arbitrary concepts $\mathcal{C}$ described by texts, we extract the dense features $\Phi_v(I)$ of the image through the visual backbone, and a set of L mutually exclusive binary masks $M_l \in \mathbf{R}^{H \times W},\ l = 1 \dots L$ using the proposed OpenCut approach. Further, we leverage the procedure described in Section 3.2.1.3 to obtain a set of K visual prototypes for each input text. For a binary mask M_l, we upsample it at the resolution $H' \times W'$ of the dense features through bilinear interpolation, and perform a mean-region pooling to construct the region embedding $v_{Rl} \in \mathbf{R}^D$:

$$v_{Rl} = \frac{\sum_{i=1}^{H'} \sum_{j=1}^{W'} M_{lij} \Phi(I)_{ij}}{\sum_{i=1}^{H'} \sum_{j=1}^{W'} M_{lij}}. \tag{3.9}$$

For a given pixel that is not covered by masks from OpenCut, we consider the patch that covers it as its corresponding mask, with the dense feature vector associated with that patch as its region embedding. For each region embedding v_{Rl}, we compute the cosine similarity against the K prototypes $v_{Pk},\ k = 1 \dots K$ of each concept c_a in $\mathcal{C}$, followed by a sigmoid. Finally, we consider the resulting similarity between a region and a concept as the ensembling between the mean of the K similarities against the prototypes and the maximum of them, as follows

$$\bar{s}(v_{Rl}, c_a) = (1 - \gamma) \frac{\sum_{k=1}^{K} s(v_{Rl}, v_{Pk})}{K} + \gamma \max_{k=1}^{K} s(v_{Rl}, v_{Pk}), \tag{3.10}$$

where γ is a weighting hyper-parameter.

3.2.2 Experiments

3.2.2.1 Experimental Setup

Implementation Details. For the generation of the Reference Collection, we use all 5 captions per image from COCO Captions [55], Stable Diffusion [266]

Method	Training Dataset	Support Dataset	Similarity		VOC	Context	COCOStuff	ADE	Cityscapes
			Textual	Visual					
GroupViT [363]	CC12M [41] + RedCaps [84]	-	✓	✗	79.7	23.4	15.3	9.2	11.1
MaskCLIP [422]	-	-	✓	✗	74.9	26.4	16.4	9.8	12.6
ReCo [286]	-	ImageNet1K [270]	✗	✓	57.7	22.3	14.8	11.2	21.1
TCL [39]	CC3M [281] + CC12M [41]	-	✓	✗	77.5	30.3	19.6	14.9	23.1
OVDiff [147]	-	-	✗	✓	81.7	33.7	-	14.9	-
FOSSIL	-	COCO Captions [55]	✗	✓	**81.8**	**35.8**	**24.8**	**18.8**	**23.2**

Table 3.4: Comparison with other state-of-the-art unsupervised open-vocabulary semantic segmentation models under the mIoU metric. In "Support Dataset" we report datasets used to create a collection of references. In "Similarity" we report whether the model exploits similarity in a multi-modal embedding space or the unimodal visual space.

2.1 base with 50 diffusion steps and a threshold equal to 0.45 to binarize the heatmaps extracted through DAAM [304]. As visual encoder we use DINOv2 ViT-L/14 [234] on images resized to 518 $\times$ 518, producing dense features $\Phi_v(I) \in \mathbb{R}^{37\times37\times1024}$. As the text encoder, we use CLIP [249] ViT-L/14 [92] and the set of 7 prompt templates proposed in [249] for zero-shot classification. For building the Textual Retrieval Embedding, we adopt a weight equal to 0.9 for the word in the templates and 0.1 for the caption. For MaskCut, we use the hyperparameters proposed in [339]: three stages t on images resized to 480 $\times$ 480 pixels, keys from the last attention layer of a DINO [37] ViT-B/8 backbone as dense self-supervised features, and Conditional Random Field [158] to post-process masks. For mask refinement during the iterations of OpenCut, we use η equal to 16, μ equal to 0.8, and ρ equal to 0.7. We use the faiss library [139] for both efficient retrieval and clustering.

Evaluation Protocol. We follow the unsupervised open-vocabulary semantic segmentation evaluation protocol proposed by Cha *et al.* [39]. We use the class names from the default version of MMSegmentation [70] without other modifications. We resize the input image to have a short side of 448 and employ a sliding window approach with a stride of 224 pixels. We use mean Intersection-over-Union (mIoU) to assess the segmentation performance.

3.2.2.2 Comparison with the State of the Art

In Table 4.1 we compare FOSSIL with prior works under the same evaluation protocol: GroupViT [363], MaskCLIP [422], ReCo [286], TCL [39], and OVDiff [147]. As can be seen, our proposal largely outperforms the other methods in all settings, thus confirming the appropriateness of the proposed strategies. Noticeably, our training-free approach outperforms the performances of approaches that employ larger support datasets and which employ extensive training data.

When comparing across different datasets, we observe a larger margin of improvement on COCOStuff and ADE, respectively, of 5.2 and 3.9 mIoU points. Noticeably, these two datasets are the ones with the largest number of classes, respectively 171 and 150, thus underlying the ability of FOSSIL when dealing with a higher number of semantic classes. Further, this also shows that our method is able to maintain excellent recognition abilities when the number of prototypes in the visual backbone space grows.

Overall, our results show that the contribution provided by the ability to localize concepts generated with Stable Diffusion through DAAM to extract representative dense features largely compensates for the domain shift introduced

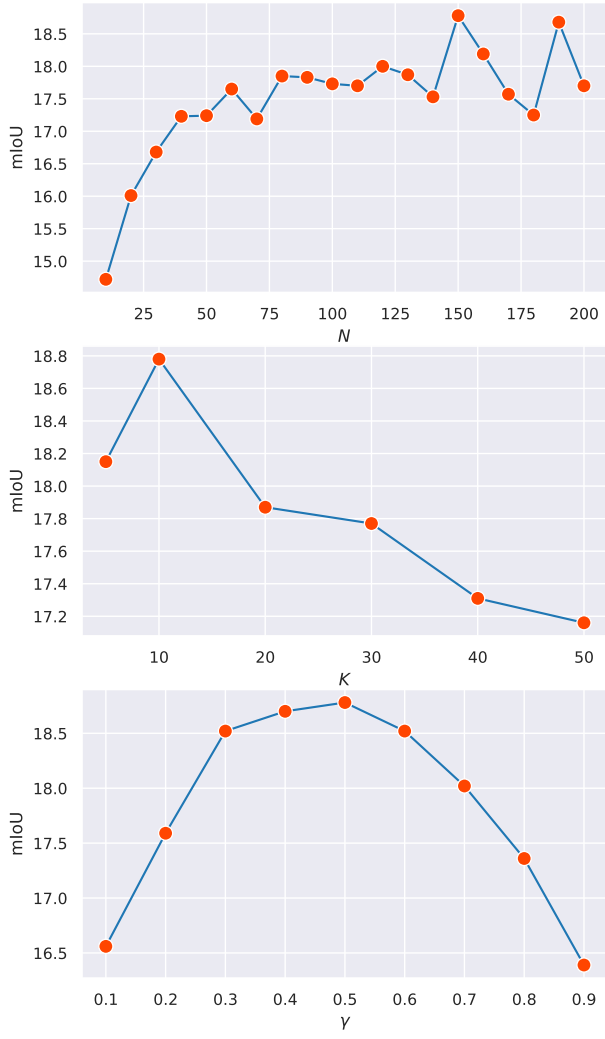

Figure 3.5: Ablation study on the three inference hyper-parameters N, K and γ on ADE. We test each parameter starting from our best configuration, with $N = 150$, $K = 10$ and $\gamma = 0.5$.

Table 3.5: Ablation on different visual backbones considering our best configuration on ADE, in terms of mIoU score.

Visual Backbone	Architecture	ADE
MAE [122]	ViT-L/14	2.0
DINO [37]	ViT-B/8	12.0
DINOv2 [234]	ViT-L/14	**18.8**

between synthetic and real images, also due to the high quality reached by diffusion models. Hence, this research direction is proving to be more promising than learning a pixel-level alignment from real images which do not provide locality information for a large vocabulary.

3.2.2.3 Ablation Studies

After comparing with other state-of-the-art approaches, we also provide two ablation studies, so as to assess the role of different components of our approach. In particular, we investigate the role of the visual backbone selection and the sensitivity to hyperparameters.

Visual Backbone choice. As the proposed approach is backbone-agnostic, any self-supervised backbone can be employed to build visual prototypes and extract dense features at inference time. To showcase this, and prove the effectiveness of different self-supervised backbones, in Table 3.5 we report an ablation study on performance obtained on the ADE benchmark with three different backbones: MAE ViT-L/14 [122], DINO ViT-B/8 [37] and DINOv2 ViT-L/14 [234].

As it can be observed, MAE presents a considerably lower performance with respect to the other two backbones, confirming the appropriateness of employing DINO-based backbones. We hypothesize that this is due to the fact that features learned with MAEs have limited semantic coherence when encoding the same concept across different images. When comparing the two considered DINO-based backbones, instead, we notice that DINOv2 largely outperforms DINO, due to a larger architecture (ViT-L/14 instead of ViT-B/8) and to its improved training strategy (as reported in [234]). In the following, we will consider DINOv2 for all the other experiments.

Hyperparameter choice. While being training-free, our approach relies on three main hyperparameters. Depending on the visual context and the distribution of classes to be detected, these can significantly influence the inference performance and therefore require to be accurately tuned. In particular, these are as follows:

Benchmark	# classes	N	K	γ
VOC	20	90	1	-
Context	59	70	10	0.7
COCOStuff	171	95	25	0.5
ADE	150	150	10	0.5
Cityscapes	19	95	30	0.55

Table 3.6: Our best configurations of the hyper-parameters when evaluating each of the 5 benchmarks.

- the number of Textual Retrieval Embedding and Visual Reference Embedding pairs that are retrieved for each arbitrary concept (N);
- the number of visual prototypes obtained as centroids of the K-Means algorithm, applied on the retrieved Visual Reference Embeddings (K);
- the weight attributed to the maximum of the similarities against the set of prototypes of an arbitrary concept, with respect to the mean on that similarities, when computing the concept assigned to a region (γ).

To show that different hyperparameter values can be chosen to obtain a better performance, in Table 3.6 we report the best configurations obtained on each benchmark. While we did not observe a clear relation between these hyper-parameters and the raw number of classes in a dataset, we hypothesize that their optimal values depend also on other factors, such as the semantic relation in the set of classes and their semantic variance. For instance, in Cityscapes, where classes belong to the urban street domain, a large value of γ may lead to assigning outliers introduced during the creation of the Reference Collection, thus resulting in better performance.

Moreover, in Figure 3.5 we report an ablation study about hyper-parameters N, K, and γ on the ADE benchmark. We vary each parameter one at a time starting from the best configuration. We observe that the parameter that influences performance the most is N, in particular for low values that represent an insufficient amount of references to capture the variance in visual appearances for that class.

3.2.2.4 Qualitative results

To complement our evaluation, in Figure 4.3 we provide a qualitative visualization of the segmentation masks obtained by FOSSIL, on images from the ADE dataset. Here, we also ablate our approach by removing the OpenCut mask proposals and

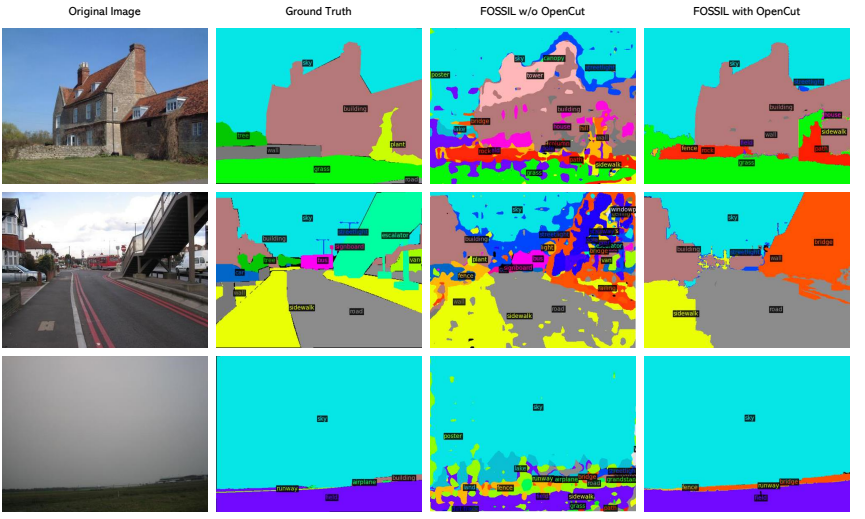

Figure 3.6: Qualitative results, comparing FOSSIL with and without the OpenCut component.

comparing them with our full pipeline, to showcase the role of OpenCut in the final segmentation quality. We firstly observe that FOSSIL is capable of properly segmenting all objects on the scene, assigning them to the correct semantic class, and providing curated segmentation masks that properly align with ground-truth borders. Further, comparing the last two columns of the Figure, the role of the OpenCut component can be clearly observed. As it can be seen, indeed, the mask proposals provided by OpenCut have a high degree of quality, and their adoption results in a cleaner and significantly less noisy result.

3.3 FreeDA

In this Section, we introduce FreeDA, a training-free method for open-vocabulary semantic segmentation based on the generation of context-aware textual-visual reference embeddings through diffusion models. These embeddings are retrieved in the inference pipeline that, leveraging the semantic correspondence of DINOv2 [234], superpixel algorithms, and a combination of local and global similarities, achieves precise and robust segmentation prediction, as shown in

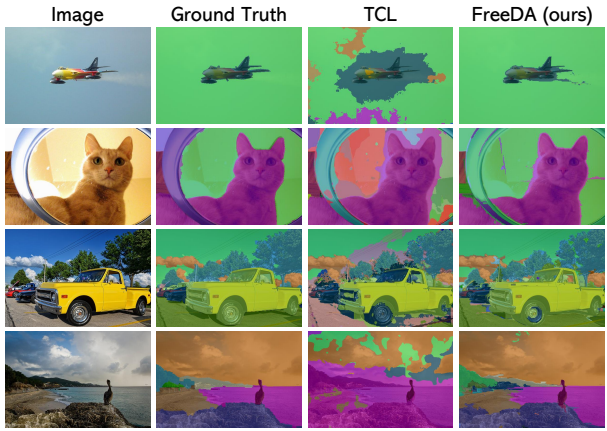

Figure 3.7: Open-vocabulary segmentation with: (a) TCL [39], which performs end-to-end learning of region-text alignment; (b) our FreeDA, which leverages generated textual-visual embeddings with global-local similarities and does not require any training.

Fig. 3.7.

3.3.1 Method

In FreeDA, we decouple the open-vocabulary semantic segmentation task into two phases: a *diffusion-augmented prototype generation* phase, which is carried out in an offline manner (visually represented in Figure 3.8), and a *semantic correspondence-based inference* stage, which is employed at test time to perform prediction over an input image. This second stage is visually depicted in Figure 3.9.

3.3.1.1 Diffusion-Augmented Prototype Generation

During the pre-processing phase, we collect a large set of visual prototypes and corresponding textual key embedding vectors, which describe semantic instances along with their textual and visual contexts. A textual key represents a semantic category and its textual context as described in a caption. A visual prototype, instead, describes an instance of that semantic category contextualized in an image. Collections of prototypes belonging to the same semantic class, thus, represent examples of the visual variety of that class.

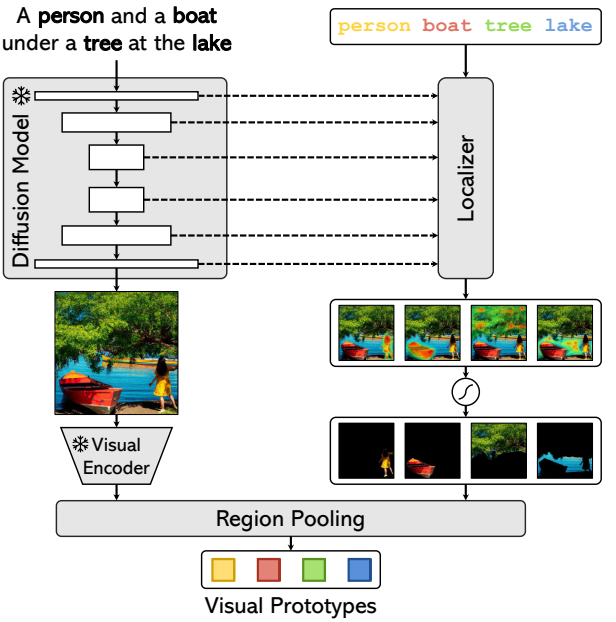

Figure 3.8: Overview of the diffusion-augmented prototype generation phase of FreeDA. Visual prototypes are generated by pooling self-supervised visual features on weak localization masks extracted from Stable Diffusion.

Extracting Localized Masks with Diffusion Models. As prototypes will be employed to predict semantic classes in a non-parametric way, it is crucial to build a large collection of prototypes with high semantic variance. To this aim, we follow FOSSIL (Sec. 3.2), and we generate a large set of real-world scenes using Stable Diffusion [266] starting from a large set of captions. Generating images rather than collecting real images from web-scale datasets allows us to control the resulting semantic distribution and its variance. Most importantly, also, latent-based diffusion models can predict the location of objects in the generated scene [304].

Diffusion models, indeed, map word embeddings of the conditioning text to the activations of their denoising subnetwork (*e.g.*, U-Net [266, 267]) through cross-attention layers applied at different scales. Cross-attention activations, therefore, relate each word of the conditioning caption to a portion of the image and can be employed to generate weak localization masks. As each layer of the de-

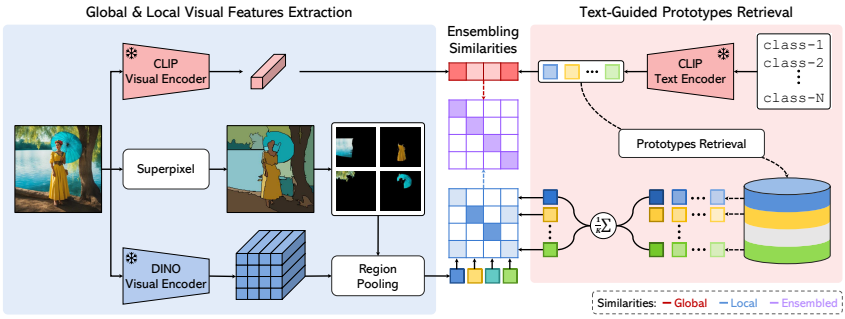

Figure 3.9: Overview of the inference process in FreeDA. Local (region-level) and global similarities are computed by employing, respectively, visual self-supervised and multimodal contrastive embedding spaces, and by comparing them with input texts and prototypes, built during the off-line stage.

noising network produces cross-attention maps at a different scale, we upscale all intermediate maps at the original image size. Then, we collapse across heads, layers, and diffusion time steps to obtain a single object mask.

Formally, the attribution map of a word w from the conditioning caption over a generated image I is expressed as

$$A(I, w) = \frac{1}{TLH} \sum_{t,l,h} \text{upsample}(\mathcal{A}(I, w)_{t,l,h}), \tag{3.11}$$

where $\mathcal{A}(I, w)$ indicates the collection of cross-attention maps with respect to the tokens of word w, and t, l, and h index diffusion time steps, denoising layers, cross-attention heads respectively. Finally, usample$(\cdot)$ denotes a bilinear interpolation operator.

With the aforementioned approach for building localized masks, we employ a set of captions, designed to describe real images, to condition Stable Diffusion [266] and generate the corresponding set of synthetic images. Through a noun parser [205], from each caption we also extract mentioned nouns $\{w_1, \ldots, w_N\}$ and obtain their corresponding attribution maps $A(I, w_i) \in \mathbb{R}^{H \times W}$ over the generated image. Then, we normalize the scores of the attribution maps in the range $[-1, 1]$, apply a sigmoid function, and binarize the result by thresholding it to a constant value γ. The output of this process is a weak localization mask $M(I, w_i) \in \{0, 1\}^{H \times W}$ for each noun w_i mentioned in the input caption.

Visual Prototypes Extraction. To encode the content of the aforementioned weak

localization masks, we adopt DINOv2 [234], which showcases good localization and semantic matching capabilities. Given a generated image $I \in \mathbb{R}^{H \times W \times 3}$, we extract its dense features $v(I) \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times d_v}$, where P is the input patch size of the backbone and d_v is the dimensionality of its embedding space. For every noun w_i in the sentence, we interpolate the weak localization mask $M(I, w_i)$ to the size of the dense features, obtaining a resized version of the localization mask $\hat{M}(I, w_i) \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P}}$. Then, we perform a region pooling operation to aggregate visual features over the localization mask, as follows:

$$p(I, w_i) = \frac{\sum_{h=0}^{\frac{H}{P}} \sum_{w=0}^{\frac{W}{P}} v(I)[h, w] \hat{M}(I, w_i)[h, w]}{\sum_{h=0}^{\frac{H}{P}} \sum_{w=0}^{\frac{W}{P}} \hat{M}(I, w_i)[h, w]}, \tag{3.12}$$

where square brackets indicate indexing over spatial axes. The resulting vector $p(I, w_i) \in \mathbb{R}^{D_v}$ is the *visual prototype* for the noun w_i extracted from the input image I, and is defined as the mean of the dense features covered by the corresponding binary mask. Prototypes built with this approach embed a visual descriptor of the corresponding word localized in a synthetic context, obtained from a real description.

Textual Keys Extraction. In addition to representing visual prototypes, we employ a text encoder to represent nouns in their lexical context. To this aim, we define a set of textual templates $\mathcal{T}$ (*e.g.*, `A photo of a [NOUN]`), and embed each noun in all templates. This results in a textual embedding for each template, $t_i(w) \in \mathbb{R}^{D_t}, i = 1 \dots, T$, where T is the number of templates. We define $\hat{t}(w) = \frac{\sum_{i=1}^{T} t_i(w)}{T}$ as the mean noun embedding, and then linearly interpolate with the full caption embedding $\hat{c}$ to also capture the global context of the entire scene. Specifically, the resulting textual key vector $k(c, w)$ for a word w taken from a caption c is then defined as

$$k(c, w) = \alpha \hat{t}(w) + (1 - \alpha)\hat{c}, \tag{3.13}$$

where $\alpha \in (0, 1)$ is a scalar weight. Similar to prototypes, keys obtained through this process represent nouns contextualized in the caption in which they have been extracted. As each textual key is associated with a visual prototype, the set of textual keys extracted from a dataset can be indexed via an approximate nearest neighbor search to efficiently retrieve visual prototypes given a textual query.

3.3.1.2 Training-Free Mask Prediction

At inference time, our goal is to query the keys of the pre-built collection index to retrieve their corresponding prototypes. Then, we employ these prototypes as references to segment the input image through semantic correspondence with both local and global features.

Retrieving Prototypes. Given a set of textual categories $\{c_1, \dots, c_S\}$, we consider the same set of templates $\mathcal{T}$ employed during textual keys computation and embed each category as $\hat{t}(c_i) = \frac{\sum_{j=1}^{T} t_j(c_i)}{T}$, where $t_j(c_i)$ is the text embedding of a template applied on a category. For each category c_i, we leverage $\hat{t}(c_i)$ to query the key embeddings of the pre-built collection index and retrieve the K most similar ones according to cosine similarity. Each key embedding corresponds to the combination of the text embeddings of both a noun and the caption in which the noun is mentioned, and is uniquely linked with a visual prototype. Hence, we compute a representative visual prototype for each category as the mean of retrieved prototypes. Formally,

$$\overline{p}(c_i) = \frac{\sum_{k=1}^{K} p_{ik}}{K}, \tag{3.14}$$

where $\{p_{ik}\}_{k=1}^{K}$ is the set of retrieved prototypes for the given category c_i.

Superpixel-based Local Regions. Once a visual representation of a class has been obtained through the aforementioned procedure, a straightforward solution to predict a segmentation mask for an image I would be computing the semantic correspondences (*i.e.*, cosine similarities) for each of its dense feature $v(I)$ against the representative prototypes of input categories $\overline{p}(I, c_i), \quad i = 1, \dots, S$, and interpolate the result to the original image size. However, such an approach would lead to noisy segmentation masks.

In particular, it has been observed that DINOv2 shows good matching properties across objects from different images, but lacks in recognizing shapes and boundaries [406]. Hence, we propose to exploit a superpixel algorithm (*i.e.*, the Felzenszwalb's algorithm [102]) to partition the image by grouping pixels into class-agnostic non-overlapping regions according to their visual appearances and positions.

Each superpixel can be interpreted as a binary mask $R \in \{0, 1\}^{H \times W}$ that is active on pixels belonging to it. Similar to the construction of visual prototypes, we interpolate each superpixel at the size of the dense features and perform a region pooling stage as defined in Eq. 3.12 to produce superpixel embeddings

$r_i \in \mathbb{R}^{D_v}, \quad i = 1, \dots, |R|$. Then, for each superpixel embedding, we compute the cosine similarity against the representative prototypes of the categories. We associate each pixel with the unique region that includes it and we refer to this similarity in the unimodal space of the visual backbone as *local similarity*.

Combining Local and Global Similarities. While retrieved prototypes are linked with text, their feature vectors show good local matching properties but weaker global semantic capabilities. As correctly classifying pixels from a semantic point of view is crucial in segmentation, we propose to combine the local similarities obtained at the superpixel level with a global similarity measure which refers to the entire image. We compute this in the multimodal space of a vision-language model (*i.e.*, CLIP [249]), which instead has good semantic classification capabilities.

Specifically, we embed the input image using the image encoder of CLIP to produce an image embedding $i(I) \in \mathbb{R}^{D_t}$. Then, we compute cosine similarities between the image embedding and all the category embeddings $\hat{t}(c_i), \quad i = 1, \dots, c_S$. Finally, we combine this global similarity with the single local similarities associated with class-agnostic regions. The final similarity between a local region and a semantic class is therefore computed as

$$s(r_j, c_i) = \beta l(r_j, c_i) + (1 - \beta) g(I, c_i), \tag{3.15}$$

where r_j indicates the local region, c_i the semantic class, and I the input image. Further, $l(r_j, c_i)$ is the local similarity between the region of interest and the class, and $g(I, c_i)$ is the global similarity extracted from CLIP space. To obtain the final segmentation mask, each region is then associated with the semantic class with the highest similarity.

3.3.2 Experiments

3.3.2.1 Experimental Setup

Datasets. We evaluate FreeDA on the validation splits of Pascal VOC 2012 [97], Pascal Context [224], COCO Stuff [31], Cityscapes [71], and ADE20K [420, 421]. In addition to these datasets, for which we do not consider pixels not belonging to any category, we also validate our method when considering them as part of an additional "unknown" class (also referred to as "background" class in the literature). For these experiments, we again employ Pascal VOC 2012 and Pascal Context, and also include the COCO Objects dataset [31], which is a variant of COCO-Stuff with 80 foreground categories on the same validation split. To assess the segmentation performance, we employ the mean Intersection-over-Union

(mIoU) on all the classes of each dataset. As in FOSSIL, we follow the unified evaluation protocol for unsupervised open-vocabulary semantic segmentation established by Cha *et al.* [39]. Specifically, we evaluate the model considering the class names from the default version of the `MMSegmentation` toolbox. We resize the images to have a shorter side equal to 448 and employ a sliding window approach with a stride of 224 pixels.

Implementation Details. Textual sentences used as input in our diffusion-augmented prototype generation pipeline are taken from the COCO Captions dataset [55, 190]. We consider all five captions available for each image, thus obtaining a large set of captions describing natural images that can be used as input for a diffusion-based generative architecture. It is worth noting that we do not utilize the images associated with these captions. To generate the collection of visual prototypes, we employ Stable Diffusion v2.1 [266] with 50 diffusion steps and a threshold γ equal to 0.45. The scalar weight α that combines the mean noun embeddings and caption embeddings to form keys is equal to 0.9.

We use DINOv2 [234] pre-trained on the LVD-142M dataset as the self-supervised visual backbone, using both the ViT-B/14 and the ViT/L-14 versions, with an input image size of 518×518. This leads to dense features with size corresponding to 37×37. We also employ CLIP [249] as the multimodal encoder using the original OpenAI weights, on top of the ViT-B/16 and ViT-L/14 architectures. We use the same CLIP model for both key embeddings and global similarity computation, so that *(i)* we embed the arbitrary categories at inference time just one time and *(ii)* we do not need to load two different text encoders into memory.

To extract superpixels, we use the Felzenszwalb's algorithm [102]. We build and leverage an efficient exact retrieval index through the `faiss` library [139] based on cosine similarity. We consider the number of retrieved prototypes K equal to 350 for all datasets and the ensembling weight β between local and global similarities equal to 0.8 for all benchmarks except for Pascal VOC for which we use β equal to 0.7.

Textual Templates. To encode through the CLIP text encoder both the nouns extracted during prototype generation and the input categories utilized at inference time, we employ the following set of templates $\mathcal{T}$, introduced in [249]:

```
itap of a {}.
a bad photo of the {}.
a origami {}.
a photo of the large {}.
a {} in a video game.
```

Model	PAMR	Dataset	Parameters (M)		Similarity		mIoU				
			Total	Trainable	Textual	Visual	VOC	Context	Stuff	Cityscapes	ADE
ReCo [286]	✗	ImageNet1k★	313.0	0.0	✗	✓	57.7	22.3	14.8	21.1	11.2
GroupViT [363]	✗	CC12M+RedCaps♦	55.8	55.8	✓	✗	79.7	23.4	15.3	11.1	9.2
MaskCLIP [422]	✗	-	291.0	0.0	✓	✗	74.9	26.4	16.4	12.6	9.8
TCL [39]	✗	CC3M+CC12M♦	178.3	21.7	✓	✗	77.5	30.3	19.6	23.1	14.9
OVDiff [147]	✗	-	1,226.4	0.0	✗	✓	81.7	33.7	-	-	14.9
MaskCLIP [422]	✓	-	291.0	0.0	✓	✗	72.1	25.3	15.1	11.2	9.0
ReCo [286]	✓	ImageNet1k★	313.0	0.0	✗	✓	62.4	24.7	16.3	22.8	12.4
GroupViT [363]	✓	CC12M+YFCC♦	55.8	55.8	✓	✗	81.5	23.8	15.4	11.6	9.4
TCL [39]	✓	CC3M+CC12M♦	178.3	21.7	✓	✗	83.2	33.9	22.4	24.0	17.1
FreeDA (ViT-B)	✗	COCO Captions★	236.1	0.0	✗	✓	**85.6** (+2.4)	**43.1** (+9.2)	**27.8** (+5.4)	**36.7** (+12.7)	**22.4** (+5.3)
FreeDA (ViT-L)	✗	COCO Captions★	732.0	0.0	✗	✓	**87.9** (+4.7)	**43.5** (+9.6)	**28.8** (+6.4)	**36.7** (+12.7)	**23.2** (+6.1)

Table 3.7: Comparison with state-of-the-art unsupervised open-vocabulary semantic segmentation models on Pascal VOC [97], Pascal Context [224], COCO Stuff [31], Cityscapes [71], and ADE20K [420, 421], without considering the unknown category. The markers ♦ and ★ refer, respectively, to datasets used for training and support only.

```
art of the {}.
a photo of the small {}.
```

As discussed in [249], these templates provide a powerful means of contextualizing textual input, making them particularly well-suited for our application in the context of prototype generation and inference.

Prototypes generation. The foundation of our prototype generation lies in the utilization of a dataset of images paired with captions. To ensure the reproducibility of our results, we detail the negative prompts employed during the generation of images with Stable Diffusion in Table 3.8. These negative prompts play a crucial role in guiding the generation process, aiming to produce prototypes that are realistic and high-quality. The prototypes generation is performed offline and requires around 5.2 sec for each COCO caption. During inference, computing a category embedding and performing prototypes retrieval takes around 10.8 ms and 12.9 ms for the Base and Large versions of FreeDA.

3.3.2.2 Comparison with the State of the Art

We first compare FreeDA with recent state-of-the-art approaches for unsupervised open-vocabulary semantic segmentation. Table 4.1 shows the results on the five benchmarks without the unknown category (*i.e.*, Pascal VOC, Pascal Context, COCO Stuff, Cityscapes, and ADE20K). We report the performance of two variants of our approach: one based on DINOv2 ViT-B/14 and CLIP ViT-B/16, and the other based on DINOv2 ViT-L/14 and CLIP ViT-L/14, respectively denoted as FreeDA (ViT-B) and FreeDA (ViT-L). For this comparison, since the usage of superpixels to improve the adherence of predictions on the image can be interpreted as a mask refinement step, we also report the performance of the considered competitors when using the Pixel-Adaptive Mask Refinement (PAMR) proposed in [9] to refine the final predictions. As it can be seen, both variants of our solution achieve the best results on all datasets, surpassing all the competitors by a consistent margin. Specifically, when comparing with methods without PAMR, FreeDA achieves an average improvement of 10.0 and 10.9 mIoU points with respect to TCL [39], respectively, for the ViT-B and ViT-L variants. This performance improvement is confirmed also when comparing FreeDA with PAMR-based approaches, leading to an average increase of 7.0 and 7.9 mIoU points compared to the best-performing method.

In Table 3.9, we instead report the results on the three segmentation datasets, namely Pascal VOC, Pascal Context, and COCO Object, used to validate the effect-

3d	*abstract*	*art*
asymmetric	*bad anatomy*	*bad art*
bad proportions	*blurry*	*canvas frame*
cartoon	*cartoonish*	*cgi*
cloned face	*colorless*	*computer graphic*
cropped	*cut off*	*deformed*
dehydrated	*digital*	*digital art*
disfigured	*doll*	*duplicate*
error	*extra arms*	*extra fingers*
extra legs	*extra limbs*	*fused fingers*
fuzzy	*grainy*	*graphic*
gross proportions	*inaccurate*	*jpeg artifacts*
long neck	*low quality*	*low-resolution*
lowres	*malformed limbs*	*misshaped*
missing arms	*missing legs*	*morbid*
mutant	*mutated*	*mutated hands*
mutation	*mutilated*	*octane*
out of focus	*out of frame*	*oversaturated*
photoshop	*poorly drawn face*	*poorly drawn hands*
render	*retro*	*signature*
text	*too many fingers*	*ugly*
unreal	*unreal engine*	*unrealistic*
username	*video game*	*watermark*
weird colors	*worst quality*	

Table 3.8: Negative prompts employed in Stable Diffusion during prototypes generation.

iveness of segmentation methods when also considering the additional "unknown" category. Following [363], we apply a threshold on the final similarities to detect pixels that do not belong to any of the provided input categories. In particular, we apply the threshold on the similarity values obtained after ensembling local and global similarities. For this experiment, we restrict the comparison to methods that do not employ specific techniques to take into account the background of the scene, but instead perform thresholding, as done in our case. Notably, FreeDA achieves the best results on all three benchmarks, surpassing both methods that do not employ any mask refinement stages and approaches that instead refine their predictions using PAMR [9]. In particular, FreeDA reaches 55.4, 38.3, and 37.4

Model	PAMR	Training Dataset	mIoU		
			VOC	Context	Object
GroupViT [363]	-	CC12M+RedCaps	50.4	18.7	27.5
MaskCLIP [422]	-	-	38.8	23.6	20.6
ReCo [286]	-	-	25.1	19.9	15.7
ViewCo [260]	-	CC12M+YFCC	52.4	23.0	23.5
SegCLIP [209]	-	CC3M+COCO Captions	52.6	24.7	26.5
TCL [39]	-	CC3M+CC12M	51.2	24.3	30.4
OVSegmentor [366]	-	CC4M	53.8	20.4	25.1
GroupViT [363]	✓	CC12M+YFCC	51.1	19.0	27.9
MaskCLIP [422]	✓	-	37.2	22.6	18.9
TCL [39]	✓	CC3M+CC12M	55.0	30.4	31.6
FreeDA (ViT-L)	-	-	**55.4** (+0.4)	**38.3** (+7.9)	**37.4** (+5.8)

Table 3.9: Comparison with state-of-the-art unsupervised open-vocabulary semantic segmentation models on the validation sets of Pascal VOC [97], Pascal Context [224], and COCO Object [31], when considering the additional unknown category.

mIoU points respectively on Pascal VOC, Pascal Context, and COCO Object, which correspond to an improvement of 0.4, 7.9, and 5.8 points with respect to the best method (*i.e.*, TCL [39] using PAMR as mask refinement technique).

These results highlight the effectiveness of our solution, which, despite being completely training-free, achieves a new state of the art for unsupervised open-vocabulary semantic segmentation on all eight considered benchmarks. Some qualitative results are shown in Figure 4.3.

3.3.2.3 Ablation Studies and Analyses

We then evaluate the contribution of each component employed in our final solution and the effectiveness of different backbones to extract visual and textual features.

Effect of Changing the Visual Backbone. We first consider the performance of our approach when using different visual backbones to compute local similarities. In particular, we evaluate DeiT-III [314] pre-trained for image classification on ImageNet1k and based on ViT-L/16, CLIP [249] in both its ViT-B/16 and ViT-L/14 versions, DINO [37] based on the ViT-B/16 architecture, and our final choice DINOv2 [234] using both the variant based on ViT-B/14 and the one based on ViT-L/14. Given that different input and patch sizes can lead to different output feature sizes, we resize all images to 518×518 when using visual backbones with a patch size of 14 and 592×592 when employing visual backbones with a patch

Backbone	Global Similarity	Superpixels	mIoU		
			VOC	Cityscapes	ADE
CLIP (ViT-B/16)	✗	✗	61.3	21.3	13.4
DINO (ViT-B/16)	✗	✗	34.2	26.0	9.5
DINOv2 (ViT-B/14)	✗	✗	75.6	34.4	20.7
DeiT-III (ViT-L/16)	✗	✗	54.8	21.8	11.4
CLIP (ViT-L/14)	✗	✗	45.9	20.0	11.4
DINOv2 (ViT-L/14)	✗	✗	70.2	33.2	19.5
DINO (ViT-B/16)	✓	✗	80.4	27.8	16.5
DINOv2 (ViT-B/14)	✓	✗	86.2	35.0	21.9
DINOv2 (ViT-L/14)	✓	✗	87.2	34.5	21.6
DINO (ViT-B/16)	✓	✓	81.1	29.8	17.3
DINOv2 (ViT-B/14)	✓	✓	87.0	36.6	**23.2**
DINOv2 (ViT-L/14)	✓	✓	**87.9**	**36.7**	**23.2**

Table 3.10: Ablation study results using different visual backbones and validating the contribution of the key components of our solution. Results are reported on the validation sets of Pascal VOC [97], Cityscapes [71], and ADE20K [420, 421].

Local Backbone	Textual/Global Backbone	mIoU		
		VOC	Cityscapes	ADE
DINO (ViT-B/16)	CLIP (ViT-B/16)	80.8	30.6	17.0
DINOv2 (ViT-B/14)	CLIP (ViT-B/16)	85.6	**36.7**	22.4
DINOv2 (ViT-L/14)	CLIP (ViT-B/16)	86.9	36.3	22.3
DINOv2 (ViT-L/14)	CLIP (ViT-L/14)	**87.9**	**36.7**	**23.2**

Table 3.11: Performance analysis when employing visual and textual backbones of different sizes.

size of 16, thus always having features with a spatial size equal to 37×37. To validate only the role of different visual backbones, we apply them without global similarities and without superpixels to extract mask proposals. When considering the variant without superpixels, we directly compute the local similarities on the dense features, and we interpolate them to the original image size.

Results are reported in the upper part of Table 3.10, using the CLIP ViT-L/14 model to extract textual features. As it can be noticed, DINOv2 exhibits the best performance among both architectures based on ViT-B and ViT-L, confirming the power of self-supervised features in this setting.

Adding Global Similarities and Superpixels. To evaluate the contribution of

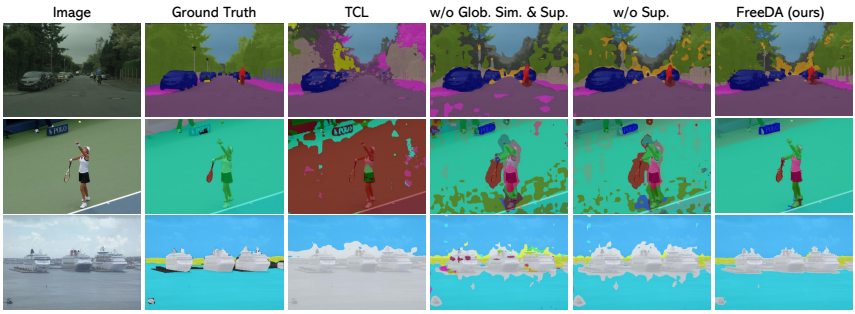

Figure 3.10: Qualitative results of FreeDA in comparison with TCL [39], with and without global similarities and superpixels.

global features and superpixel-based mask proposals, we report in the lower part of Table 3.10 the performance of FreeDA first adding only global similarities and then also including superpixels to extract mask proposals. Both strategies give a consistent contribution to the final performance, also when considering different visual backbones to compute local similarities. For example, when using DINOv2, global features bring an improvement of 0.9 mIoU points on the ADE20K dataset, while superpixels further enhance the final performance by an additional 1.6 mIoU points. Additionally, it is worth noting that the contribution of global similarities is more significant in Pascal VOC, where images are characterized by the presence of a single or a few objects occupying large areas of the scene, thus favoring global features instead of local ones.

Impact of Backbone Size. In Table 3.11, we investigate how much using a ViT-Large architecture to extract both visual and textual features increases the performance compared to a ViT-Base model. As also demonstrated by the complete results of the two variants of FreeDA reported in Table 4.1, this corresponds to around 2.3 mIoU points on Pascal VOC when employing DINOv2 to extract local features, while obtaining similar performance on Cityscapes and ADE20K.

Superpixel Algorithms and Prototype Aggregation Strategies. In Table 4.4, we instead validate the choice of employing Felzenszwalb's algorithm [102] to extract superpixels by comparing it with three widely adopted superpixel proposal algorithms, namely Watershed [129], SLIC [2], and SEEDS [319]. While different versions of superpixel algorithms lead to similar performance, the usage of Felzenszwalb's algorithm helps to further improve the results on all three datasets considered. In addition to comparing different superpixel extraction strategies, we also include the results obtained using PAMR [9] as a mask refinement method.

Model	Superpixels	mIoU		
		VOC	Cityscapes	ADE
w/ mean embedding (PAMR)	-	87.0	34.4	23.0
w/ mean embedding	Watershed	87.0	32.7	21.8
w/ mean embedding	SLIC	87.3	33.5	21.8
w/ mean embedding	SEEDS	87.5	32.3	22.4
w/ mean similarity	Felzenszwalb	79.5	29.3	18.8
w/ max similarity	Felzenszwalb	82.0	26.2	17.6
FreeDA (w/ mean embedding)	Felzenszwalb	**87.9**	**36.7**	**23.2**

Table 3.12: Performance analysis using different algorithms to compute superpixels and different prototypes aggregation strategies.

For this experiment, we first compute local similarities for dense features and ensemble them with the global similarity, then we apply PAMR to refine the resulting segmentation masks. Notably, employing superpixels to extract mask proposals leads to improved final results.

To validate the aggregation strategy used in FreeDA, in which we aggregate retrieved prototypes by computing their average embedding (*i.e.*, "mean embedding" in Table 4.4), we compare it with two different approaches based on first computing local similarities for all retrieved prototypes and then aggregating them by considering the mean or the maximum (*i.e.*, "mean similarity" and "max similarity"). Computing the average embedding of all retrieved prototypes brings the best results across all datasets.

Retrieval Performance Analysis. Finally, we analyze the performance when varying the retrieval parameters. Since our method leverages an exact retrieval index, we first validate how much using an approximate search impacts the performance. Specifically, the left plot of Figure 3.11 shows the trade-off between speed and performance when using a graph-based HNSW (Hierarchical Navigable Small World) index [214]. We report the CPU times to search the most similar $K = 350$ key embeddings when changing the depth of exploration in the index, and their corresponding mIoU scores. This parameter controls the size of the dynamic list of candidate nearest neighbors that are explored during the search process. On the right plot of Figure 3.11, we instead show the performance variation when changing the number K of searched keys. Results are reported on the ADE20K dataset. As it can be seen, using an approximate index only partially deteriorates the performance while consistently reducing the computing time. On the same

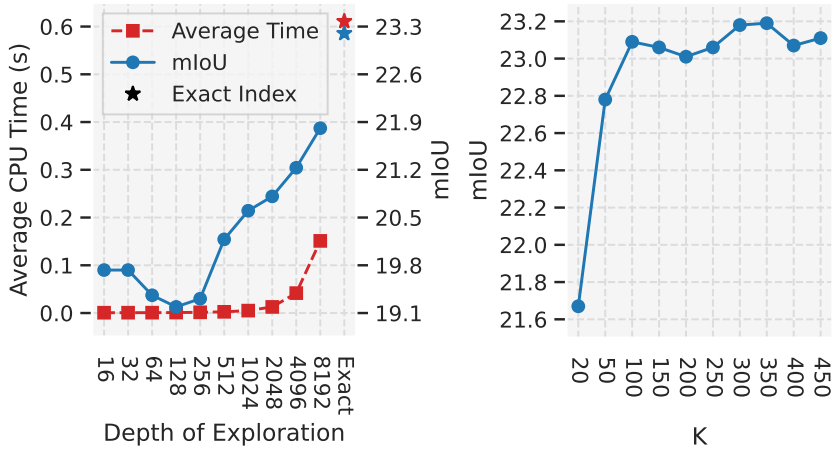

Figure 3.11: Retrieval results when using an approximate index (left) and varying the number of retrieved key-prototype pairs (right).

line, increasing the number of retrieved key embeddings does not improve the final performance, while retrieving a reduced number of items partially leads to lower results.

Effect of Superpixel Parameters. Felzenszwalb *et al.* [102] introduced an efficient superpixel algorithm that employs a graph-based approach. The algorithm initiates by constructing a graph representation of the image, where each pixel serves as a node, and edges connect neighboring pixels. Edge weights are determined based on the RGB color space differences between adjacent pixels. Consequently, connected components, initially established as individual components for each pixel, are progressively merged. The growth of each component is regulated by the scale of observation parameter k. The algorithm also incorporates two additional parameters: the diameter of the Gaussian filter used for pre-processing to enhance image smoothness and counter artifacts (σ), and the enforced minimum size of superpixels, μ. We employ the implementation of the `skimage`[2] library.

In Table 3.13, we report the parameter values employed on the examined datasets. Figure 3.12 further shows the performance variations obtained when altering these parameters on the ADE20K dataset [420, 421]. Notably, minor variations in these parameters have negligible effects on final performance. However, imposing

[2] `https://scikit-image.org/`

Dataset	μ	σ	k
Pascal VOC	100	0.7	20
Pascal Context	100	1.0	20
COCO Stuff	100	1.0	100
Cityscapes	50	0.5	20
ADE20K	100	1.0	20

Table 3.13: Parameters employed for Felzenszwalb's algorithm on each dataset.

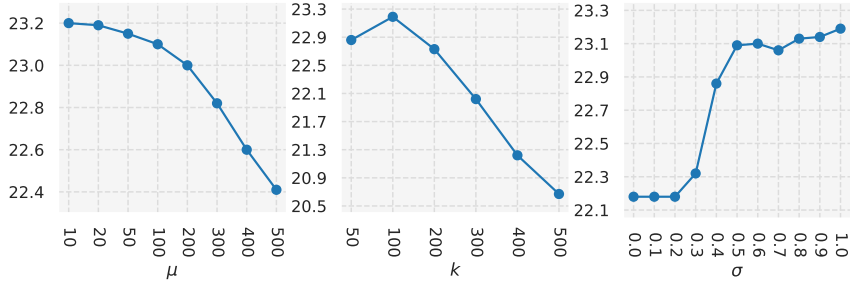

Figure 3.12: Effect of the variation of superpixel hyperparameters on ADE20K, measured in terms of mIoU.

large superpixels through a minimum size or scale of observation can significantly degrade the results.

Impact of caption context. In Sec. 3.3.1.1, we outline our methodology for extracting textual key embeddings. Specifically, we employ a linear combination of the word embedding $\hat{t}$ and the caption embedding $\hat{c}$, controlled by a parameter α. In our main results, we set α to 0.9 to effectively incorporate the textual context into the key embedding.

In Table 3.14, we conduct an ablation study on this choice. The case without caption context corresponds to setting α to 1. It is noteworthy that the inclusion of textual context proves to be particularly beneficial for input categories that consist of more than one word, such as `chest of drawers`. This scenario is prevalent in in-the-wild situations, thus emphasizing the practical utility of our approach in diverse and real-world settings.

Impact of unimodal global matching. In Table 3.15, we investigate the impact of employing DINOv2 for local and global matching. Since DINOv2 embeddings are not aligned with text, we compute global matching by using the similarity

	Caption Context	mIoU		
		Context	Stuff	ADE
	✗	43.1	27.4	22.2
FreeDA	✓	**43.5**	**28.8**	**23.2**

Table 3.14: Effect of full caption embeddings on the performance of key embeddings.

Local Backbone	Global Backbone	VOC	Cityscapes	ADE
DINOv2 (ViT-B/14)	DINOv2 (ViT-B/14)	78.4	30.7	17.8
DINOv2 (ViT-L/14)	DINOv2 (ViT-L/14)	74.4	33.5	20.3
DINOv2 (ViT-B/14)	CLIP (ViT-B/16)	85.6	**36.7**	22.4
DINOv2 (ViT-L/14)	CLIP (ViT-L/14)	**87.9**	**36.7**	**23.2**

Table 3.15: mIoU results with DINOv2 for local/global matching.

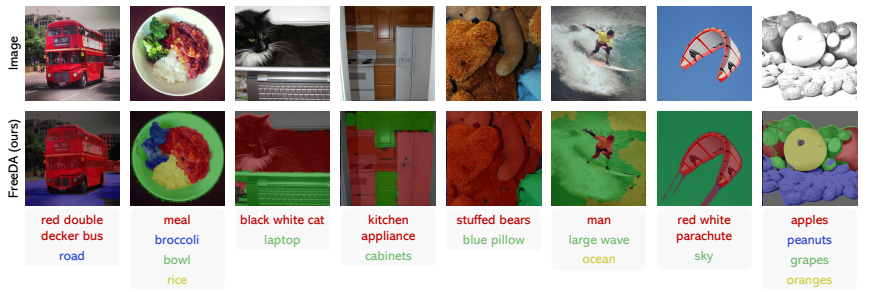

Figure 3.13: In-the-wild segmentation results obtained by prompting our model with diverse free-form textual inputs.

between the `CLS` token of DINOv2 and the representative visual prototypes of the categories. As can be observed, the usage of a text-aligned CLIP backbone improves performance with respect to the unimodal DINOv2 global features.

In-the-wild results. In Figure 3.13 we report a collection of in-the-wild examples obtained by prompting our model with diverse free-form textual inputs. Specifically, we extract noun chunks from sample captions of the COCO Captions validation set using the `spaCy`[3] NLP library. After removing stop-words, the noun chunks are utilized as input categories for segmenting the corresponding

[3] `https://spacy.io/`

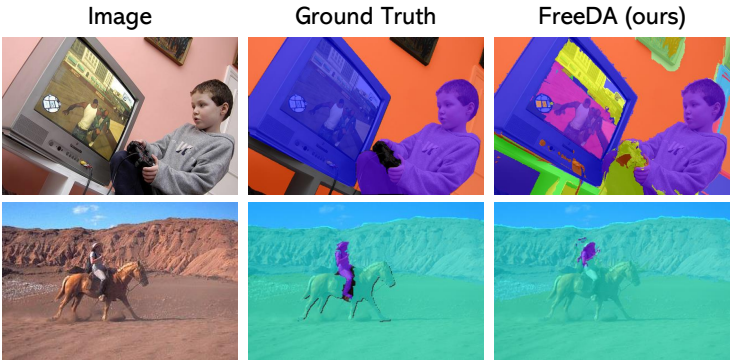

Figure 3.14: Sample failure cases.

images. These results extend our analysis beyond curated datasets and demonstrate the adaptability and robustness of our approach in handling real-world scenarios with varied and unstructured textual descriptions.

Failure cases. Finally, in Figure 3.14 we report sample scenarios in which our model encounters challenges and exhibits failure cases. The first row illustrates an image of a TV displaying a video game. Owing to the strong semantic correspondence properties at the token-level of DINOv2, our model tends to segment individual elements shown on the TV screen, thereby impacting the overall segmentation performance for the TV class. The second row of the figure instead presents another failure case featuring an image of a person atop a horse. However, the segmentation is incomplete and only partially captures the person. This limitation can be attributed to the prototypes corresponding to horses ridden by persons, whose noisy binarized masks include their legs. Overall, these failure cases shed light on areas where our model may struggle, emphasizing the need for further refinement and consideration of complex visual contexts.

Additional Qualitative Results. Figure 3.16 showcases additional qualitative results on Pascal VOC [97], Pascal Context [224], COCO Stuff [31], Cityscapes [71], and ADE20K [420, 421]. These qualitative samples offer a comprehensive view of the performance of our approach and highlight its versatility and effectiveness across a range of scenes and categories, reinforcing the applicability in various real-world scenarios.

3.3.2.4 Explainability

A notable advantage of our prototype-based approach lies in its inherent explainability, as the set of referring images used to generate prototypes can be visualized a posteriori. In our approach, in particular, we can visualize the generated images associated with the retrieved prototypes for a given input category, along with the corresponding attribution maps and binary masks.

Figure 3.15 illustrates the explainability capabilities of our solution, showcasing examples of retrieved prototypes for a specified category, highlighted within the captions in which the corresponding noun was mentioned. We further include the corresponding generated images, attribution maps, and binarized masks, providing a comprehensive view of the explainability achieved by our approach.

Figure 3.15: Examples of retrieved prototypes for a specified textual category. From left to right, we show the original COCO caption, the corresponding generated image, the attribution map, and the binarized mask (area highlighted in red).

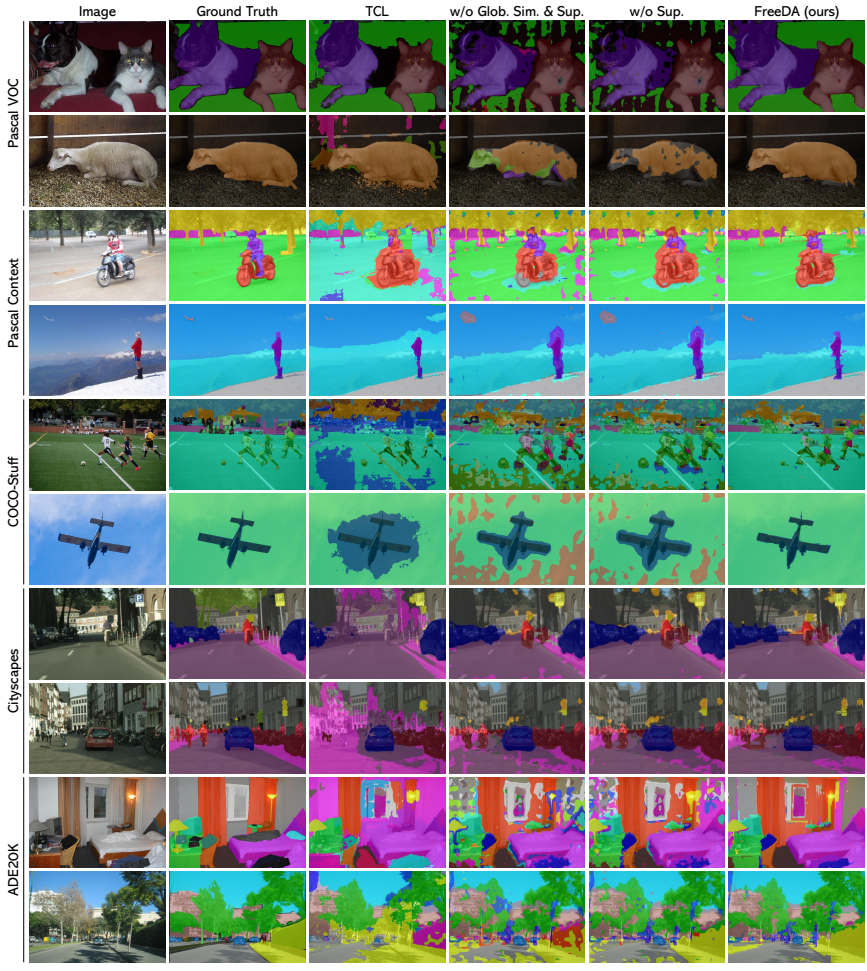

Figure 3.16: Additional qualitative results of FreeDA in comparison with TCL [39], with and without global similarities and superpixels.

Chapter 4

Open-Vocabulary Segmentation via Contrastive Learning

Multimodal pre-trained models like CLIP [249] have demonstrated impressive performance on tasks that demand a holistic understanding of vision and language modalities [419, 179, 220, 425], and have therefore been employed for unsupervised open-vocabulary semantic segmentation [422, 328, 166]. Although the CLIP-based backbones exhibit strong cross-modal capabilities, they are primarily trained to predict a global similarity score between text and images, which limits their spatial understanding and consequently affects tasks based on dense predictions. Several works have tackled this limitation by introducing modifications to the architecture of CLIP [422, 328, 116]. However, the spatial understanding constraints imposed by the training modality hinder the effectiveness of such backbones in open-vocabulary segmentation and highlight the potential benefits of exploring alternative models with enhanced perceptual capabilities.

On the other hand, self-supervised vision-only backbones like DINO and DINOv2 [37, 234, 77] have instead shown remarkable abilities in capturing fine-grained and localized spatial features without the reliance on annotated data. Specifically, the self-attention mechanism in such backbones generates attention maps that consistently pinpoint relevant regions within the image and has been widely leveraged for foreground object segmentation [289, 290, 341, 340, 339].

While this property makes them a powerful tool for tasks requiring fine-grained spatial understanding, the embedding space derived from visual self-supervised networks is not inherently aligned with textual concepts, making it incompatible with the open-vocabulary segmentation task.

In the previous Chapter, we explored an alternative path to open-vocabulary segmentation by introducing prototypical representations in the DINOv2 space. In particular, FOSSIL and FreeDA demonstrated that it is possible to construct and retrieve category-specific prototypes aligned with DINOv2 features, effectively bridging the visual and textual modalities without explicit cross-modal supervision. These results suggest that DINOv2, despite being trained without language supervision, implicitly supports a concept organization that can interface with textual categories through appropriate representations. Building on this insight, this Chapter investigates whether it is possible to directly align DINOv2 with a text encoder through contrastive learning, enabling open-vocabulary segmentation in a fully frozen backbone setting.

4.1 Talking to DINO

In this Section, we introduce Talk2DINO, the first model to equip DINOv2 with language capabilities via a lightweight non-linear warping function that maps CLIP text embeddings into the DINOv2 feature space. Talk2DINO proposes a novel training strategy that automatically selects the most semantically relevant visual self-attention head and performs alignment without fine-tuning either the vision or language backbones. Furthermore, we propose a new mechanism based on DINOv2 self-attention to enhance foreground–background separation, further improving segmentation performance in the open-vocabulary setting where the categories are only foreground objects.

4.1.1 Method

4.1.1.1 Preliminaries

Task Definition. Let $I \in \mathbb{R}^{H \times W \times 3}$ be an image and $v(I) \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times D_v}$ its dense feature map extracted by a Transformer-based visual backbone with input patch size P and dimensionality of embedding space D_v. Let $\{T_j\}_{j=1,\dots,M}$ be a set of arbitrary textual categories and $t(T_j) \in \mathbb{R}^{D_t}$ their embeddings extracted by a pre-trained textual backbone. To simplify the notation, in the following, we will refer to $v(I)$ as v and to $t(T_j)$ as t_j. Assuming a multimodal setting in which

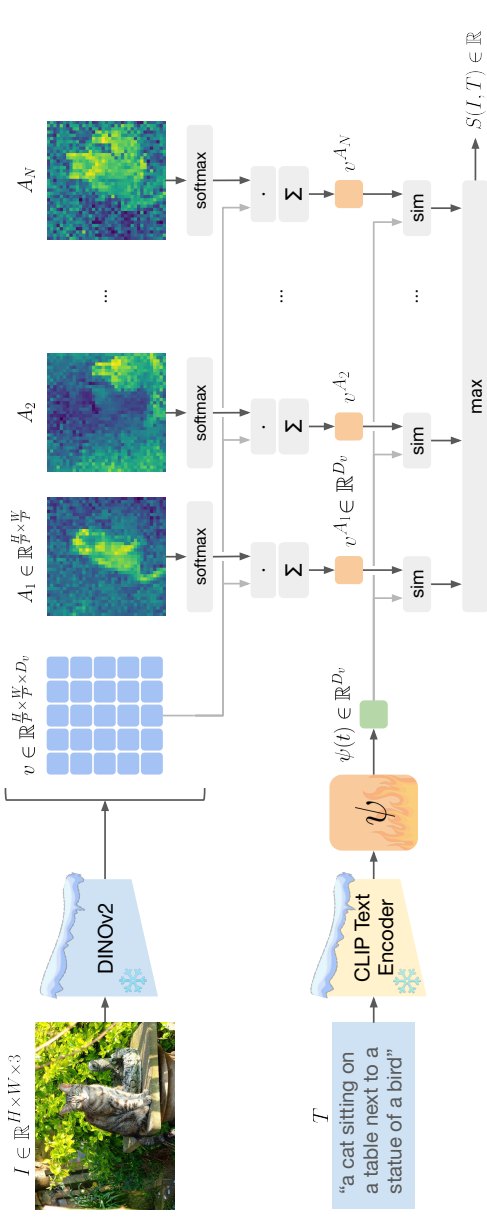

Figure 4.1: **Overview of the training methodology of Talk2DINO.** We learn a projection $\psi(\cdot)$ that maps the CLIP textual embeddings to the visual embedding space of DINOv2. Given the dense feature map and the attention maps extracted from DINOv2, we generate N visual embeddings by computing a weighted average of the feature map with each attention map. We then compute the similarity between each visual embedding and the projected text embedding, and use the maximum similarity as the global alignment score.

$D_t = D_v$, we could define the similarity map $\mathcal{S}(I, T_j) \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P}}$ for the image I and category T_j as the cosine similarity between t_j and each spatial entry of v. Formally, the similarity map is defined as

$$\mathcal{S}(I, T_j)_{[h,w]} = \frac{v_{[h,w]} \cdot t_j^\intercal}{||v_{[h,w]}|| \, ||t_j||}, \tag{4.1}$$

where $\cdot_{[h,w]}$ represents indexing over spatial axes. The full resolution similarity map $\hat{\mathcal{S}}(I, T) \in \mathbb{R}^{H \times W}$ is recovered by upsampling $\mathcal{S}(I, T)$ (*e.g.*, via bilinear interpolation). Segmentation masks $\mathcal{M}(I, T_1, ..., T_M)$ are then derived by assigning pixels to the category with the highest similarity score, *i.e.*,

$$\mathcal{M}(I, T_1, ..., T_M)_{[h,w]} = \underset{j=1,...,M}{\arg\max} \, \hat{\mathcal{S}}(I, T_j)_{[h,w]}. \tag{4.2}$$

In order for Eq. 4.1, and therefore the segmentation from Eq. 4.2, to work correctly, not only the two v and t spaces should share the same dimensionality, but they should also be constructed so that they also share the same semantics.

CLIP and DINO Duality. Existing vision-language models trained on image-text pairs (*e.g.*, CLIP [249]) can naturally fit the formulation mentioned above, as they provide dense visual and textual embeddings in the same space. However, while CLIP can correctly align global features coming from texts and images (*i.e.*, through the similarities corresponding to `CLS` tokens), it lacks a precise alignment between the textual feature t and spatial patches v.

Conversely, purely visual self-supervised backbones like DINOv2 [234] have shown remarkable semantic and local consistency of spatial embeddings, enabling agnostic image segmentation [290, 289, 341]. These abilities occur naturally in the last attention layer of DINOv2, where the attention maps computed between the `CLS` token and the spatial tokens align with relevant objects within the image (see Fig. 4.1). Despite the remarkable results observed on image-only tasks, DINOv2 lacks a solid bridge with natural language, making it impossible to directly compute the similarities with the text features, as expressed in Eq. 4.1.

While DINOv2 and CLIP embedding spaces are traditionally thought as being uncorrelated spaces, we show that the CLIP textual embedding space can be projected into the DINOv2 space through a learnable nonlinear warping.

4.1.1.2 Augmenting DINO with Semantics

Warping CLIP Embedding Space. We learn a projection $\psi : \mathbb{R}^{D_t} \rightarrow \mathbb{R}^{D_v}$ to map textual embeddings t into the space of the visual patch embeddings v

of DINOv2, leveraging weak supervision from image-text pairs. We build the projection ψ applied to textual features by composing two affine transformations with a hyperbolic tangent activation, which provides nonlinear warping. Formally,

$$\psi(t) = \boldsymbol{W}_b^\intercal \left(\tanh(\boldsymbol{W}_a^\intercal t + b_a)\right) + b_b, \tag{4.3}$$

where $\boldsymbol{W}_a \in \mathbb{R}^{D_t \times D_v}$ and $\boldsymbol{W}_b \in \mathbb{R}^{D_v \times D_v}$ are learnable projection matrices and b_* are learnable bias vectors.

Mapping DINO to the Warped CLIP Space. To learn the nonlinear projection ψ, we exploit the intrinsic segmentation capability of DINOv2 to identify the precise spatial subsets of v to which $\psi(t)$ should be aligned with.

Specifically, we first extract the N attention maps $A_i \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P}}$ (one for each of the $i = 1, ..., N$ heads) which DINOv2 computes between the `CLS` vector and its patch features from the last layer. One of the key features of DINOv2 is that each A_i highlights different semantic regions within the image. For each attention map A_i, we compute a visual embedding $v^{A_i} \in \mathbb{R}^{D_v}$ as a weighted average of the dense feature map v, emphasizing the spatial areas that A_i highlights. We then calculate the cosine similarity between each v^{A_i} and the projected text embedding $\psi(t)$, resulting in N similarity scores. Formally, the cosine similarity score between a head and the text embedding is defined as

$$\text{sim}(v^{A_i}, t) = \frac{v^{A_i} \cdot \psi(t)^\intercal}{||v^{A_i}||\,||\psi(t)||}, \tag{4.4}$$

$$\text{with } v^{A_i} = \sum_{h,w} v_{[h,w]} \text{softmax}(A_i)_{[h,w]}. \tag{4.5}$$

To obtain the most relevant score for alignment, we apply a selection function over the similarity scores obtained for different heads. In particular, we choose the maximum similarity $\max_{i=1,...,N} \text{sim}(v^{A_i}, t)$ score across all heads, therefore promoting a robust alignment between textual and visual representations that adapts to the most salient visual features corresponding to the text query.

Training Procedure. To optimize the alignment between text and visual embeddings, we employ the InfoNCE loss, which leverages similarity scores across a batch of image-text pairs. For each pair (I_i, T_i), we compute similarity scores between the projected text embedding $\psi(t_i)$ and the maximally-activated visual embedding $\tilde{v_i}$, where $\tilde{v_i}$ is the visual embedding derived from the most relevant attention head for the corresponding text t_i, *i.e.*,

$$\tilde{v_i} = v_i^{A_j} \mid j = \underset{k=1,...,N}{\arg\max}\, \text{sim}(v_i^{A_k}, t_i). \tag{4.6}$$

Treating the true image-text pair as the positive instance and the remaining pairs within the batch as negatives, this contrastive approach drives the model to increase similarity for matching pairs and decrease it for non-matching pairs. Formally, the InfoNCE loss $\mathcal{L}_{\text{InfoNCE}}$ for a batch of B image-text pairs is defined as

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{2B}\sum_{i=1}^{B}\log\frac{\exp(\text{sim}(\tilde{v}_i, t_i))}{\sum_{j=1}^{B}\exp(\text{sim}(\tilde{v}_j, t_i))} -\frac{1}{2B}\sum_{i=1}^{B}\log\frac{\exp(\text{sim}(\tilde{v}_i, t_i))}{\sum_{j=1}^{B}\exp(\text{sim}(\tilde{v}_i, t_j))}.$$

This formulation effectively strengthens alignment by maximizing the similarity for true pairs and minimizing it for mismatched pairs across the batch.

Inference. The projection learned during the training procedure that warps the CLIP embedding space into the DINOv2 space enables the textual embeddings to be directly comparable with the dense feature embeddings from DINOv2. Hence, at inference time, given an image $I \in \mathbb{R}^{H\times W\times 3}$ and a set of textual arbitrary categories $\{T_j\}_{j=1,\dots,M}$, we can obtain the segmentation masks as defined in Eq. 4.2 by considering the projected text embeddings $\psi(t_j)$ in the similarity map computation from Eq. 4.1.

4.1.1.3 Identifying Background Regions

An additional challenge that OVS approaches need to face, especially when tasked with benchmarks like Pascal VOC [97] and COCO Objects [31], is that of identifying "background" regions, *i.e.* regions that do not belong to the set of categories considered in the benchmark. The standard approach consists in applying a threshold on the similarity or probability score to identify where the model is not certain about the predicted category and classify these locations as background. However, previous works [147, 357, 356] have introduced custom approaches to improve the capabilities of the model in recognizing the background.

Following this line, we propose a background cleaning procedure, depicted in Fig. 4.2, that is based on the capabilities of the DINOv2 backbone in focusing on coherent areas and highlighting the foreground through the self-attention heads. Specifically, given N attention maps $A_i \in \mathbb{R}^{\frac{H}{P}\times\frac{W}{P}}$ and M projected textual embeddings of classes $\psi(t_j)$, we first compute the average visual embeddings v^{A_i} as in Eq. 4.5. Similarly to the training procedure, we then compute the similarity

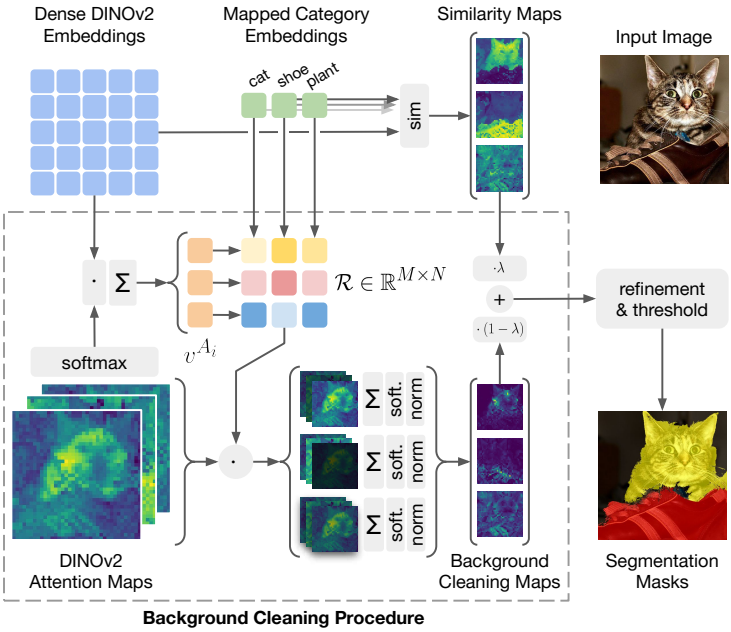

Figure 4.2: **Inference procedure.** At the top, we compute the similarity between mapped text embeddings and the DINOv2 patches to produce the initial similarity maps. In the bottom part, we produce a background cleaning map for each class derived from the different DINOv2 attention heads. We obtain the final enhanced similarity map of each category through a convex combination of the similarity and background cleaning maps. The output segmentation then results from the final refinement and thresholding steps.

between each v^{A_i} and $\psi(t_j)$, resulting in a matrix of similarity scores $\mathcal{R} \in \mathbb{R}^{M \times N}$, which is additionally normalized row-wise through a softmax operation. These scores represent how much each self-attention head is related to each textual category. Formally, $\mathcal{R}$ is defined as

$$\begin{aligned} \mathcal{R} &= [\mathcal{R}_1, ..., \mathcal{R}_j, ..., \mathcal{R}_M]^\intercal, \text{ with} \\ \mathcal{R}_j &= \text{softmax}\left(\text{sim}(v^{A_1}, \psi(t_j)), ..., \text{sim}(v^{A_N}, \psi(t_j))\right). \end{aligned} \tag{4.7}$$

Then, for each category T_j we compute its average attention map $\mathcal{F}_j \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P}}$ as

$$\mathcal{F}_j(A_i, \mathcal{R}_{ij}) = \sum_{i=1}^{N} \mathcal{R}_{ij} A_i, \tag{4.8}$$

and normalize $\mathcal{F}$ by applying a softmax normalization over both spatial axes, and linearly re-projecting its values in the range

$$[\min_{j,h,w} \mathcal{S}(I, T_j)_{[h,w]}, \max_{j,h,w} \mathcal{S}(I, T_j)_{[h,w]}], \tag{4.9}$$

where $\mathcal{S}(\cdot)$ is the similarity map defined in Eq. 4.1. We exploit the resulting normalized average attention per category to shape the similarity map by activating the foreground region and deactivating the background. The resulting shaped similarity map $\overline{\mathcal{S}} \in \mathbb{R}^{H \times W}$ is defined as

$$\overline{\mathcal{S}}(I, T_j)_{[h,w]} = \lambda \mathcal{S}(I, T_j)_{[h,w]} + (1 - \lambda) \mathcal{F}_{j,[h,w]}, \tag{4.10}$$

where λ is a hyperparameter representing the relevance of the background shaping in computing the segmentation masks. The background mask is then identified as the collection of pixels for which the shaped similarity map is lower than a threshold across all semantic categories.

4.1.2 Experiments

4.1.2.1 Experimental Setup

Datasets. We evaluate our approach on the eight semantic segmentation benchmarks introduced in Sec. 3.3, which we categorize based on the inclusion of a background class. Specifically, we conduct experiments on the validation sets of Pascal VOC 2012 [97], Pascal Context [224], COCO Stuff [31], Cityscapes [71], and ADE20K [420, 421], that contain 20, 59, 171, 150, and 19 semantic categories respectively and do not include the "background" class. We report additional experiments on the COCO Objects dataset [31], which consists of 80 different foreground object classes, and on modified versions of Pascal VOC 2012 and Pascal Context in which the "background" category is, instead, included (*i.e.*, with 21 and 60 semantic categories respectively).

Implementation Details. For the main experiments, we employ DINOv2 ViT-B/14 as the base model and DINOv2 ViT-L/14 as the large model, both with the CLIP ViT-B/16 text encoder. We use the DINOv2 variant with registers [77] since

Model	Visual Encoder	Frozen	ViT-Base (mIoU)									ViT-Large (mIoU)								
			V20	C59	Stuff	City	ADE	V21	C60	Object	**Avg**	V20	C59	Stuff	City	ADE	V21	C60	Object	**Avg**
without Mask Refinement																				
GroupViT [363]	Custom ViT	✗	79.7	23.4	15.3	11.1	9.2	50.4	18.7	27.5	29.4	-	-	-	-	-	-	-	-	-
ReCo [286]	CLIP	✗	57.7	22.3	14.8	21.1	11.2	25.1	19.9	15.7	23.5	-	-	-	-	-	-	-	-	-
TCL [39]	CLIP	✗	77.5	30.3	19.6	23.1	14.9	51.2	24.3	30.4	33.9	-	-	-	-	-	-	-	-	-
SILC [228]	Custom ViT	✗	77.5	31.6	20.8	26.9	19.3	-	-	-	-	-	-	-	-	-	-	-	-	-
MaskCLIP [422]	CLIP	✓	74.9	26.4	16.4	12.6	9.8	38.8	23.6	20.6	27.9	29.4	12.4	8.8	11.5	7.2	23.3	11.7	7.2	13.9
CLIP-DIY [356]	CLIP+DINO	✓	79.7	19.8	13.3	11.6	9.9	59.9	19.7	31.0	30.6	-	-	-	-	-	-	-	-	-
SCLIP [328]	CLIP	✓	80.4	34.2	22.4	32.2	16.1	59.1	30.4	30.5	38.2	70.6	25.2	17.6	21.3	10.9	44.0	22.3	26.9	29.9
CLIP-DINOiser [357]	CLIP	✓	80.9	35.9	24.6	31.1	20.0	**62.1**	32.4	34.8	40.2	-	-	-	-	-	-	-	-	-
ClearCLIP [165]	CLIP	✓	80.9	35.9	23.9	30.0	16.7	51.8	32.6	33.0	38.1	80.0	29.6	19.9	27.9	15.0	-	-	-	-
NACLIP [116]	CLIP	✓	79.7	35.2	23.3	35.5	17.4	58.9	32.2	33.2	39.4	78.7	32.1	21.4	31.4	17.3	52.2	28.7	29.9	36.5
`dino.txt` [140]	DINOv2(reg)	✓	-	-	-	-	-	-	-	-	-	62.1	30.9	20.9	32.1	20.6	-	-	-	-
FreeDA [16]	DINOv2	✓	77.1	37.1	24.9	34.0	19.5	51.7	32.6	24.4	37.7	71.8	35.4	24.2	32.3	19.4	44.9	31.1	24.6	35.5
FreeDA [16]	CLIP+DINOv2	✓	84.3	39.7	25.7	34.1	20.8	51.8	35.3	36.3	41.0	85.7	**39.7**	26.3	33.6	21.4	44.1	34.8	33.9	39.9
ProxyCLIP [166]	CLIP+DINOv2(reg)	✓	83.0	37.2	25.4	33.9	19.7	58.6	33.8	37.4	41.1	85.2	36.2	24.6	35.2	21.6	56.6	33.0	36.7	41.1
ProxyCLIP [166]	CLIP+DINO	✓	80.3	39.1	26.5	**38.1**	20.2	61.3	**35.3**	37.5	42.3	83.2	37.7	25.6	**40.1**	**22.6**	**60.6**	**34.5**	**39.2**	**42.9**
Talk2DINO (Ours)	DINOv2(reg)	✓	**87.1**	**39.8**	**28.1**	36.6	**21.1**	61.5	35.1	**41.0**	**43.8**	**87.1**	39.1	**27.0**	35.8	21.1	60.1	34.2	37.6	42.8
with Mask Refinement																				
GroupViT [363]	Custom ViT	✗	81.5	23.8	15.4	11.6	9.4	51.1	19.0	27.9	30.0	-	-	-	-	-	-	-	-	-
ReCo [286]	CLIP	✗	62.4	24.7	16.3	22.8	12.4	27.2	21.9	17.3	25.6	-	-	-	-	-	-	-	-	-
TCL [39]	CLIP	✗	83.2	33.9	22.4	24.0	17.1	55.0	30.4	31.6	37.2	-	-	-	-	-	-	-	-	-
MaskCLIP [422]	CLIP	✓	72.1	25.3	15.1	11.2	9.0	37.2	22.6	18.9	26.4	-	-	-	-	-	-	-	-	-
SCLIP [328]	CLIP	✓	83.5	36.1	23.9	34.1	17.8	61.7	31.5	32.1	40.1	76.3	27.4	18.7	23.9	11.8	47.8	23.8	26.9	32.1
CLIP-DINOiser [357]	CLIP	✓	81.5	37.1	25.3	31.5	20.6	64.6	33.5	36.1	41.3	-	-	-	-	-	-	-	-	-
LaVG [143]	CLIP+DINO	✓	82.5	34.7	23.2	26.2	15.8	62.1	31.6	34.2	38.3	-	-	-	-	-	-	-	-	-
NACLIP [116]	CLIP	✓	83.0	38.4	25.7	38.3	19.1	64.1	35.0	36.2	42.5	84.5	36.4	24.6	37.1	19.6	57.9	36.4	34.6	41.4
FreeDA	DINOv2	✓	79.5	40.2	27.1	34.4	20.9	52.0	35.2	25.8	39.4	75.2	39.0	27.0	33.1	21.3	45.3	34.3	26.7	37.7
FreeDA	CLIP+DINOv2	✓	85.2	42.1	27.0	33.8	21.8	51.8	37.4	38.6	42.2	87.1	42.4	28.1	33.8	22.6	55.4	37.1	36.1	42.8
ProxyCLIP [166]	CLIP+DINOv2(reg)	✓	83.1	38.9	26.6	35.4	20.3	62.0	35.2	38.7	42.5	85.8	37.6	25.6	37.5	22.5	59.4	34.6	39.0	42.8
ProxyCLIP [166]	CLIP+DINO	✓	80.3	39.4	26.9	**38.6**	20.2	60.8	35.3	37.2	42.3	83.2	38.0	26.2	**41.0**	22.6	60.7	34.7	39.4	43.2
Talk2DINO (Ours)	DINOv2(reg)	✓	**88.5**	**42.4**	**30.2**	38.1	**22.5**	**65.8**	**37.7**	**45.1**	**46.3**	**89.8**	**42.7**	**29.6**	38.4	**22.9**	**66.1**	**37.3**	**42.3**	**46.1**

Table 4.1: Comparison with unsupervised OVS models on Pascal VOC [97], Pascal Context [224], COCO Stuff [31], COCO Object [31], Cityscapes [71], and ADE20K [420, 421]. For each method, we specify the visual backbone used, along with whether it is frozen or fine-tuned. We report both the variants with and without background for Pascal VOC (V21 and V20) and Pascal Context (C60 and C59). Best results with and without mask refinement are highlighted in bold, overall best results are underlined.

our method benefits from the removal of artifacts in self-attention maps. We train the model with the Adam optimizer, a batch size of 128, and a learning rate of 1×10^{-4} for a total of 100 epochs on the COCO Captions 2014 training split [190], composed of around 80k images.

Following previous works [39, 147, 116], we optionally employ the mask refinement stage to counteract any inaccuracies in the final masks based on the Pixel-Adaptive Mask Refinement (PAMR) [9], which is an iterative post-refinement method aimed at enhancing the fidelity of the similarities to the visual characteristics of the image. In our experiments, we use λ equal to $\frac{5}{6}$ for background cleaning, a threshold of 0.55 on the similarity score to determine which pixels belong to the "background" category, and, when using mask refinement, employ PAMR with 10 iterations.

Evaluation Protocol. As in FOSSIL and FreeDA, we follow the standard evaluation protocol for unsupervised OVS [39], where prior access to the target data before evaluation is not allowed, and use the default class names provided by the `MMSegmentation` toolbox. The images are resized to have a shorter side of 448, using a sliding window approach with a stride of 224 pixels. All models are evaluated using mean Intersection-over-Union (mIoU) on all the classes of each dataset.

4.1.2.2 Comparison with the State of the Art

We compare Talk2DINO with previous state-of-the-art approaches for unsupervised OVS on the five benchmarks that do not include the "background" category and the three benchmarks with the "background" category. We consider as competitors: (i) prototype-based approaches, such as ReCo [286] and FreeDA (Sec. 3.3), which aim to create visual prototypes associated with the textual categories, (ii) CLIP adaptations, as MaskCLIP [422], CLIP-DIY [356], SCLIP [328], ClearCLIP [165], and NACLIP [116], which propose architectural modifications to enhance its localization properties, (iii) methods trained on sets of image-caption pairs with objectives designed to force the segmentation capabilities to emerge, like GroupViT [363], TCL [39], SILC [228], and `dino.txt` [140], and (iv) methods that aim to combine the properties of CLIP and DINO, as CLIP-DINOiser [357], LaVG [143], and ProxyCLIP [166].

Table 4.1 reports the results on the five benchmarks without background (*i.e.*, Pascal VOC-20, Pascal Context-59, COCO Stuff, Cityscapes, and ADE) and the three benchmarks with background (*i.e.*, Pascal VOC-21, Pascal Context-60, and COCO Object). Specifically, we report the performance of both the base and

large configurations of both Talk2DINO and the competitors, according to their definitions in the original papers. Moreover, we divide the table into two sections depending on whether a mask refinement technique is employed. Specifically, LaVG exploits a custom region proposer combined with DenseCRF [158], while all the other methods refine their masks with PAMR. Regarding datasets with background, Pascal VOC and COCO Objects present only foreground categories, also referred to as "things" in the literature, while Pascal Context presents both categories from foreground and background, also mentioned as "stuff". Hence, following CLIP-DINOiser [357], we report the performance with the background cleaning procedure described in Sec. 4.1.1.3 only on Pascal VOC and COCO Objects.

As it can be observed, our approach achieves the best average mIoU on all the configurations and presents a consistent improvement compared to the considered competitors, with and without the mask refinement, across all datasets except Cityscapes. The most straightforward comparison is the one with FreeDA without global similarity (*i.e.*, with DINOv2 only as visual backbone). As presented in Sec. 3.3, it builds a bridge between DINOv2 and the CLIP text encoder by retrieving from a collection of visual-textual embedding pairs and by building visual prototypes for each textual category. The significant improvement achieved by Talk2DINO demonstrates that training a direct projection from the CLIP text encoder to DINOv2 leads to a more accurate bridge between the two embedding spaces without the overhead in computation and memory provided by the retrieval procedure.

Fig. 4.3 depicts qualitative segmentation results, in which we highlight the segmentation capabilities of Talk2DINO along with other state-of-the-art models (*i.e.*, FreeDA, ProxyCLIP [166], and CLIP-DINOiser [357]). We show two images from Pascal VOC, in which it can also be appreciated how the background cleaning procedure leads to high-quality masks and localization, and two images from COCO Stuff and Pascal Context where Talk2DINO effectively segments "things" in the scene, such as the `boat` and `teddy bear`, and "stuff" categories, such as `sky` and `road`.

4.1.2.3 Ablation Studies and Analyses

Choosing Different Visual Backbones. In Table 4.2 we show the performance of our approach when varying the visual backbone and the size of the employed ViT architecture. We observe that backbones that differ from DINOv2 present unsatisfactory results and can not be aligned to the CLIP textual encoder with a learnable

Visual Backbone		mIoU				
		V20	C59	Stuff	City	ADE
DINO	ViT-S	27.7	13.2	7.4	14.8	5.2
DINOv2 (without registers)	ViT-S	83.7	**38.3**	**25.8**	**32.9**	**19.8**
DINOv2 (with registers)	ViT-S	**86.9**	35.3	24.5	27.2	16.9
MAE	ViT-B	9.5	4.3	2.0	4.0	1.1
CLIP	ViT-B	55.4	14.9	12.2	6.2	3.8
DINO	ViT-B	27.3	11.2	7.9	13.6	4.5
DINOv2 (without registers)	ViT-B	74.2	31.9	23.0	27.9	16.5
DINOv2 (with registers)	ViT-B	**87.1**	**39.8**	**28.1**	**36.6**	**21.1**
MAE	ViT-L	6.7	2.7	1.4	4.6	0.9
CLIP	ViT-L	16.6	5.0	7.7	0.9	2.0
DINOv2 (without registers)	ViT-L	56.0	20.1	14.9	18.1	8.2
DINOv2 (with registers)	ViT-L	**87.1**	**39.1**	**27.0**	**35.8**	**21.1**

Table 4.2: Ablation study results using different visual backbones and with different sizes of the ViT architecture.

mapping. In particular, while DINO achieves the second-best performance on average, Talk2DINO heavily benefits from the strong semantic representation of the dense features of DINOv2 and the capabilities of its self-attention heads in highlighting coherent regions of the image, properties that are not reflected in other visual backbones. We attribute this performance gap between DINOv2 and DINO, MAE, and CLIP to two major factors: (i) the quality of the attention maps and (ii) the semantic richness of the patch representations.

For the first point, we qualitatively analyze the self-attention patterns of the different backbones. Fig. 4.11 showcases the average self-attentions between the `CLS` token and the other tokens in the first row, breaking down the contributions from the various self-attention heads in the successive rows. We observe that the self-attention heads of CLIP introduce a noise pattern similar to what we observed for DINOv2 without registers, which limits the effectiveness of our training pipeline. On the other hand, the self-attention maps of DINO and MAE appear cleaner and emphasize homogeneous image regions. However, in these cases, the performance gap with DINOv2 can be attributed to the insufficient semantic richness of the extracted dense features.

To quantitatively assess the patch-level semantics of these backbones, we conduct an experiment in which we classify each patch through linear probing on the images of VOC. We determine the ground-truth labels of the patches via majority voting and evaluate accuracy on the validation set (batch size = 16,

Visual Backbone	ViT-S	ViT-B	ViT-L
MAE	-	0.56	0.56
CLIP	-	0.89	0.89
DINO	0.62	0.73	-
DINOv2 (without registers)	0.95	0.96	0.95
DINOv2 (with registers)	**0.96**	**0.97**	**0.96**

Table 4.3: Patch linear probing accuracy on VOC for ViT-Small, ViT-Base, and ViT-Large.

learning rate = 5×10^{-3}, for 3 epochs, with 32×32 patches per image, using ViT-B as the backbone). The results, reported in Table 4.3, align with the overall trends highlighted in the paper: DINOv2 consistently emerges as the best-performing backbone, MAE as the worst, and CLIP and DINO as intermediate. These findings further confirm that the semantic richness of the features extracted by different backbones plays a crucial role in the effectiveness of our approach. Similar conclusions were drawn in the ablation study of FreeDA (Sec. 3.3), where a comparable performance drop was observed when using CLIP or DINO instead of DINOv2.

Moreover, our results emphasize the critical role of registers [77] in DINOv2, as demonstrated by the comparison between its variants with and without registers. Registers are a recently proposed mechanism to mitigate the presence of artifacts in the feature maps of ViT-based backbones. Artifacts are tokens that exhibit a significantly higher norm with respect to the other tokens and retain less information about their original position in the image. The alignment process in our method relies on high-quality attention maps, and the presence of artifacts poses a challenge by limiting the selection of the most relevant self-attention heads. Interestingly, since register-related artifacts are more pronounced in larger backbones, the ViT-S variant without registers maintains competitive performance compared to its register-enabled counterpart. See "*Role of DINO Registers*" for more details. Finally, we observe that our approach maintains robust and consistent performance across different ViT sizes, achieving strong results even with the compact ViT-S backbone. This suggests that our method is effective across a range of model sizes, making it adaptable to varying computational constraints. See "*ViT-B vs. ViT-L*" for more details.

Impact of the Proposed Components. Table 4.4 reports the results of Talk2DINO evaluating the impact of its core components on the overall performance. Specifically, in the first section of the table, we analyze the effect of the adopted projection

	mIoU				
	V20	C59	Stuff	City	ADE
Effect of Projection					
Linear Projection (text only)	85.1	37.9	26.7	35.6	20.1
Our mapping (both vision and text)	59.2	27.3	18.9	23.5	13.5
Our mapping (vision only)	84.6	35.2	26.2	20.4	15.5
Our mapping (text only)	**87.1**	**39.8**	**28.1**	**36.6**	**21.1**
Effect of Self-Attention Selection and Aggregation					
`CLS` only (without self-attention)	84.5	30.6	23.0	22.6	17.2
Standard average	**89.6**	36.9	25.6	33.5	19.7
`CLS`-weighted average	87.6	35.2	23.1	29.3	17.5
`CLS` similarity-weighted sampling	88.2	32.9	22.6	27.0	17.9
Max `CLS` similarity	87.1	**39.8**	**28.1**	**36.6**	**21.1**

Table 4.4: Ablation study evaluating the impact of the core components of the proposed architecture on the final performance. We report the results using the base model of DINOv2.

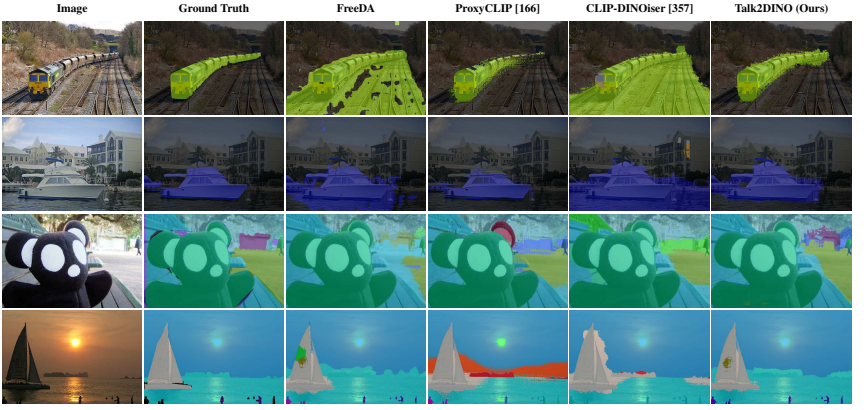

Figure 4.3: Qualitative results of Talk2DINO in comparison with FreeDA, ProxyCLIP [166], and CLIP-DINOiser [357].

ψ. Replacing it with a linear projection leads to a slight performance drop. The good performance obtained by a linear transform evidences how the DINOv2 and CLIP spaces are intrinsically compatible, as the former can be obtained through an affine transform of the latter without losing too much information. Interestingly,

	Mask Refinement	mIoU	
		V21	Object
without background cleaning	✗	59.9	37.1
with background cleaning	✗	**61.5**	**41.0**
without background cleaning	✓	63.9	40.3
with background cleaning	✓	**65.8**	**45.1**

Table 4.5: Ablation study on the impact of the background cleaning procedure. We report the results using DINOv2 ViT-B.

applying the proposed projection on top of DINOv2 or using two projections on both spaces significantly lowers performance, confirming the appropriateness of the proposed approach.

In the second section of the table, we instead study the effect of the selection and aggregation strategy of self-attention heads $A_i, i = 1, ..., N$ during training. In particular, we test aligning (i) directly the visual `CLS` token to the textual `CLS` token, (ii) the visual embedding from the standard average self-attention, (iii) the weighted mean of the head embeddings v_{A_i}, where the weights are given by their softmaxed similarity with the textual `CLS` token, (iv) a strategy in which we sample a single head embedding, where the sampling probability is given by the softmaxed similarity with the `CLS` token, and (v) our adopted solution in which we select the head embedding which is the most similar to the textual `CLS` token. The results show that only on the Pascal VOC dataset, composed mostly of large subjects in the foreground, the embedding from the standard average self-attention presents improved performance. On all other benchmarks, our approach proves to be the most effective, further validating the robustness of our selection method.

Effect of Background Cleaning. Table 4.5 shows how the performance is affected by the background cleaning mechanism and by the usage of PAMR for mask refinement. It can be observed that the background cleaning procedure has a significantly positive impact on Pascal VOC and COCO Object, leading, respectively, to a +1.6 and +3.9 increase in mIoU score. Further, it can be noticed that the effectiveness of the proposed background cleaning procedure is confirmed also when applying the mask refinement. Fig. 4.4 shows a set of qualitative results in which we highlight the advantages of using the proposed background cleaning procedure with respect to directly thresholding the similarities with the input categories to detect the background. In particular, the first two rows show four qualitatives on images from COCO Object, and the last two rows from VOC. These

Figure 4.4: Qualitative results obtained with and without the proposed background cleaning strategy, on COCO Object and Pascal VOC.

results demonstrate that background cleaning removes the noise in the background from the image and improves the fitting of the masks on the foreground objects.

Analysis of Model Parameters. Fig. 4.5 reports a comparison of the relationship between the performance, in terms of average mIoU, and the number of parameters of the models. As it can be observed, Talk2DINO presents a lower number of parameters than the competitors FreeDA and ProxyCLIP [166], along with an improved average mIoU. Models with a comparable number of parameters, such as TCL [39], GroupViT [363], and MaskCLIP [422], exhibit a lower performance compared to Talk2DINO. Finally, it shall be noted that models such as FreeDA and ReCo [286] require maintaining external sources of knowledge, which increases memory consumption. Further discussion on the comparison between Talk2DINO, ProxyCLIP, and FreeDA can be found in the following sections (see "*Comparison with ProxyCLIP and FreeDA*").

Role of DINO Registers. The main configuration of Talk2DINO, with both the base and large sizes, leverages the variant of DINOv2 with registers. In Fig. 4.10, we depict, on the first row, the average self-attentions between the `CLS` and the other tokens for the ViT-S, ViT-B, and ViT-L architectures with and without registers, while in the following rows, we show the various self-attention heads for each backbone. It can be observed that in the ViT-S the artifacts are not present, and the average self-attention between the model with and without the registers is nearly identical. Instead, the ViT-B exhibits artifacts in the top left corner, resulting in an average self-attention that is especially focused on that portion of the image. This side effect is even more noticeable with the ViT-L, for which the artifact is the

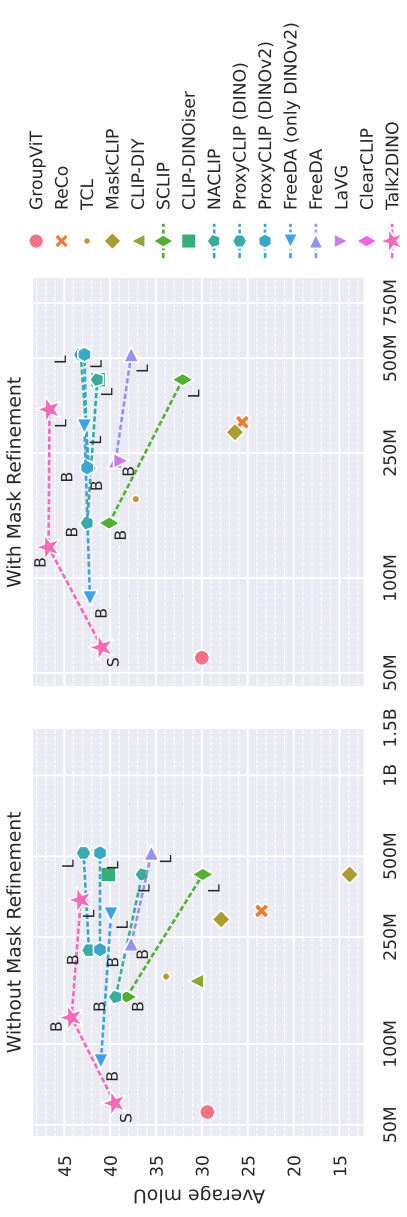

Figure 4.5: **Performance vs. Parameter Count.** The y-axis denotes the obtained mIoU averaged over all the five benchmarks reported in the Table 4.1. Dashed lines connect methods tested with multiple backbone sizes. These are labeled S/B/L for Small/Base/Large ViT models. Talk2DINO offers the best trade-off between performance and number of parameters.

	mIoU				
	V20	C59	Stuff	City	ADE
Effect of Training CLIP Last Layer					
Trained	77.9	31.5	21.3	34.6	18.7
Frozen	**87.1**	**39.8**	**28.1**	**36.6**	**21.1**
Effect of Text Token Selection					
Average text tokens	84.7	37.9	25.7	33.6	20.0
Text token to best self-attn map	83.9	33.8	24.2	29.5	18.1
Text token to best self-attn map (NS)	80.8	33.9	23.7	27.5	18.6
`CLS` token only	**87.1**	**39.8**	**28.1**	**36.6**	**21.1**

Table 4.6: Ablation study on the effect of training the last layer of CLIP and text token selection strategies.

only visible token in the average self-attention. These observations align with the results reported in Tab. 4.2, which show a downgrade in performance without the registers that is directly related to the presence of the artifacts in the self-attentions. Indeed, the largest difference in performance is measured in the ViT-L architecture, while in the ViT-S case, the backbone without registers performs better on four benchmarks out of five.

Effect of Training CLIP Last Layer. Table 4.6 reports a comparison between Talk2DINO when training only the $\psi(t)$ projection and when instead unfreezing the last layer of CLIP [249]. Despite this experiment exhibiting a small performance gap between the two configurations, unfreezing the last layer of CLIP, interestingly, leads to worse results. This outcome highlights that the textual representations provided by CLIP, which have been pre-trained to match their visual counterpart, if trained inside a different pipeline, can be harmed and can lose part of their capabilities in multimodal understanding.

Using CLIP Text Tokens. In Table 4.6, we report the results of utilizing the dense output of the CLIP text encoder instead of its `CLS` token for alignment. While our primary experiments align the `CLS` token with the best attention map embedding to target the patches most relevant to the text, we also explore aligning individual text tokens to the best attention map embeddings. This approach is motivated by the hypothesis that each word in the text might correspond to a distinct region in the image. During inference, since we perform the alignment on individual text tokens rather than the `CLS` token, we average the text tokens to calculate similarity with the visual patches. However, this method yields inferior results compared to using the `CLS` token. We then refine this approach by aligning only a subset of text tokens selected using nucleus sampling ($\alpha = 0.6$) to filter out potentially

irrelevant words, such as stop words. Despite this effort, performance does not improve.

These observations suggest that the global objective of the training of CLIP, similar to its effect on visual patch embeddings, may not endow text tokens with strong local properties that accurately reflect the specific word each embedding represents. This limitation likely contributes to the noisiness of such alignments. Additionally, we evaluate the use of the average of CLIP text tokens in both training and inference as an alternative to the `CLS` token. While this approach slightly improves over aligning individual tokens, it still underperforms compared to the `CLS` token, indicating that it encapsulates the most useful and less noisy information for alignment with DINOv2 patches.

Impact of Image Resolution. According to the evaluation protocol introduced in GroupViT [363] and standardized in TCL [39], the images are resized to have a shorter side of 448, and a sliding window approach with a stride of 224 pixels is employed. However, Wang *et al.* [328] observed that the approaches based on CLIP benefit from employing a shorter side of 336 with 224×224 windows and a stride of 112 pixels, leading to an equivalent computational effort but better performance. This phenomenon is attributed to two reasons: (i) each window has the same resolution on which CLIP has been originally trained, and (ii) CLIP presents an impressive global understanding but lacks localization capabilities, hence relying on many smaller windows is more advantageous than more patches. This variation of the evaluation setting is not necessary for Talk2DINO because DINOv2, which is the frozen underlying visual encoder, has been trained with a 518×518 resolution and presents an outstanding patch-level understanding, However, Tab. 4.6 reports the results obtained following the setting used in SCLIP, employing a shorter side of 336 for VOC, Context, COCO-Stuff and ADE, of 560 for Cityscapes, with 224×224 windows and stride 112. Results show that Talk2DINO, on average, performs better with a resolution of 448, but the performance slightly varies when changing the setting to 336. This confirms that the semantics of the patch-level features of DINOv2 are robust towards variations of resolution and that our learned bridge is valid for both scenarios. Moreover, for a fair comparison, we also report the results of SCLIP, NACLIP, and ProxyCLIP when adopting the 448 resolution of the standard protocol, in which Talk2DINO largely outperforms the competitors.

Comparison with ProxyCLIP and FreeDA. GroupViT [363] has been the first model to tackle the weakly-supervised OVS. It trains a custom ViT architecture from scratch by hierarchically merging tokens at different layers. Afterward,

Model	Visual Backbone	Resolution	ViT-Base (mIoU)						ViT-Large (mIoU)					
			V20	C59	Stuff	City	ADE	Avg	V20	C59	Stuff	City	ADE	Avg
without Mask Refinement														
SCLIP [328]	CLIP	336	80.4	34.2	22.4	32.2	16.1	37.1	70.6	25.2	17.6	21.3	10.9	29.1
NACLIP [116]	CLIP	336	79.7	35.2	23.3	35.5	17.4	38.2	78.7	32.1	21.4	31.4	17.3	36.2
ProxyCLIP [166]	CLIP+DINOv2	336	80.5	37.3	25.3	35.8	19.0	39.6	83.5	36.7	25.0	35.8	21.0	40.4
ProxyCLIP [166]	CLIP+DINO	336	78.2	38.8	26.2	**39.7**	19.7	40.5	82.1	**38.2**	**26.2**	**41.2**	**22.2**	**42.0**
Talk2DINO (Ours)	DINOv2	336	**88.3**	**39.1**	**27.4**	38.2	**20.2**	**42.6**	**86.6**	**38.2**	26.0	36.4	19.3	41.3
with Mask Refinement														
SCLIP [328]	CLIP	336	79.3	34.6	22.3	20.3	15.4	34.4	66.6	22.4	14.7	6.9	7.7	23.7
NACLIP [116]	CLIP	336	83.0	38.4	25.7	38.3	19.1	40.9	84.5	36.4	24.6	37.1	19.6	40.4
ProxyCLIP [166]	CLIP+DINOv2	336	80.9	39.3	26.6	37.7	19.9	40.9	83.5	36.7	26.4	38.6	22.1	41.5
ProxyCLIP [166]	CLIP+DINO	336	78.5	39.3	26.7	40.1	20.0	40.9	82.6	38.7	26.7	**42.1**	**22.5**	42.5
Talk2DINO (Ours)	DINOv2	336	**89.4**	**41.5**	**29.4**	**40.3**	**21.2**	**44.4**	**89.5**	**41.7**	**29.8**	38.7	20.8	**44.1**
without Mask Refinement														
SCLIP [328]	CLIP	448	77.8	33.0	21.1	19.8	14.6	33.3	61.2	20.5	13.1	6.7	7.0	21.7
NACLIP [116]	CLIP	448	71.3	34.8	22.9	33.7	17.7	36.1	74.5	32.6	21.6	30.5	17.8	35.4
ProxyCLIP [166]	CLIP+DINOv2	448	83.3	37.8	25.6	28.8	19.1	38.9	85.0	36.6	25.0	33.8	20.6	40.2
ProxyCLIP [166]	CLIP+DINO	448	80.4	39.0	26.2	31.7	19.5	39.4	83.1	37.8	25.9	**37.5**	**21.6**	41.2
Talk2DINO (Ours)	DINOv2	448	**87.1**	**39.8**	**28.1**	**36.6**	**21.1**	**42.5**	**87.1**	**39.1**	**27.0**	35.8	21.1	**42.0**
with Mask Refinement														
SCLIP [328]	CLIP	448	79.3	34.6	22.3	20.3	15.4	34.4	66.6	22.4	14.7	6.9	7.7	23.7
NACLIP [116]	CLIP	448	74.9	37.6	25.2	36.1	18.4	38.4	79.8	36.8	25.0	35.6	18.4	39.1
ProxyCLIP [166]	CLIP+DINOv2	448	83.1	39.3	26.7	29.5	19.7	39.7	85.1	37.8	25.9	35.3	21.4	41.1
ProxyCLIP [166]	CLIP+DINO	448	80.0	39.1	26.5	31.7	19.5	39.4	82.8	37.8	26.2	26.2	21.6	38.9
Talk2DINO (Ours)	DINOv2	448	**88.5**	**42.4**	**30.2**	**38.1**	**22.5**	**44.3**	**89.8**	**42.7**	**29.6**	**38.4**	**22.9**	**44.7**

Figure 4.6: Comparison with unsupervised OVS models on Pascal VOC [97], Pascal Context [224], COCO Stuff [31], Cityscapes [71], and ADE20K [420, 421] following the evaluation setting proposed in SCLIP [328] (*resolution 336*) and TCL [39] (*resolution 448*).

	Image→Text					Text→Image				
	R@1 ↑	R@5 ↑	R@10 ↑	Median ↓	Mean ↓	R@1 ↑	R@5 ↑	R@10 ↑	Median ↓	Mean ↓
ViT-Base										
CLIP	**41.3**	**65.8**	**76.3**	**2**	13.4	22.6	44.1	54.9	8	52.5
Talk2DINO	29.5	56.0	69.0	4	16.4	12.5	34.0	48.4	11	38.4
+ Custom Alignment	28.6	58.8	72.0	4	**12.0**	**28.0**	**55.6**	**68.7**	**4**	**19.3**
ViT-Large										
CLIP	**45.4**	**71.1**	**79.2**	**2**	**11.0**	**26.5**	48.7	59.0	6	44.2
Talk2DINO	26.5	53.7	65.6	5	18.8	12.7	33.7	47.8	11	43.1
+ Custom Alignment	37.9	64.7	75.1	3	13.1	24.4	**50.1**	**63.2**	**5**	**27.8**

Table 4.7: Retrieval performance on the COCO Captions test set.

	Visual Encoder	Params (M)	FLOPS (G)	Ext. (GiB)
ProxyCLIP	CLIP ViT-B/16 + DINO ViT-B/8	172.0	521.2	-
ProxyCLIP	CLIP ViT-B/16 + DINOv2 ViT-B/14	172.8	180.8	-
FreeDA	CLIP ViT-B/16 + DINOv2 ViT-B/14	172.8	125.1	12.5
Talk2DINO	DINOv2 ViT-B/14	86.6	107.4	-

Table 4.8: Number of parameters, FLOPS, and the dimension of the external knowledge for ProxyCLIP, FreeDA, and Talk2DINO.

several works followed this direction, investigating how to let the segmentation capabilities to emerge by training over a large corpora of image-caption pairs. On the contrary, more recent works focused on finding modifications to the architecture of CLIP in order to improve its localization properties. Moreover, some methods consider the usage of further visual encoders with enhanced localization capabilities to help CLIP on dense tasks. Among these methods, ProxyCLIP and FreeDA study how to combine DINO and DINOv2 with CLIP. ProxyCLIP proposes to leverage the semantic coherence of a visual encoder such as DINO or DINOv2 to guide the computation of the patch-level embeddings of CLIP. This guidance is performed inside an attention module, in which the patch-level embeddings of DINO act as queries and keys while those of CLIP act as values.

Talk2DINO, similarly to our proposed FreeDA and ProxyCLIP, investigates how to leverage DINOv2 to compensate for CLIP. However, it employs a contrastive learning over a large set of image-caption pairs based on maximum similarity between the attention head embeddings and texts, to learn a functional mapping that bridges the CLIP text embeddings into the DINOv2 space. This approach demonstrates that the two spaces can be directly connected to set the new state-of-the-art in the unsupervised OVS field. Table 4.8 shows a quantitative comparison in terms of the number of parameters and FLOPS of the visual encoders and the dimension of the external knowledge (*i.e.*, the database of FreeDA), when

Model	Visual Encoder	V20	C59	Stuff	City	ADE	V21	C60	Object	Avg
DINOv2 ViT-B/14 with registers (without Mask Refinement)										
FreeDA	DINOv2	83.4	39.5	25.9	35.2	20.7	50.1 ▷ 43.6	34.3	23.8 ▷ 24.7	39.1 ▷ 38.4
FreeDA	CLIP+DINOv2	87.0	**40.6**	25.7	34.2	**21.2**	49.3 ▷ 41.8	**35.7**	34.8 ▷ 34.7	41.1 ▷ 40.1
ProxyCLIP	CLIP+DINOv2	83.0	37.2	25.4	33.9	19.7	58.6 ▷ 60.0	33.8	37.4 ▷ 37.3	41.1 ▷ 41.3
Talk2DINO	DINOv2	**87.1**	39.8	**28.1**	**39.6**	21.1	59.9 ▷ **61.5**	35.1	37.1 ▷ **41.0**	43.5 ▷ **44.2**
DINOv2 ViT-B/14 with registers (with Mask Refinement)										
FreeDA	DINOv2	84.9	42.3	27.7	36.8	22.0	50.2 ▷ 43.7	36.7	24.5 ▷ 25.5	40.6 ▷ 40.0
FreeDA	CLIP+DINOv2	87.4	**42.4**	26.6	34.8	22.1	49.4 ▷ 41.7	37.2	36.6 ▷ 36.7	42.1 ▷ 41.1
ProxyCLIP	CLIP+DINOv2	83.1	38.9	26.6	35.4	20.3	62.0 ▷ 63.4	35.2	38.7 ▷ 38.6	42.5 ▷ 42.7
Talk2DINO	DINOv2	**88.5**	**42.4**	**30.2**	**41.6**	**22.5**	63.9 ▷ **65.8**	**37.7**	40.3 ▷ **45.1**	45.9 ▷ **46.7**

Table 4.9: Comparison between FreeDA, ProxyCLIP, and Talk2DINO when using DINOv2 with and without registers. For VOC21, Object, and the average, we report the results without background cleaning on the left and with background cleaning on the right.

assuming an input image with a resolution of 448×448. The results highlight that our method is more practical and less demanding in computation and memory, while presenting improved results against all competitors.

In Tab. 4.1, we followed the original configurations of the competitors and, hence, ProxyCLIP uses DINOv2 with registers while FreeDA does not. We report a comparison with and without registers in Tab. 4.9. The registers present the greatest impact on Talk2DINO, because, as described in "*Role of DINO Registers*", the presence of anomaly tokens leads all the self-attention heads to focus only on them, preventing the selection of diverse areas during training and, hence, limiting the efficacy of our proposal. Moreover, in Tab. 4.9 we report the effect of the background cleaning also on FreeDA and ProxyCLIP. This approach is effective only on Talk2DINO due to the learned alignment between text and average embeddings of the self-attention heads, while it leads to lower results when applied to the other methods.

ViT-B vs ViT-L. Tab. 4.1 shows that, without mask refinement, the results achieved by Talk2DINO with DINOv2 ViT-B as vision encoder are slightly better than the ones achieved with ViT-L, while the opposite should be expected. However, when we apply the PAMR for mask refinement, the results of ViT-L significantly improve, surpassing the ViT-B on five benchmarks out of eight. A similar phenomenon can be observed in other competitors, such as MaskCLIP, SCLIP, ClearCLIP, and NACLIP, while in FreeDA and ProxyCLIP we cannot establish an encoder size that prevails on the other. Even from the experiment in Tab. 4.3 on patch-level linear probing, we can observe that ViT-B performs slightly better than ViT-L. These results suggest that DINOv2 ViT-L has a comparable semantic understanding

with respect to ViT-B, but presents inferior localization properties, which are compensated through PAMR. We hypothesize that training the model with a form of weak- or self-supervision by exploiting the innate capabilities of pre-trained backbones lacks a direct relation between performance and model size. Indeed, the impressive semantic and localized understanding of DINOv2 is a consequence of its training procedure but not the direct objective. From Figure 4.10, it is noteworthy that the activations of ViT-S, ViT-B, and ViT-L have very different behaviors, impacting the results of Talk2DINO.

4.1.2.4 Image-Text Matching Results

While Talk2DINO is primarily designed for OVS, we also assess its performance on image-text retrieval to evaluate its capabilities in global image understanding. For this task, we adopt the same text encoding approach used in segmentation, projecting the CLIP text embedding. The global image representation is derived by averaging the embeddings computed from each DINOv2 attention map. Specifically, for each attention map A_i, we calculate a visual embedding $v^{A_i} \in \mathbb{R}^{D_v}$ as the weighted average of the dense feature map v. The final global image representation is then obtained by taking the mean of all v^{A_i} embeddings.

In Table 4.7, we assess the retrieval performance on the COCO Captions test set [190] using both ViT-B and ViT-L configurations. While Talk2DINO generally performs slightly below CLIP across most metrics, it demonstrates a notable advantage in the mean rank for the text-to-image retrieval task. This result underscores the ability of Talk2DINO to better address extreme failures compared to CLIP, indicating improved robustness in handling challenging or outlier queries. In addition to computing text-image similarities using cosine similarity between a global text token and a global image token, we experiment with a similarity function that mirrors the one used during training. Specifically, instead of representing the image with the mean of the v^{A_i} embeddings and calculating similarity as the cosine similarity between this representation and the text encoding, we represent the image using all v^{A_i} embeddings. We compute the similarity as $\max_{i=1,\dots,N} \text{sim}(v^{A_i}, t)$, taking the maximum similarity score across all heads. This alternative similarity function leads to significant performance improvements, allowing Talk2DINO to surpass CLIP on several metrics. This enhancement is likely due to the ability of the model to evaluate captions at a finer granularity. Captions often describe multiple aspects of an image, including both foreground and background elements. By individually examining different regions of the image as detected by distinct attention heads, the model can assign more

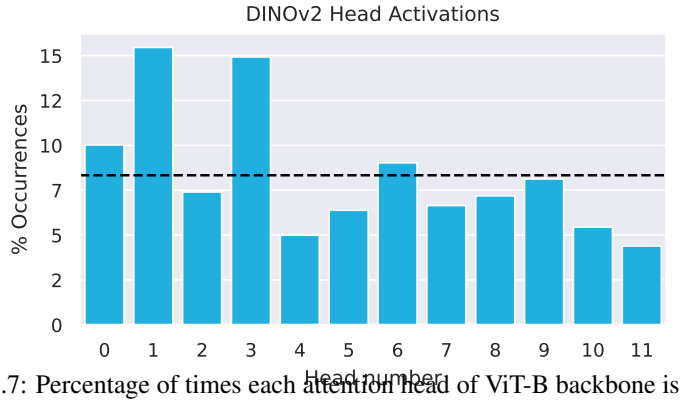

Figure 4.7: Percentage of times each attention head of ViT-B backbone is selected for alignment to textual embeddings on the final epoch of training. The dashed line denotes uniform distribution.

precise scores, ultimately boosting retrieval accuracy.

4.1.2.5 Activation Map Visualizations

In Fig. 4.7, we show the distribution of attention heads selected for alignment with the text input during the final epoch of training. The results indicate that certain heads, particularly heads 1 and 3, are more often aligned with the text than others. However, aside from these, the remaining heads are relatively evenly distributed. These findings are noteworthy because they suggest that some heads specialize in capturing features that align more closely with the input caption, while all heads contribute meaningfully during training. Notably, no head shows a negligible activation frequency, highlighting the importance of the entire set of attention heads in the alignment process.

Fig. 4.8 presents examples from the training set, showcasing images paired with their corresponding captions and the attention maps selected for alignment. Despite describing the same scene, variations in the captions lead the alignment procedure to focus on different regions of the image. For instance, in the first row, the caption mentioning the fans also focuses on the background, while captions that reference only the player, the ball, and the racket do not.

In-the-Wild Qualitative Examples. Fig. 4.9 depicts a few examples of "in-the-wild" segmentation, obtained by providing to Talk2DINO sample images from

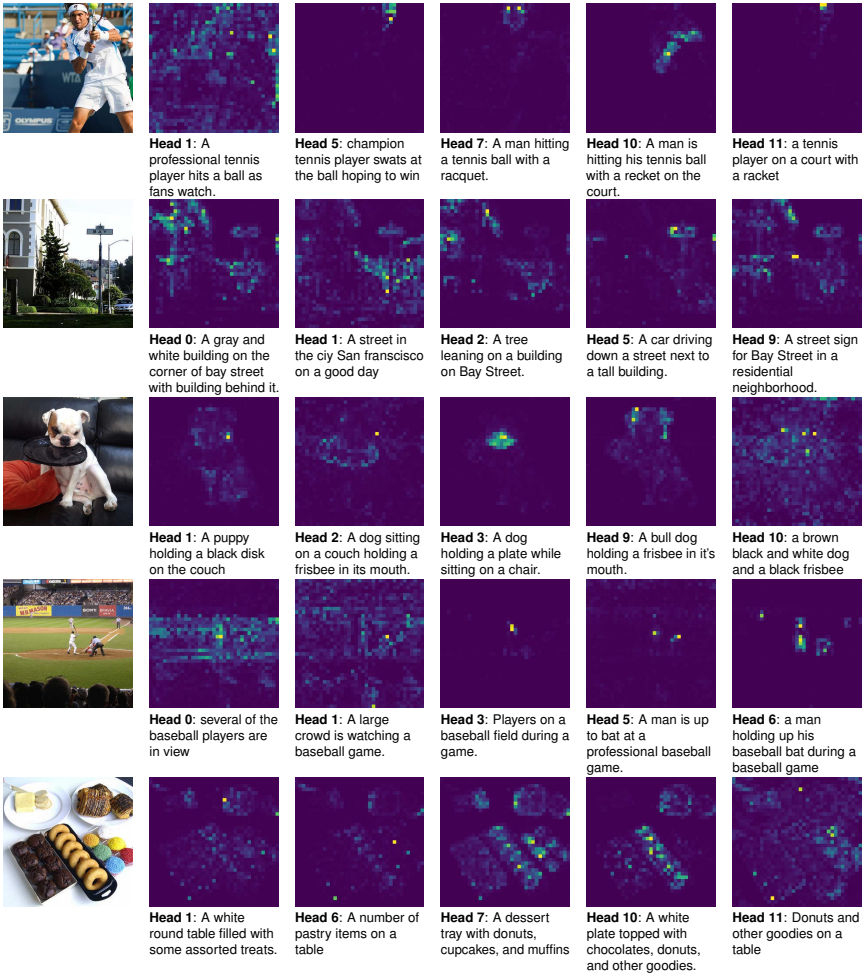

Figure 4.8: Sample images from the training set paired with their corresponding captions and the attention maps selected for alignment during the last epoch of training.

the web and asking it to segment uncommon categories, such as "`pikachu`", "`millennium falcon`", and "`westminster abbey`", and free-form text, like "`golden retriever puppy`". On the left, we show three examples in

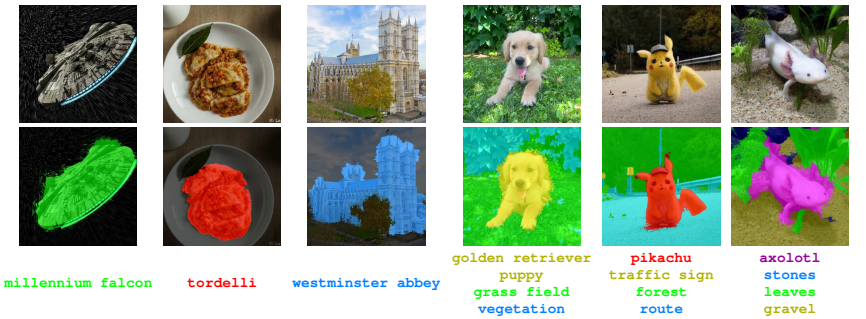

Figure 4.9: "In-the-wild" segmentation results obtained by prompting Talk2DINO with uncommon textual categories on images retrieved from the web.

which we task the model with also finding the background, while exploiting the background cleaning procedure, and, on the right, three examples in which the model has to assign a provided category to each pixel. The high quality of the resulting masks demonstrates the efficacy of our approach, even on out-of-domain images. From these examples, we can appreciate the capabilities of the model in combining the knowledge from CLIP with the semantic localization of DINOv2 on unconventional concepts, such as fictional character names and proper nouns of historical buildings.

Additional Qualitative Results. Finally, in Fig. 4.12 we report a set of qualitative results on the five datasets used for the evaluation of the models, in addition to the qualitative results depicted in Fig. 4.3. We compare the segmentation masks of Talk2DINO with the ones of FreeDA, ProxyCLIP [166], and CLIP-DINOiser [357], which represent our main competitors. In particular, we report a pair of images from Pascal VOC with background and eight pairs of images from Pascal Context, COCO Stuff, Cityscapes, and ADE20K, without background. As it can be seen, these qualitative results further highlight the impressive segmentation capabilities of Talk2DINO with both background and foreground categories.

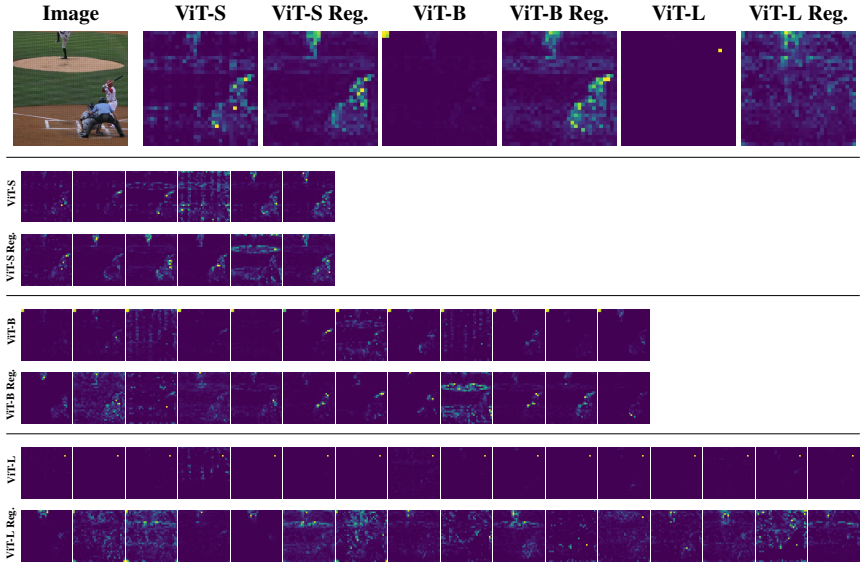

Figure 4.10: Comparison of DINOv2 with and without registers across different visual backbones (ViT-S, ViT-B, and ViT-L). The results highlight how the ViT-B and ViT-L backbones without registers exhibit artifacts that introduce noise during the alignment process.

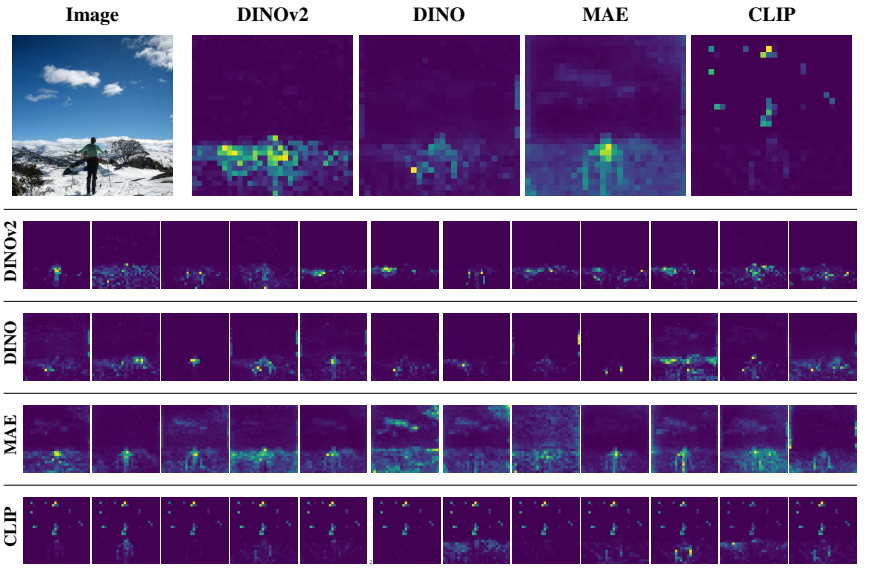

Figure 4.11: Self-attention activations of different visual backbones (*i.e.*, DINOv2, DINO, MAE, CLIP).

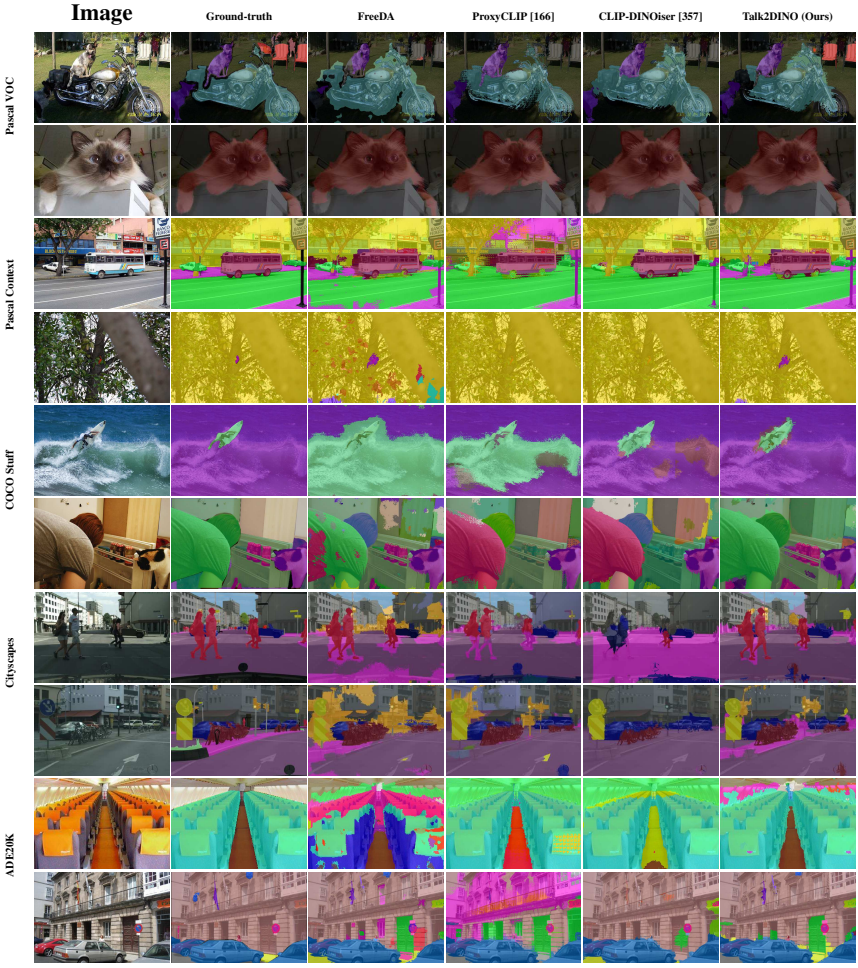

Figure 4.12: Additional qualitative results of Talk2DINO in comparison with FreeDA, ProxyCLIP [166], and CLIP-DINOiser [357].

Chapter 5

Personalized Instance-based Navigation

Imagine a scenario where your child wants their favorite teddy bear, and it has gone missing somewhere in your house. In the near future, a "smart" domestic robot could be tasked with finding it. The robot would begin exploring the environment in search of the bear. However, this task goes beyond simply detecting the category "teddy bear": the robot must locate that specific instance, which may appear in varying contexts, could be occluded or partially visible, and may be visually similar to other toys. Solving such a task requires perceptual grounding of both category-level and instance-specific concepts in realistic environments, informed by both visual and linguistic cues.

This example highlights the challenge of personalized instance-based navigation, a setting that demands not only object recognition but also the ability to identify and distinguish specific object instances across varying scenes and contexts. Throughout this thesis, we have explored the synergy between self-supervised vision models and vision–language pretraining for open-vocabulary semantic segmentation. In particular, we have shown how self-supervised models like DINOv2 exhibit strong spatial localization capabilities and can be connected to textual representations via prototypes or alignment functions, as demonstrated in FOSSIL, FreeDA, and Talk2DINO. In this chapter, we expand upon these findings by exploring a complementary capability of DINOv2: its ability to localize the same visual instance across different environmental contexts based solely on patch-

level similarities. This demonstrates that, beyond its utility for vision–language bridging, DINOv2 also excels at fine-grained visual matching in realistic embodied settings, a key requirement for achieving reliable personalized navigation.

In this Chapter, we introduce Personalized Instance-based Navigation (PIN), a new benchmark that extends the embodied AI paradigm to personalized instance-level object localization in realistic indoor environments. Unlike traditional navigation tasks that rely on predefined categories or contextual cues, PIN requires agents to identify and reach a specific object instance using visual references (*i.e.*, images of the target object) and optional textual descriptions. This setting opens up new opportunities for leveraging the open-vocabulary and spatial reasoning capabilities of vision-language and self-supervised models, pushing the boundary of embodied AI toward more human-centric interaction.

5.1 PIN

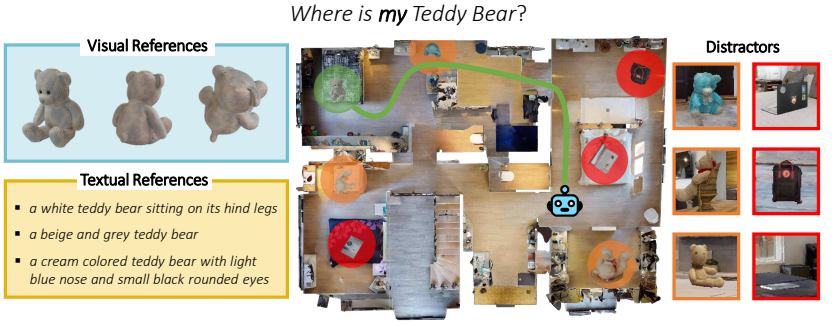

Figure 5.1: We introduce the PIN task, where the agent is asked to navigate toward a personalized object instance using multimodal references and distinguish it from distractors (*i.e.*, other objects of the same category as the target or of other categories). The target object, same category distractors, and other distractors are circled, respectively, in green, orange, and red. The total number of available objects in the dataset is 338, corresponding to different instances of 18 object categories.

The majority of current object-driven navigation tasks in Embodied AI define their goals as a general semantic category represented through text [6, 17, 344]

(*e.g.*, "chair", "sofa") or as a specific target instance defined by an image or description including the surrounding context in which the object can be found [22, 62, 150, 160, 429]. Moreover, these datasets rely on objects that were present at the time of acquisition of the environment [21, 40, 76, 150, 160, 212, 274, 358, 373]. On the contrary, procedurally generated environments can freely contain additional objects and annotations [78, 80, 157, 180]. However, the appearance discrepancy between these environments and the real world or photo-realistic environments could affect the performance of the agents when deployed on robotic platforms [141]. Previous work has proposed loading additional 3D objects inside photo-realistic environments [213] to improve agent navigation performance, to allow object interaction in static environments [282], or to enable navigation towards multiple goals [344]. However, no previous work has targeted loading objects that can be moved frequently and can appear in multiple contexts, since loaded 3D models are kept in their initial spawn position.

To overcome these issues, in this Section we propose the novel task of *Personalized Instance-based Navigation* (PIN), where the agent needs to locate and reach a specific personalized target instance in the environment provided as reference images and textual descriptions, without information about the surrounding context. An overview of PIN is shown in Fig. 5.1. In parallel with the definition of the task, we release PInNED (Personalized Instance-based Navigation Embodied Dataset), a dedicated dataset of episodes for this setting that leverages the main advantages of both photo-realistic and procedurally generated embodied environments. In each episode, along with a unique target instance, distractor objects are placed in the scene to confound the navigation of the agent. Specifically, we built the dataset on top of the semantic annotations [373] and scenes of Habitat-Matterport3D Dataset (HM3D) [253] with the injection of additional photo-realistic 3D objects accurately selected from Objaverse-XL [79]. The objects are positioned in each environment through a procedural spawning method on predefined suitable surfaces. PInNED comprises 865.5k training episodes and 1.2k validation episodes built on top of 338 additional objects.

Finally, we adapt and test available navigation agents on the proposed dataset, showcasing the shortcomings of relevant approaches. In particular, we compare the performance of the two main categories of navigation agents for object-driven navigation, modular and end-to-end approaches, where we demonstrate that the versatility of modular methods leads to superior performance compared to the end-to-end counterparts; still, the task is far from being resolved. These experiments assess the difficulties posed by PIN task, highlighting the need for further research on the topic.

Table 5.1: Comparison of the different object-driven datasets for embodied navigation, considering the photo-realism of scenes and targets, the availability of additional objects with variable spawn locations, the modalities of the provided references, and whether the dataset is instance-oriented.

Dataset	Photo-Realistic Scenes	Photo-Realistic Targets	Additional Objects	Visual Reference	Descriptive Reference	Variable Placement	Instance Goal
MP3D [40]	✓	✓	✗	✗	✗	✗	✗
AI2-THOR [157]	✗	✗	✓	✗	✗	✓	✗
Gibson [358]	✓	✓	✗	✗	✗	✗	✗
Robo-THOR [78]	✗	✗	✓	✗	✗	✓	✗
MultiON* [344]	✓	✗	✓	✗	✓	✓	✗
HM3D [253]	✓	✓	✗	✗	✗	✗	✗
ProcTHOR	✗	✗	✓	✗	✗	✓	✗
ION [180]	✗	✗	✓	✗	✓	✓	✓
THDA [213]	✓	✓	✓	✗	✗	✓	✗
ZSON [212]	✓	✓	✗	✗	✓	✗	✗
InstanceImageNav [159]	✓	✓	✗	✓	✗	✗	✓
ZIPON [76]	✓	✓	✗	✗	✓	✗	✓
GOAT-Bench [150]	✓	✓	✗	✓	✓	✗	✓
PInNED (Ours)	✓	✓	✓	✓	✓	✓	✓

5.1.1 Task

In this section, we outline the Personalized Instance-based Navigation task, highlighting its key characteristics and comparing it to existing embodied tasks. Following, we detail the composition and generation process of the PInNED dataset.

5.1.1.1 Task Definition

The PIN task requires the agent to navigate toward a predetermined specific object instance (*e.g.*, "*a yellow backpack with red straps*") in an unexplored environment. Each target object needs to be found in the environment, distinguishing it from multiple distractors of the same category and other objects of different categories. In this setting, the target object can be provided to the agent in two different modalities: (i) as a set of RGB images depicting the target object rendered in an isolated context on a neutral background, and (ii) as a set of textual descriptions of the object instance appearance.

At the beginning of each episode of PIN, the agent is initialized at a random pose $\boldsymbol{x}_0$ in an unseen environment. A single target instance o^i is selected as the goal g, such that $g \in C^a \subset O$, where C^a is a set of instances belonging to the same object category and O is the set of all available objects. The goal g is placed in the environment at a position $\boldsymbol{z}$. Additionally, n distinct instances o^j ($o^j \in C^a \wedge i \neq j$) are positioned in the environment, along with m distinct

instances o^k ($o^k \in (O \setminus C^a)$). At the end of the episode, the navigation is considered successful if the agent selects the '*stop*' action before the maximum allowed number of timesteps T, with an Euclidean distance between the position of the agent at the current timestep $\boldsymbol{x}_t$ and the target position $\boldsymbol{z}$ lower than 1 meter. The action space of the agent for the task is defined by six possible actions, where at each timestep t, the action $a_t \in \{$'*stop*', '*move ahead*', '*turn left*', '*turn right*', '*tilt up*', '*tilt down*'$\}$.

Comparison with Other Tasks. The proposed task locates itself among PointNav [6], ObjectNav [6, 17], ImageNav [62], and the recently defined task of InstanceImageNav [160]. PIN exhibits similarities to ObjectNav, InstanceImageNav, and the recently introduced GOAT-Bench [150]. However, it diverges from the traditional ObjectNav task because, differently from the standard objective of finding any instance of a general object category, PIN requires locating a specific instance, such as "*black and white striped trekking backpack*" instead of any "*backpack*". PIN leverages zero-shot properties at the instance level, as the object instances used for the training split differ from those included in the validation episodes. This requires agents to focus on the specific characteristics of the target object defined by the input references and avoid being misled by instances of the same category that are not the actual target.

Furthermore, PIN differs from InstanceImageNav and GOAT-Bench in various aspects. First, the target object is represented by a collection of images with neutral backgrounds, rather than being shown in its current spatial context. InstanceImageNav and GOAT-Bench are based on a set of general object categories that are included in the dataset of scenes and, therefore, these objects are static and frozen in the 3D model of the environment. Instead, the peculiarity of PIN is that it is created using a set of additional photo-realistic personal objects from a collection of 3D objects that can be placed and moved in different locations of the environment between different episodes. Using additional objects allows to avoid reconstruction errors and artifacts that can distort the appearance of the target. This unique characteristic compels the agent to discern and extract the defining features of the target object while maintaining invariance to the surrounding context in which it is situated since personal objects can be moved frequently and could be placed in multiple suitable locations.

Similarly to GOAT-Bench, PIN provides a multimodal input to the agent, including textual descriptions of the target instances alongside the images. However, GOAT-Bench ignores the presence of instances of the same category of the target in the scene, whereas this is the core challenge of PIN. Additionally, it is worth noting that while text alone can sometimes provide precise identification

of the specific instance, it can also be ambiguous. Visual references, although generally clearer, are not always available in real-world scenarios. Therefore, the two modalities cover different real-world requirements and both deserve to be studied. An extensive comparison of current object-driven dataset properties is reported in Table 5.1, which presents the following columns:

- *Photo-Realistic Scenes*: the presence of photo-realistic scans taken from real-world environments (e.g. the scenes of HM3D are built from scans of real environments, while scenes in AI2-THOR are hand-built by professional 3D artists);
- *Photo-Realistic Targets*: the availability of photo-realistic objects that can be used as navigation targets. In PInNED we carefully selected objects with realistic appearances. Procedurally-generated datasets, instead, tend to favor customizability over realism;
- *Additional Objects*: the inclusion of target objects that were not present at the time of capture. Datasets like GOAT-Bench target objects which were already present in the acquired scene, while PInNED targets objects injected in the scene afterward;
- *Visual Reference*: providing visual target references for each navigation episode;
- *Descriptive Reference*: providing natural language descriptions as targets for each episode;
- *Variable Placement*: the possibility of having variable spawning positions for the targets within the dataset;
- *Instance Goal*: the inclusion of navigation episodes in which the goal is to reach the exact instance indicated to the agent.

Moreover, a qualitative comparison of goal objects observed in their position in the environment from different datasets is depicted in Fig. 5.2.

Comparison with ProcTHOR. ProcTHOR [80] is a framework built on AI2-THOR [157] to procedurally generate interactive environments, enabling the evaluation of data augmentation and large-scale training in different Embodied AI tasks. PInNED is a dataset designed specifically to study the newly introduced PIN task, in which the agent is tasked with finding a specific instance according to target images or textual descriptions.

Table 5.2: Configuration of the main parameters used for executing each episode of the PIN task contained in the PInNED dataset.

Action Space		***Episode Configuration***		***Depth Sensor***	
forward step	0.25	*success distance*	1.0	*width*	360
turn angle	30	*max episode steps*	1000	*height*	640
tilt angle	30	***RGB Sensor***		*hfov*	42
Agent Configuration		*width*	360	*position*	[0, 1.31, 0]
visual sensors	rgb, depth	*height*	640	*min depth*	0.5
height	1.41	*hfov*	42	*max depth*	5.0
radius	0.17				
position	[0, 1.31, 0]				

ProcTHOR includes 1,633 instances across 108 object categories, with the ability to vary brightness, colors, materials, and object states. These categories include several household objects, covering generic objects, such as '*pen*' or '*apple*', objects that can be personal, such as '*mug*' and '*watch*', and large objects that are unlikely to change their placement in the environment, such as '*fridge*', and '*window*'. PInNED presents 18 object categories that can be personal, with the specific purpose of accompanying the task in which the agent has to distinguish instances belonging to the same category. All the categories represent objects that can be moved frequently in the environment and do not have a predefined location.

As well as most procedural datasets, ProcTHOR sacrifices realism in favor of interactivity, scalability, and customizability. PInNED, as a task-specific dataset, favors photo-realistic environments and objects. Indeed, it is the first instance-based navigation dataset based on both photo-realistic environments and injected objects, that can be moved frequently and with multimodal targets. Interactivity with the objects is out of scope for this work, however, the addition of external objects paves the way for possible future enhancements where interactivity is needed.

Configurations. Table 5.2 presents the relevant hyperparameters employed for executing each episode of the PInNED dataset. The configuration used for a PIN episode comprises a maximum duration of $1,000$ time steps, with the agent's action space defined by discrete forward steps of 0.25 m, a turn angle of $30°$, and a head tilt angle of $30°$. Each episode is considered successful if the position of the agent is within 1 meter from the position of the target object, and it predicts the '*stop*' action before the end of the time step budget. The configurations used for the navigation experiments reflect the settings employed to simulate the camera

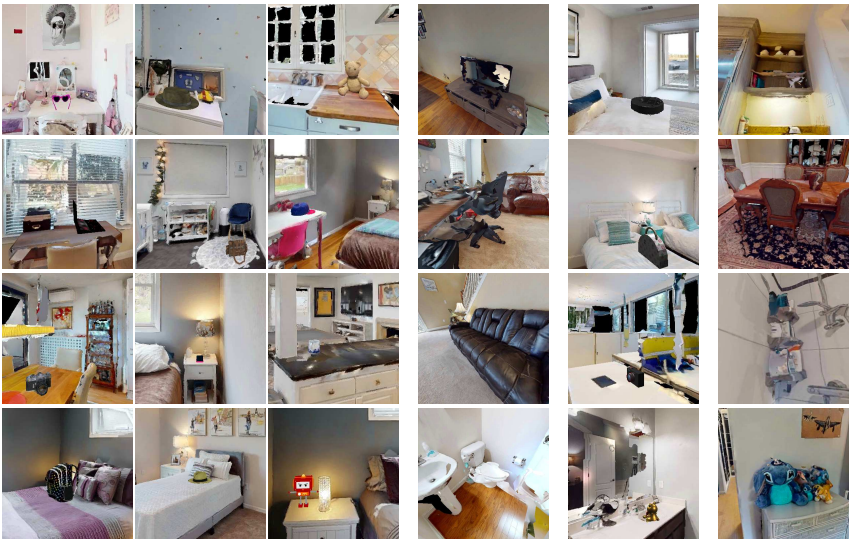

Figure 5.2: Comparison of observations depicting different target objects of PInNED dataset with the target objects of InstanceImageNav, MultiON, and GOAT-Bench datasets.

sensors and space occupation of the HelloRobot Stretch[1] platform.

5.1.1.2 Dataset

Categories and Instances. We selected a pool of 18 object categories from the assets contained in Objaverse-XL dataset [79]: '*backpack*', '*bag*', '*ball*', '*book*', '*camera*', '*cellphone*', '*eyeglasses*', '*hat*', '*headphones*', '*keys*', '*laptop*', '*mug*', '*shoes*', '*teddy bear*', '*toy*', '*visor*', '*wallet*', '*watch*', for a total of 338 additional objects. Each category contains an average of 18.8 objects, with a standard deviation of 5.5. The 3D objects are selected with human supervision to ensure photo-realism and uniqueness, which are critical requirements for tackling the PIN task. Finally, the 3D models of the objects are manually rescaled to have comparable dimensions to their real-world counterparts. In this procedure, we

[1] https://hello-robot.com/stretch

Table 5.3: Statistics about the number of distractors placed in the episodes of the training and validation sets of PInNED dataset. We consider the distractors belonging both to the same category of the target and to other categories.

# of Distractors	*Same Object Category*		*Other Categories*	
	Train	**Val**	**Train**	**Val**
Max	6	3	13	10
Average	2.93	2.90	7.75	7.19
Standard Deviation	0.33	0.37	2.84	2.82

rendered each given object in a scene from HM3D and varied the scale of the object until the result was realistic according to our judgment. Hence, each of the 338 additional objects has a manually fixed scale that is adopted when the object is injected into the navigation episodes.

Input References. The input images for each target personalized object are generated by rendering the 3D mesh of the object in an isolated setting. Specifically, the input images are not expected to match the camera specification of the navigating agent [160]. The digital camera undergoes a 30-degree yaw rotation to capture a favorable perspective of the objects. Each instance is then rotated 180 degrees in yaw to view its reverse side, followed by a 90-degree pitch rotation to observe the object from above. This procedure produces a set of three input images for each target object. An illustration of the acquired reference images is displayed in Fig. 5.3. Moving on to the textual references, manually annotated descriptions are produced for each target personalized object with the scope of highlighting the details that allow the agent to distinguish it from other instances of the same category. Specifically, we provide three descriptions for each personalized object in the PInNED dataset. To annotate the descriptions, we provided two object instances at a time to the annotators, asking them to describe one of the two objects in such a way that it is distinguishable from the other. This procedure results in a total of 960 unique words and an average of 10.7 words per description.

Additionally, we present samples including both visual and textual modalities for the input references associated with some of the object instances of PInNED dataset in Fig. 5.12 and Fig. 5.13. In particular, we show the three views composing the set of visual references and the three manually annotated descriptions for the textual references.

Scenes. The benchmark defined by the PIN dataset is situated in the indoor photo-realistic scenes (*e.g.*, apartments, offices, houses) within the semantically-

Figure 5.3: Sample visual references of personalized targets from PInNED dataset. We include three instances from each object category used in the benchmark.

annotated subset [373] of Habitat-Matterport3D (HM3D) [253] which consists of 145 environments for the training split and 36 for validation set. However, one validation scene is ignored as it represents an art gallery and has no suitable spawnable surfaces. HM3D was selected due to its status as the largest publicly available dataset of semantically annotated indoor spaces with photo-realistic quality for embodied navigation.

Episode Generation. During the generation of the dataset, the bounding boxes

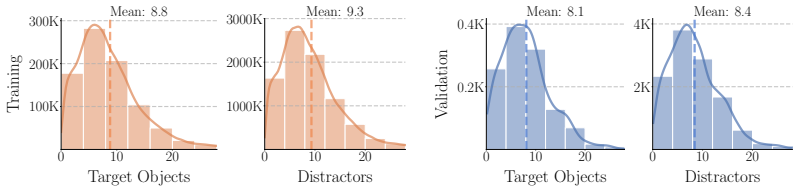

Figure 5.4: Plots of the distance statistics for the splits of PInNED dataset. The episodes of the training (orange) and validation splits (blue) are presented in terms of geodesic distance from the start position to the target object (left) and to all the distractors (right). All the distances are plotted in meters, and the mean value of each plot is shown on top.

of the surfaces in the environment are extracted using the semantic annotations of the scene. To obtain the bounding box from the texture, we extracted the point cloud 3D model of each scene and ensured that each point retained its associated annotation color. Subsequently, points were clustered by annotation color to create the bounding box associated with each piece of furniture. The spawning position of each object is selected by sampling from the positions of a curated set of suitable surface macro-categories included in the semantic annotations of HM3D. The surface categories selected for the creation of the dataset are: *armchair, bed, bench, cabinet, piano, rug, sofa, table*. These specific surfaces are chosen because of the high probability of personalized objects being positioned on top.

In each episode of the PInNED dataset, a single instance of a specific category is chosen as the target object. Consequently, up to 6 instances belonging to the same category, and up to 13 objects from other categories, are added to the environment as distractors. All additional objects placed in the environment are constrained to be on the same level/floor as the agent by selecting spawnable surfaces with a bounding box position within 0.5 meters from the starting position of the agent along the vertical axis. For each environment in the training split, a set of 400 episodes is sampled for each one of the possible categories. For the generation of the validation split, each target category is used twice. Finally, episodes where the target object is not reachable by an agent following the shortest path are removed from the dataset. The resulting dataset for PIN is defined by a total of $865,519$ generated episodes for the training split, while the validation split contains $1,193$ episodes.

In Table 5.3, we provide statistics on the number of distractors placed in the training and validation episodes of PInNED dataset. During the generation of PIN

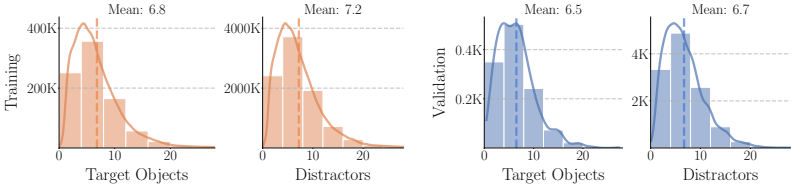

Figure 5.5: Euclidean distances of the objects included in the episodes of training (orange) and validation (blue) splits of PInNED dataset. The plots consider the distances from the start position to the target object (left) and to all distractors (right). Distances are measured in meters, with the mean value for each plot displayed at the top.

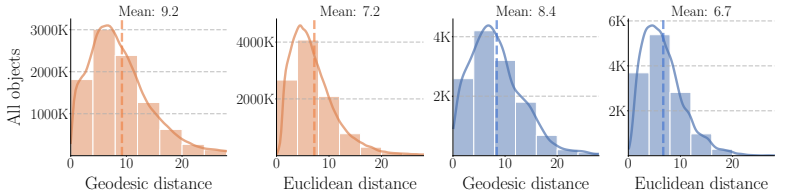

Figure 5.6: Plots of the geodesic and Euclidean distances for all the objects placed in the episodes of PInNED dataset. Training (orange) and validation splits (blue) are presented in terms of distances from the start position to all the spawned additional objects. All the distances are plotted in meters, and the mean value of each plot is shown on top.

episodes, a maximum number of distractors, both from the same category as the target instance and from other categories, is sampled from the set of available objects. The final number of additional objects in each episode is determined by the number of suitable surfaces and the available space on these surfaces. During the dataset generation process, objects are positioned above these surfaces and lowered until they contact the surface. If an object cannot be initially placed due to size constraints or collisions with other elements or walls, the placing process for that object is aborted, and another one is sampled from unused object instances. After the generation of the dataset of episodes, an additional assessment is performed through the Habitat simulator to remove the episodes containing objects that are not reachable from the starting position of the agent.

Object Distances. The geodesic distances of the target and distractors from the

Table 5.4: Ratio of the part of the sample images depicting each target object category in comparison with the area of the full image and the percentage of activations of the open-set object detector. The perthousand object/image ratio is computed using both the segmentation mask and the bounding box of each target objects. The activation ratio of the object detector is instead computed as a percentage.

Category	‰SegMask	‰BBox	%Act	Category	‰SegMask	‰BBox	%Act
Backpack	12.76	16.56	48.55	**Keys**	0.16	0.34	0.17
Bag	8.10	12.81	47.26	**Laptop**	7.20	12.25	56.84
Ball	4.54	5.96	33.33	**Mug**	1.28	1.54	1.33
Book	3.73	5.06	16.03	**Shoes**	3.49	5.53	19.35
Camera	1.99	2.59	4.76	**Teddy Bear**	12.36	20.78	78.85
Cellphone	0.75	0.80	1.28	**Toy**	4.30	9.68	13.78
Eyeglasses	0.49	1.22	3.69	**Visor**	2.71	4.57	0.64
Hat	3.34	5.97	11.60	**Wallet**	0.85	1.05	2.05
Headphones	3.36	5.84	11.72	**Watch**	0.30	0.52	0.80

starting position of the agent in the episodes of PInNED are shown in Fig. 5.4. In the figure, the distribution of the distances of targets and distractors significantly overlaps, hence prior information on the target object distance is hardly exploitable. Fig. 5.5, insteadm presents a plot depicting the Euclidean distances of target objects and distractors from the starting position of the agent in the episodes of both training and validation splits of PInNED datase, showing a consistent distribution of the distances of the additional objects with the geodesic distances. Furthermore, the plots of the distances of all additional objects (target instances and distractors) are presented in Fig. 5.6.

Object Selection and Distribution Criteria. The scope of PIN is to provide a benchmark to evaluate an agent tasked with finding a specific object that can be located anywhere in an unexplored environment, where distractors of the same category are present; hence, the object categories are selected according to the following criteria: (i) objects that are highly customizable in terms of shapes, colors, and other visual aspects, (ii) objects that are frequently moved and can be placed anywhere, and (iii) objects of common use for which is reasonable to ask a robot to find.

Object Size Analysis. Taking into account that personalized objects are defined as predefined instances with distinct characteristics, the primary challenge in the PIN task lies in effectively recognizing these specific details, especially when dealing with subtle features and limited interaction capabilities within the environment. In

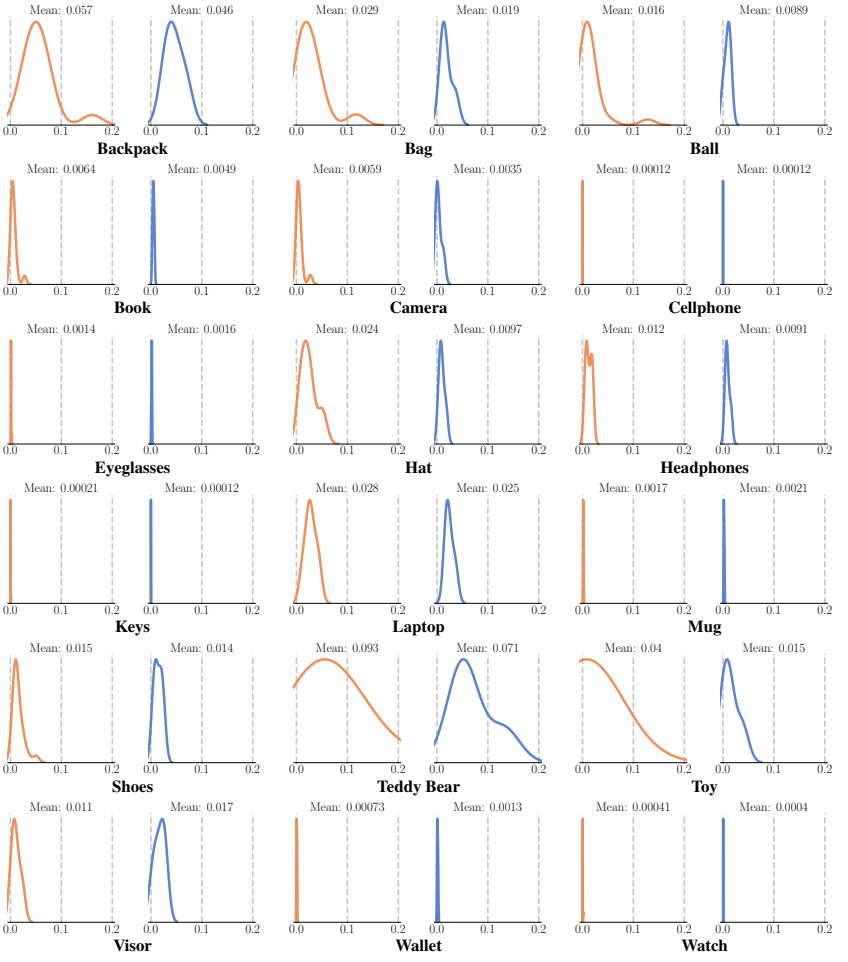

Figure 5.7: Distribution of the volumes in meters of the bounding boxes of the objects in PInNED dataset. Two plots are shown for each semantic category, reflecting respectively the objects of training (orange) and validation (blue) splits. Each plot is accompanied by the corresponding mean of bounding box volumes of the objects in each split.

this analysis, we present a category-wise size analysis of the objects in the dataset by computing and measuring the 3D bounding box of each object. In Fig. 5.7, we plot the distribution of the volumes of the bounding boxes associated with each object category, showing that the distributions between training and validation splits remain consistent.

Surfaces Details. As described in Sec. 5.1.1.2, the spawning position of each object in the PInNED dataset is selected by sampling from the positions of a curated set of suitable surface macro-categories included in the semantic annotations of HM3D. The surface categories selected for the creation of the dataset are: *armchair, bed, bench, cabinet, piano, rug, sofa, table*. These surfaces are valid for all the object categories and there are no subsets of surfaces dedicated to specific categories. There are categories, especially '*shoes*', that are unlikely to be placed on certain surfaces. However, the scope of the task is to have objects that could be placed everywhere and teach a robotic agent to find them. A teddy bear is not necessarily located on the bed, but could be located anywhere, even on the kitchen table. If we assume a real-world scenario in which a child forgets the teddy bear on the kitchen table, the agent should not go directly to the bedroom, but look for the object in the whole environment. This is the reason for which we adopted a consistent spawning mechanism across all the categories. We identify this combination of objects that could be placed everywhere and the consistent spawning mechanisms as the correct approach for providing a dataset covering a large set of possible real-world scenarios that avoid the exploitation of prior knowledge on the object placement.

In Fig. 5.8, we showcase the occurrences of the suitable surfaces in the environments of HM3D [253]. Notably, the distribution of spawnable surfaces remains consistent between the training and validation splits. This implies a recurring pattern in the furnishing of indoor spaces contained in the HM3D dataset and used for the PIN task.

5.1.2 Baselines

In this section, we present the set of approaches that are revisited and tested on our introduced PInNED dataset. These methods are recent object-driven methods and can be grouped into two categories: (i) **modular agents** that decouple the navigation task into specialized sub-modules and (ii) **end-to-end agents** based on a monolithic policy trained using reinforcement learning. Fig. 5.9 shows an overview of these two approaches.

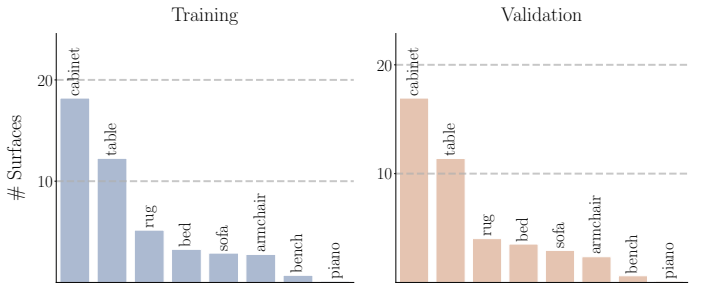

Figure 5.8: Plot of the mean number of surfaces in each environment that are suitable for object placement in the training (left) and validation (right) splits of the PInNED dataset.

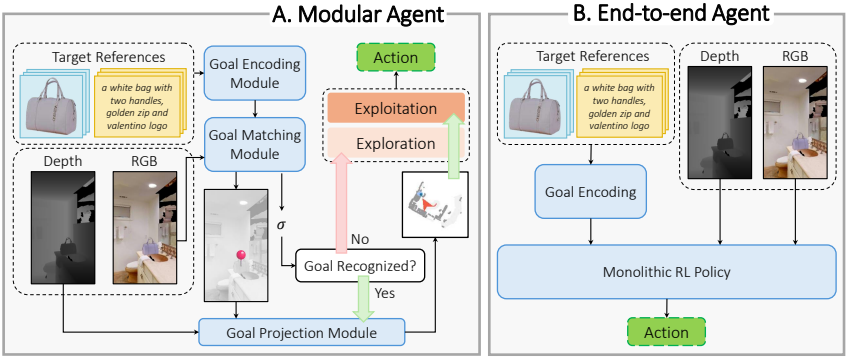

Figure 5.9: Overview of the baselines designed for the PIN task: modular agent (on the left) and end-to-end agent based on a monolithic reinforcement learning-based policy (on the right).

5.1.2.1 Modular Agents

In recent years, modular agents gathered an increasing interest in various embodied settings. These agents tackle the high-level navigation tasks by decoupling them into a chain of specialized sub-modules, each of which handles a smaller task. Specifically, Chaplot *et al.* [42] proposed SemExp, a modular agent designed for the ObjectNav task composed of three main modules: exploration, object detection, and exploitation. The core idea is that the agent explores as much as possible the unseen environment while the detection module localizes the semantic objects in

the acquired observations. Inspired by this approach, Mod-IIN [159] and CLIP on Wheels (CoW) [105] adapt the detection module to handle specific instances and open-vocabulary targets, respectively. For our modular agent baselines, we consider the same exploration and exploitation modules used in these previous works, while changing and adapting the object detection module for the PIN task.

Exploration Module. The exploration module is entitled to explore the unseen areas of the environment with the scope of encountering the target object. As in Mod-IIN and CoW, we adopt a frontier-based exploration (FBE [374]) approach. The agent builds an occupancy map of the environment during navigation, and at every time step, if the goal is not detected, the unexplored frontier on the map which is closest to the agent is selected as the current goal.

Object Detection Module. The object detection module receives the visual or textual references and the current RGB observation of the agent. Then, it is tasked with providing (i) a **matching score** that, whether it exceeds a certain matching threshold σ, determines that the goal has been recognized; and (ii) a **series of coordinates** on the observation which correspond to where the goal is located, that are used by the exploitation module to project the goal on a 2D map. We select three categories of approaches to implement this module:

- **Keypoint Matching**: In this category, the visual target references and the current RGB observation are provided to a keypoint matching method. We tested SuperGlue [273], following the approach proposed by Mod-IIN [159], and the framework introduced in IEVE [170]. In particular, SuperGlue outputs a confidence score for each matched keypoint pair. We use the sum of these confidences as the matching score and the keypoints that exceed a given confidence threshold τ as the localization coordinates. Regarding the Exploration-Verification-Exploitation framework proposed in IEVE, we adapted some components to match the different requirements of our task. Specifically, we first collected an auxiliary dataset, which includes, for each goal in the training set, 10 positive samples and one negative sample containing a distractor from the same category as the goal. We trained InternImage [336] to classify the 18 categories of our dataset using the goal images of the training set. Instead of the InternImage segmentation model, since, to the best of our knowledge, no segmentation dataset contains all our categories, we adopted the open-vocabulary segmenter GroundedSAM [261]. For the image-matching step, we exploited LightGlue [193] on the keypoints extracted with DISK [318] as in the original IEVE paper.

- **Patch-level Matching**: A Vision Transformer (ViT [91]) encoder divides an

image into patches and extracts patch-level embeddings. Hence, we extract a goal embedding from each reference and compute the cosine similarity with the patch-level feature vectors of the RGB observation. If at least a patch has a similarity that exceeds the matching threshold σ, the goal is considered detected. The center coordinates of these patches are used as the goal localization result. For the visual references, we employ DINO [37], DINOv2 [234], and CLIP [249] performing a region pooling over the reference objects to obtain goal feature vectors. For the textual references, a text-aligned multimodal encoder is required. Hence, we employ CLIP and, inspired by [105], CLIP with gradient relevance [46] (CLIP-Grad). We assume the mean embedding of the set of prompt templates used in CoW applied to the target descriptions as the target feature vector.

- **Detection Model**: We consider detection models that produce output regions according to a given reference. Specifically, we consider PerSAM [409] (both in the standard and one-shot finetuned versions) and OWL [221], which localize regions according to, respectively, visual and textual references. As in CoW, we exploit the output confidence to determine whether the goal has been detected and return the central coordinates of the region as the goal localization result.

Exploitation Module. The exploitation module takes control of the navigation when the goal is recognized in the current observation. After detecting the target object at a given location, the exploitation module is triggered and computes the route to reach the target object. The goal position provided by the object detection module is projected into an occupancy map, and the Fast Marching Method [43, 277] is used to plan the path from the current position of the agent to the detected goal position. When the agent reaches the goal position, the '*stop*' action is called to conclude the episode.

5.1.2.2 End-to-End Agents

In contrast to modular agents, end-to-end approaches train a neural network policy to process sensor input and predict the atomic actions needed to complete the required task. We consider two recent approaches for embodied navigation and adapt them for the Personalized Instance-based Navigation task: (i) ZSON [212], which pre-trains an ImageNav agent and evaluates downstream on ObjectNav leveraging the capabilities of CLIP multimodal embeddings; and (ii) RIM [53],

which employs a Transformer-based architecture [322] that is trained using auxiliary tasks and uses a recursive implicit map that is updated during the navigation for the ObjectNav task. We finetune both approaches on PInNED dataset. Specifically, ZSON is adapted to use image references as input during its ImageNav pretraining phase. While, for RIM, we employ two finetuning strategies: conditioning the navigation on textual features extracted from the reference descriptions and conditioning on visual features extracted from the image references. The features produced using both modalities of PInNED references are extracted using CLIP.

5.1.3 Experiments

In this section, we present the implementation details and an experimental analysis of the selected baselines on the PIN task, discussing the set of metrics used to effectively evaluate the performances and the obtained results.

5.1.3.1 Implementation Details

Modular Agents. The ability of modular agents to distinguish a specific instance in a given observation depends on the score threshold that maximizes the detection results. We tune this threshold on a subset of the training episodes. For all the backbones except for SuperGlue [273], we extract two squared crops with size 360×360 from the 360×640 observation and resize them to the image resolutions on which the backbones have been trained. Then, we consider all the matches resulting from the two crops. At least a match over the threshold is required to consider the goal detected in an observation. For the textual modalities, we employ the 80 prompt templates proposed by Radford *et al.* [249] for ImageNet [81]. In this section, we report additional implementation details for each backbone.

- **SuperGlue [273]**: We observe that SuperGlue struggles to match the visual references with the observations of the agent and that the resolution of the references influences the matching capabilities. In particular, we provide the visual references to SuperGlue as squared images 360×360, corresponding to the shortest side of the observation of the agent. For each visual reference, namely for each of the three views of the object, we provide two resizes of the object such that the longest side is, respectively, 360 and 180. This procedure results in two reference images for each view of the object, an image entirely occupied by the object and an image where the object occupies a quarter of it. In Table 5.8 we show that this approach results in a higher success rate than having a single image per object view. Moreover, we employ the

indoor weights of SuperGlue with a threshold of 0.2 on the confidence of each matched keypoints pair and a matching threshold σ of 8.0 on the confidence sum of all the matched keypoints pairs.

- **CLIP [249]**: We employ CLIP ViT-B/16 with the pre-trained weights from OpenAI for both the experiments with visual and textual references. We resize the two observation crops to 224×224, resulting in a grid of 14×14 patches. The best matching threshold σ for the visual and textual modalities are, respectively, 0.575 and 0.28.

- **CLIP-Grad**: We follow the implementation of the network interpretability method proposed in CoW [105] on top of CLIP with textual references. We employ CLIP ViT-B/32 with the pre-trained weights from OpenAI and matching threshold 0.85.

- **OWL [221]**: OWL is an open-vocabulary detector that is trained in two steps: (i) a large contrastive image-text pre-training following LiT [399] and (ii) an object-level training on publicly available detection datasets (Open Images V4 [163], Objects 365 [279], and Visual Genome [161]). We employ a matching threshold of 0.25 applied to the predicted bounding box scores.

- **DINO [37]/DINOv2 [234]**: DINO is a self-supervised backbone pre-trained according to a self-distillation training paradigm. DINOv2 is an improved version of DINO with the aim of producing general-purpose visual features. We employ DINO ViT-B/16 and DINOv2 ViT-B/14 trained, respectively, on ImageNet-1k [81] and LVD-142M [234]. We use the same input image resolutions on which they are trained, namely 224×224 and 518×518, producing 14×14 and 37×37 grids of patches. The best matching scores are, respectively, 0.575 and 0.5.

- **PerSAM/PerSAM-F [409]**: We leverage the implementation of PerSAM on SAM ViT-B/16, trained on SA-1B, with input image resolution at $1,024$. PerSAM-F is a variant of PerSAM that fine-tunes the model on the reference image, We follow the training configuration of the original implementation. We consider the maximum patch-level similarity between the reference images and the observation crop as the matching score on which we apply the thresholds 0.925 and 0.61 for, respectively, PerSAM and PerSAM-F.

End-to-End Agents. As mentioned in Sec. 5.1.2.2 end-to-end approaches use a neural network policy which is trained end-to-end to directly process sensor

observations and predict the atomic actions needed to fulfill the required task. In our case, we adapted two recent end-to-end approaches for ObjectNav, finetuning them to perform PIN task: RIM [53] and ZSON [212].

- **RIM [53]**: The model is finetuned using behavior cloning following Chen *et al.* [53] approach and starting from the pre-trained weights for ObjectNav [17]. We evaluate two variants of the fine-tuned model, conditioned on visual features and conditioned on textual features. In RIM approach, besides the episodic implicit map that is updated recursively, the input of the policy at each timestep is composed of the concatenation of the features extracted from RGB and depth observation, the pose of the agent, previous action, and the target object category. To adapt RIM for the PIN task, we modify the features extracted from the object category label. Originally each label is associated with a row in a lookup table containing learnable embeddings of length 32. In our adaptation, we replace such embeddings with CLIP (ViT-B/16) features extracted using the visual or textual references. Since each input reference modality is described by 3 images or descriptions, we compute the mean of the features extracted from each reference. Following, a learnable linear layer is trained to project CLIP features to a vector of length 32. The resulting embedding is used to condition the navigation of the RIM agent. The fine-tuning process is performed on a single GPU for a total of $\approx$ 2M fine-tuning steps over ≈ 24 hours.

- **ZSON [212]**: For the adaptation of the ZSON method, we fine-tuned the model pre-trained on the ImageNav task, following the same approach as Majumdar *et al.* [212]. The agent is fine-tuned with reinforcement learning using an adaptation of ZSON reward but ignoring the angle to the goal since it is not a component considered in the PIN task. The resulting reward is $r_t = r_{success} - \Delta_{dtg} + r_{slack}$. We refer to Majumdar *et al.* [212] for a description of the components of the reward. Moreover, while the original approach uses ImageNav goals that are represented as photos captured at the position that the agent is required to reach, we used image references of the target instance to perform the fine-tuning. The model is fine-tuned on a single GPU for ≈ 24 hours for a total of $\approx$ 5M fine-tuning steps.

Compute Information. We performed our experiments on a computing platform composed of NVIDIA RTX5000 GPUs and 8 GB of CPU memory for each job. A job can be computed on a single GPU. Each episode step for the modular agents requires an average of $\approx$ 200ms to be executed. Hence, the entire DINOv2

Table 5.5: Navigation results on PInNED on the environments of HM3D dataset, considering the presence of distractors from the same category. **Bold** text denotes the best performance among each category of approaches.

			Navigation Metrics					Detection Metrics			
	Backbone	Modality	SR↑	SPL↑	CE↓	D2G↓	Steps	%Match↑	TM↑	CM↓	NM↓
Modular Agents											
CLIP [249]	ViT-B/16	Textual	3.10	1.82	9.31	7.94	503.1	62.95	20.07	22.07	57.86
CLIP-Grad [105]	ViT-B/32	Textual	4.53	2.42	6.95	7.94	465.8	77.95	4.65	7.21	84.14
OWL [105, 221]	ViT-B/32	Textual	7.29	3.36	12.66	7.90	871.7	22.97	**62.60**	32.88	4.52
SuperGlue [159, 273]	-	Visual	3.27	1.28	7.38	8.36	804.0	29.42	16.96	3.44	79.60
IEVE [170]	-	Visual	3.52	3.07	12.25	7.73	712.1	30.03	32.39	16.01	51.60
PerSAM [409]	ViT-B/16	Visual	2.77	1.81	6.53	8.20	362.5	**81.98**	1.15	10.43	88.42
PerSAM-F [409]	ViT-B/16	Visual	1.93	1.28	**5.63**	8.12	321.3	36.13	0.60	13.48	85.92
DINO [37]	ViT-B/16	Visual	4.02	1.71	6.88	8.28	826.0	23.89	62.73	**1.36**	35.91
CLIP [249]	ViT-B/16	Visual	9.64	5.39	13.33	7.79	623.5	58.51	32.53	16.35	51.12
DINOv2 [234]	ViT-B/14	Visual	**14.84**	**7.94**	26.15	**7.28**	658.7	55.74	55.33	42.00	**2.67**
End-to-end Agents											
RIM [53]	ResNet-50	Textual	7.12	6.67	**10.44**	8.43	409.3	-	-	-	-
RIM [53]	ResNet-50	Visual	8.80	6.80	13.41	8.48	402.1	-	-	-	-
ZSON [212]	ResNet-50	Visual	**9.14**	**7.18**	21.12	**7.00**	389.9	-	-	-	-

experiment on the $1,193$ validation episodes, with an average number of steps equal to 658.7, requires ≈ 44 computation hours. The entire evaluation on the validation split for the end-to-end agents requires ≈ 5 computation hours.

5.1.3.2 Evaluation Metrics

Traditional metrics for object-driven embodied navigation are **success rate** (SR) and **success rate weighted by path length** (SPL). SR is the ratio between the number of episodes where the agent successfully reaches the target object within a maximum distance of 1 meter and the total number of episodes, while SPL weighs the success rate with the length of the path taken by the agent. Moreover, we report the **average number of steps** taken by the agent and the **average distance from the goal** (D2G) at the end of each episode. The agent designed for tackling the PIN task should be able to distinguish whether the target object is present in the current observation while exploring the unseen environment and correctly localize it, within the timesteps budget T (set to 1,000). The main challenge is represented by distractor instances belonging to the same category as the target object. Hence, we introduce the **category error** (CE) metric, which measures the percentage of episodes in which the agent stopped within one meter from instances belonging to the same category of the goal.

Table 5.6: Navigation results on PInNED on the environments of HM3D dataset, without considering the presence of distractors from the same category of the target. **Bold** text denotes the best performance among each category of approaches.

			Navigation Metrics				Detection Metrics		
	Backbone	Modality	SR↑	SPL↑	D2G↓	Steps	%Match↑	TM↑	NM↓
Modular Agents									
CLIP [249]	ViT-B/16	Textual	3.35	1.86	8.01	516.5	**61.86**	22.83	77.17
OWL [105, 221]	ViT-B/32	Textual	8.22	3.18	7.88	929.9	13.83	93.91	6.09
CLIP [249]	ViT-B/16	Visual	11.15	5.92	7.65	666.2	52.56	35.57	64.43
DINOv2 [234]	ViT-B/14	Visual	**23.13**	**11.61**	**6.62**	784.5	38.64	**96.09**	**3.91**
End-to-end Agents									
RIM [53]	ResNet-50	Textual	7.46	6.87	7.94	487.1	-	-	-
RIM [53]	ResNet-50	Visual	10.35	7.53	7.75	475.9	-	-	-
ZSON [212]	ResNet-50	Visual	**10.39**	**8.00**	**6.91**	460.1	-	-	-

In modular agents, the ability to detect the correct instance resides in having large matching scores when the target is present in the observation and small scores when the target is absent. Since in these agents it is possible to determine whether a given observation matches, we compute four additional metrics: the **percentage of episodes with at least a detected match** (%Match), the **percentage of matched observations** that contain the **target object** (TM), an **instance of the same category of the target** (CM), or **no relevant objects** (NM).

5.1.3.3 Experimental Results

Personalized Instance-based Navigation Experiments. In Table 5.5, we present the results on the PIN task. Among modular agents, DINOv2 performs best according to SR and SPL. The high values of TM, CM, and CE show that the obtained matches usually refer to the same category of the target instance. The same reasoning can be applied to OWL for the modular agents using textual references. However, OWL produces fewer matches as can be noted from the %Match metric. Models such as SuperGlue, PerSAM, and PerSAM-F, which exhibit low SR and TM, have also a corresponding high NM, demonstrating that they are not able to provide significant matching scores for distinguishing the correct instances or even the correct categories. It is noteworthy that SuperGlue struggles to match the instances of PInNED, which are represented on a neutral background, contrary to InstanceImageNav [159], where the reference image is a photo of the object in the same context in which it is located. Regarding PerSAM and PerSAM-F, the results show that the feature space of SAM [153] is not

informative enough to understand whether an instance is present in an observation. IEVE shows an improvement with respect to the other image-matching modular agent, based on SuperGlue. This is motivated by the fact that IEVE, differently from other image-matching approaches, combines LightGlue with a semantic detector, allowing the agent to focus only on observations that contain objects of the target category. This behavior is confirmed by the increased numbers of target matches, category matches, and category errors.

Moreover, end-to-end agents tend to perform worse than modular agents. This can be attributed to the imitation training performed using the ground-truth trajectory to the goal. Since in the PIN task the target instances can be placed in multiple locations, it is not possible to exploit prior semantic knowledge about the estimated location of the target instance. Moreover, end-to-end agents tend to struggle in backtracking and in recovering the navigation when moving in the wrong direction. This behavior can also be noted from the path length, which for end-to-end agents is shorter than modular agents, that continue the exploration until the whole environment is observed.

Ablation on Category Distractors. In Table 5.6, we introduce an ablation study in which we remove the distractors belonging to the same category of the target instance. Overall, metrics for all the agents improve because the presence of these distractors represents the core challenge of the PIN task. In particular, DINOv2 improves by 8.29 with respect to the main experiments, demonstrating that it embeds strong semantic correspondence properties among the same category, but that it is not trivial to identify a threshold that clearly distinguishes specific instances. The impact of same-category distractors on end-to-end agents is minor since they are finetuned to identify the correct instance.

Modular Agent Activations. In Fig. 5.10 we present a comparison of the similarities computed between the patch-level features of different backbones on the observations of the agent and the references. In particular, we show these similarities on DINOv2 [234], DINO [37], CLIP with visual references, and CLIP with textual references [249]. The resolution of the similarities extracted from DINOv2 is higher than the others since we employed the input resolution 518×518 on which the ViT-B/14 model has been trained, which corresponds to a grid of 37×37 patches, whereas DINO and CLIP are based on a ViT-B/16 backbone with 224×224 as input resolution. It is noteworthy that DINOv2 exhibits strong semantic localization properties, with high similarity values on the exact location of the image on which the target is observed. On the contrary, DINO and CLIP tend to exhibit less well-localized similarities. Moreover, CLIP with visual references has a high similarity on the patches corresponding to the laptop in the observation,

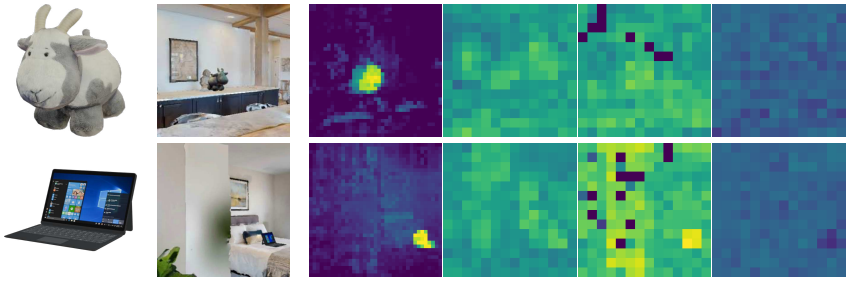

Figure 5.10: Comparison of the similarities between the patch-level features on two observations of an agent extracted with different backbones, DINOv2, DINO, CLIP with visual references, and CLIP with textual references, and the references. Purple values represent low similarity values, while yellow values represent high similarity values.

whereas CLIP with textual references has a low similarity on the same patches.

Category-wise Navigation Results. In Table 5.7 we present the navigation results of the modular agent based on DINOv2 as the matching backbone in which we compute the metrics for each category. From the results on SR and SPL we can note that there are categories that are easier to locate and reach, such as '*backpack*', '*bag*', '*ball*', '*hat*', '*laptop*', and '*toy*', and there are instances from categories that are never correctly reached, such as '*keys*', '*wallet*', and '*watch*'. This result returns the inability of the vanilla matching modules to distinguish these categories in the embodied setting. Moreover, we can observe that there is an overall positive correlation between SR and average category size, implying that small objects are particularly challenging to detect.

Similarity Analysis. The similarity of objects is a critical factor in the PIN task. The presence of distractors increases the challenge of the proposed task, as the agent must balance between being overly cautious and overly confident when identifying target instances. This trade-off is central to the effectiveness of the navigation approaches. In particular, concerning images as references of the target object, re-identification methods should be a robust solution against distractors due to considering the matching between keypoints instead of the semantic similarity between observation and reference. Indeed, in Table 5.5, the state-of-the-art re-identification method SuperGlue has a lower category error than

Table 5.7: Navigation results of the modular agent that employs DINOv2 as the matching module on the validation episodes of PInNED dataset, considering the performance of the agent for each category. Moreover, we report the average intra-category and inter-category cosine similarities computed on the frontal goal images.

	Navigation Metrics					**Detection Metrics**				**Similarity**	
Category	SR↑	SPL↑	CE↓	D2G↓	Steps	%Match↑	TM↑	CM↓	NM↓	Intra-Category	Inter-Category
Backpack	26.47	14.04	36.77	5.79	408.7	85.29	53.27	46.57	0.16	0.510	0.110
Bag	23.08	13.65	40.00	6.16	406.5	93.85	44.62	55.04	0.34	0.348	0.121
Ball	20.90	10.29	23.88	6.48	613.1	61.19	36.06	63.87	0.07	0.258	0.068
Book	19.40	10.83	35.82	5.71	484.3	86.57	58.51	40.16	1.33	0.613	0.106
Camera	7.46	3.38	7.50	8.57	883.2	20.90	69.23	23.08	7.69	0.152	0.050
Cellphone	8.96	3.11	14.92	8.63	844.8	32.84	7.81	90.96	1.23	0.506	0.112
Eyeglasses	10.45	5.08	32.83	7.70	682.0	62.69	79.80	19.95	0.25	0.846	0.104
Hat	26.87	11.95	23.88	6.45	652.8	67.16	88.08	11.89	0.03	0.549	0.084
Headphones	16.92	9.71	40.00	7.35	492.8	84.62	14.58	85.29	0.13	0.764	0.098
Keys	0.00	0.00	8.82	8.38	974.2	2.94	0.00	0.00	100.00	0.558	0.102
Laptop	21.54	11.50	49.23	7.01	455.3	93.85	16.86	82.60	0.54	0.348	0.084
Mug	10.61	4.47	10.61	8.10	911.8	22.73	92.00	4.50	3.50	0.298	0.073
Shoes	16.92	12.44	44.62	6.75	318.8	95.38	8.31	91.69	0.00	0.631	0.087
Teddy Bear	19.12	13.48	52.94	7.07	335.5	91.18	68.92	16.62	14.46	0.548	0.066
Toy	26.56	13.18	3.12	6.16	754.6	48.44	99.27	0.00	0.73	0.137	0.087
Visor	11.94	5.99	31.34	7.99	657.0	52.24	52.47	45.33	2.20	0.316	0.148
Wallet	0.00	0.00	6.15	8.39	985.3	1.54	0.00	0.00	100.00	0.282	0.105
Watch	0.00	0.00	7.69	8.39	999.0	0.00	0.00	0.00	100.00	0.566	0.102

DINOv2 and CLIP. However, it presents the worst results according to SR and SPL, showing difficulties in matching keypoints when observation and reference have discrepancies in appearance. For methods based on semantic features, the similarity threshold is the key element in balancing confidence and caution.

In Table 5.7, we report the average cosine similarities in the DINOv2 embedding space per category. In particular, we extracted the CLS token from each frontal goal image of the validation set and computed the cosine similarities against the other goal images from the same category (i.e. intra-category) and against goal images from different categories (i.e. inter-categories). The results show that the intra-category similarity presents a strong relation with the category error (CE) and category matches (CM) metrics. Indeed, the agent tends to mistake instances from categories with large intra-category similarity values, such as '*eyeglasses*', '*headphones*', and '*shoes*', while these mistakes are reduced in categories such as '*camera*' and '*toy*' that are characterized by a larger variability in their instances. When we adopt textual references as targets, the challenges concern how well multimodal spaces embed fine-grained details, and how similarity behaves accordingly. Previous work [20, 27] has shown that this challenge is non-trivial and still open.

Figure 5.11: Examples of situations in which detecting the target in the embodied environment is particularly challenging. We depict the frontal visual references of the target in the first row and a portion of an agent's observation containing the target in the second row.

Our dataset represents a further step in this direction, providing a benchmark to evaluate the capabilities of visual-language models in recognizing fine-grained details. Future works can exploit our training set to instruct the models to distinguish instances of the same category by focusing on adjectives and attributes.

Hard Detection Cases. In Fig. 5.11, we show four episodes in which detecting the target is particularly challenging. These targets belong, respectively, to the '*wallet*', '*camera*', '*watch*', and '*keys*' categories. Table 5.7 shows that these categories are the most challenging ones for the modular agent with DINOv2, which is the best-performing agent according to Table 5.5. Indeed, the categories '*keys*', '*wallet*', and '*watch*' all yielded no successful episodes. These objects are hard to detect even for a human, confirming how challenging the PIN task is.

Fine-Grained vs General Descriptions Comparison. In Table 5.9 we present an ablation study in which we compare the performance of the baselines with both fine-grained and general object categories. Specifically, we conducted the following experiments. For the modular agents based on CLIP and OWL as the matching module, we leveraged the general object category (e.g. backpack) instead of the fine-grained textual descriptions as navigation targets, while maintaining the same similarity threshold. The results on both CLIP and OWL present similar

Table 5.8: Navigation results of the modular agent that employs SuperGlue as the matching module on the validation episodes of PInNED dataset, considering different resize values of the visual references of the target provided to the matching module.

	Navigation Metrics					**Detection Metrics**			
Resize	SR↑	SPL↑	CE↓	D2G↓	Steps	%Match↑	TM↑	CM↓	NM↓
360	2.51	0.82	**7.05**	8.48	881.6	17.77	**43.76**	5.17	**51.07**
180	3.02	1.20	7.21	8.48	864.1	20.70	21.72	3.58	76.35
180, 360	**3.27**	**1.28**	7.58	**8.36**	**804.0**	**29.42**	16.96	**3.44**	79.60

Table 5.9: Navigation results on PInNED on the environments of HM3D dataset, comparing categorical and fine-grained textual modalities.

			Navigation Metrics					**Detection Metrics**			
	Backbone	Modality	SR↑	SPL↑	CE↓	D2G↓	Steps	%Match↑	TM↑	CM↓	NM↓
Modular Agents											
CLIP [249]	ViT-B/16	Categorical	3.52	2.75	10.23	7.98	148.1	95.47	5.12	15.73	79.15
CLIP [249]	ViT-B/16	Fine-Grained	3.10	1.82	9.31	7.94	503.1	62.95	20.07	22.07	57.86
OWL [105, 221]	ViT-B/32	Categorical	7.79	3.73	19.96	7.96	780.6	38.81	26.50	58.49	15.01
OWL [105, 221]	ViT-B/32	Fine-Grained	7.29	3.36	12.66	7.90	871.7	22.97	62.60	32.88	4.52
End-to-end Agents											
RIM [53]	ResNet-50	Categorical	4.61	3.78	14.25	9.23	336.0	-	-	-	-
RIM [53]	ResNet-50	Fine-Grained	7.12	6.67	10.44	8.43	409.3	-	-	-	-

behaviors: the number of successful episodes is slightly increased, but also the number of episodes in which the agent mistakes reaching distractors of the same category and the number of matches with them increased. Moreover, the reduction in the average number of steps indicates that similarities, on average, are higher. The increase in successful episodes is surprising but in line with the findings of previous works in the literature [20, 27], which demonstrate that current vision-language models struggle with fine-grained details. These results show that our work can help future works in the realization and evaluation of vision-language models with improved understanding capabilities of details. Concerning the end-to-end agent RIM, we trained the model on the CLIP embeddings extracted from the general object categories instead of the fine-grained textual descriptions. The results show a lower number of successful episodes and a higher number of episodes in which the agent reaches a distractor of the same category.

	*a yellow kanken backpack with yellow straps on the top* *a yellow monochrome kanken backpack* *a photo of a yellow backpack with a strap and red circle on the front*
	*a black camera bag with a handle and a mesh pocket* *a black camera bag with a buckle and a small silver plate* *a black camera bag with two red laces, a silver plate and a black buckle in the middle front*
	*a beach ball with alternated red, light blue and white slices* *an inflatable colored beach ball* *a beach ball with a multicolored design*
	*a stack of two books with a leather cover tied using a brown strap* *a brown book with yellowed pages with two straps and golden buckles on top* *two books tied together by a brown lace, with black leather covers and a red jurassic park logo*
	*a big black camera with a black handle and a wheel on the side* *a black cubic camera with a brown knob and a strap* *a kodak brownie hawkeye black flash camera, which is cube-shaped and has a black handle*
	*a blue smartphone with a white text on the back* *a blue phone with a black screen* *a cellphone with a gradient blue to purple color, two lenses, a fingerprint reader and the xiaomi mi logo on the back*
	*a pair of black squared eyeglasses with a golden plate on the arms* *a pair of sunglasses with a black frame and gold detail* *a pair of black thick eyeglasses with squared frame and golden hinges*
	*a sombrero with red details and a yellow stripe* *a straw hat with a yellow ribbon around it* *a sombrero with red elements on the brim and a yellow stripe with chiquito written multiple times on top*
	*a pair of black beats headphones* *a pair of headphones with a black band* *a pair of black headphones with the beats by dre logo on the ear cups and two gray lines on the headband*

Figure 5.12: Samples of visual reference images and textual reference descriptions of personalized targets from PInNED dataset.

	*a worn red key and a yellow keytag with a black text* *a yellow plastic tag with a red key* *a red rusty key and a yellow keytag with generator maintenance written on it*
	*a black and grey laptop with rgb keyboard* *a black and grey laptop with a alien head on the back* *a laptop having a gray top cover with an alien logo on the back, a black base panel and a rainbow colored keyboard*
	*a blue mug with a red fox logo on it* *a blue mug with a firefox logo on it* *a blue mug with the mozilla firefox logo, composed of a red fox around the globe, printed on it*
	*a pair of orange adidas running shoes* *a pair of orange adidas sneakers with black stripes* *a pair of orange running shoes with orange laces, black adidas stripes, and white outsoles*
	*a beige teddy bear with a red bandana on a wooden chair* *a teddy bear sitting in a wicker chair with a red bandana on its neck* *a cream-colored smiling teddy bear with a red scarf and sitting on a woven chair*
	*a black and white toy car with a number 2 on the front* *a black and white toy car* *a black toy race car, with white wheels, a number 2 painted on the side, and a ball replacing the drive*
	*a black htc visor with blue polka dots* *a blue rounded virtual reality headset with a black strap* *a htc visor having black bands and blue front side with light blue dots*
	*a black leather wallet with an orange plate* *a brown leather wallet with a button on it* *a dark brown leather wallet having an orange patch with fossil written on it*
	*a gold and grey watch with black leather strap* *a brown leather wallet with a button on it* *a rounded watch having a thick golden case, white dial and black leather band with a golden buckle*

Figure 5.13: Samples of visual reference images and textual reference descriptions of personalized targets from PInNED dataset.

Chapter 6

Multimodal Large Language Models for Visual Grounding

The introduction of the attention operation and the Transformer architecture [322] has enabled the creation of models capable of handling various modalities on an increasingly large scale. This advancement is largely attributed to the versatility of the operator and the adaptability of the architecture. Initially, this breakthrough was leveraged for language-specific models [28, 87] but quickly extended to support diverse modalities [178, 206] and facilitate their integration within unified embedding spaces [249]. Such representations have formed the foundations of the open-vocabulary paradigm in detection and segmentation. The surge in sophisticated Large Language Models (LLMs), and particularly in their capacity for in-context learning, has encouraged researchers to broaden the scope of these models to encompass multiple modalities, both as inputs and outputs. This expansion has led to the development of cutting-edge models such as GPT-4V [3] and Gemini [8], showcasing state-of-the-art performance.

The development of Multimodal Large Language Models (MLLMs) entails merging single-modality architectures for vision and language, establishing effective connections between them through vision-to-language adapters, and devising innovative training approaches. These methodologies are crucial for ensuring modality alignment and the ability to follow instructions accurately. Early MLLMs were primarily developed to tackle holistic image understanding tasks, such as image captioning, visual question answering (VQA), and visual dialogue. These

tasks focus on the comprehension of an entire image conditioned on natural language inputs and outputs, making them a natural extension of the capabilities of pretrained language models. However, the scope of MLLMs has rapidly expanded to encompass a wider range of complex and fine-grained vision-language tasks. These include visual grounding, image generation and editing, document analysis, robot navigation, and even autonomous driving, each requiring a precise alignment between textual instructions and visual content.

In this Chapter, we introduce the essential components and design principles behind MLLMs and focus in particular on the visual grounding task. We survey and categorize the current approaches to visual grounding using MLLMs, discuss their training methodologies and evaluation protocols, and compare their performance on standard benchmarks. Finally, we highlight how recent advances in self-supervised visual representation learning have been successfully adopted as visual encoders in this context, contributing significantly to the emergence of MLLMs with fine-grained understanding capabilities.

6.1 Multimodal Large Language Models

Large Language Models. [28] discovered that in-context learning, *i.e.*, prepending the prompt with a few examples demonstrating the desired output of an LLM [63, 126, 308], improves its performance, especially over unseen tasks. Generalization can be further enhanced by providing the LLM with the natural language description of the desired task for each training sample. This technique, called instruction-tuning [65, 137, 342, 343], turns out to be critical for aligning the behavior of an LLM with that of humans and currently empowers the most advanced LLMs, eventually boosted via reinforcement learning from human feedback (RLHF) [3, 11, 56, 236].

PEFT. When a pre-trained LLM needs to be adapted to a specific domain or application, parameter-efficient fine-tuning (PEFT) schemes represent an important alternative to training the entire LLM, since these strategies only introduce a few new parameters. Among these, prompt-tuning [117, 201, 171, 181] learns a small set of vectors to be fed to the model as soft prompts before the input text. Differently, LoRA [128] constrains the number of new weights by learning low-rank matrices. This technique is orthogonal to quantization methods such as QLoRA [86], which further decreases the memory footprint of the LLM compared to the usual half-precision weights.

Towards Multimodal LLMs. The development of MLLMs follows a similar path

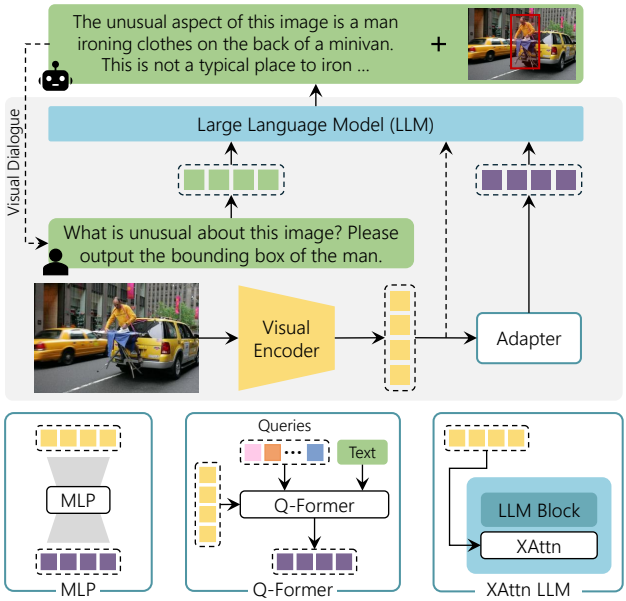

Figure 6.1: General architecture of Multimodal Large Language Models (MLLMs), composed of a visual encoder, a language model, and an adapter module that connects visual inputs to the textual space.

to that of LLMs, with Flamingo [5] being the first to explore in-context learning at scale in the vision-language field. Then, visual instruction-tuning [194] quickly became the most prominent training paradigm also in the multimodal domain, as well as the use of PEFT techniques to fine-tune the LLM. Any MLLM contains at least three components (Fig. 6.1): an LLM backbone serving as an interface with the user, one (or more) visual encoders, and one or more vision-to-language adapter modules. Popular choices for the LLM backbone often fall into the LLaMA family [315, 316]. Given that their weights are freely accessible, they have been trained on public data solely, and they boast different sizes to accommodate various use cases. In addition, their derivative versions are popular as well, such as Alpaca [305] and Vicuna [60]. The former fine-tunes LLaMA on instructions written using GPT-3, while the latter exploits user-shared conversations with ChatGPT [233]. Alternatives are OPT [411], Magneto [329], MPT [222], and the instruction-tuned [65] or multilingual [370] flavors of T5 [250], an encoder-

decoder language model pre-trained for multiple tasks.

Model Components. The main components of MLLMs are the visual encoder and the language model. The visual encoder is specifically designed to provide the LLM with the visual extracted features. It is common to employ a frozen pre-trained visual encoder while training only a learnable interface that connects visual features with the underlying LLM. The most often employed visual encoders are based on pre-trained Vision Transformer (ViT) models with a CLIP-based objective to exploit the inherent alignment of CLIP embeddings. Popular choices are the ViT-L model from CLIP [249], the ViT-H backbone from OpenCLIP [349], and the ViT-g version from EVA-CLIP [99]. As shown in [176], a stronger image encoder leads to better performance. Building on this insight, Lin *et al.* [192] and Gao *et al.* [107] propose an ensemble of frozen visual backbones to capture robust visual representations and different levels of information granularity. The utilization of such large and powerful models is made feasible by the common practice of maintaining the visual encoder frozen during training, as observed in [51, 130, 106, 176]. However, employing a frozen visual encoder has some limitations, primarily due to the constrained number of parameters, resulting in an inadequate alignment between the visual and language modalities. Specifically, the dense features extracted from the visual model may fragment the fine-grained image information and bring large computation due to the lengthy sequence when fed into the language model. To mitigate this issue, other approaches [382, 383] employ a two-stage training paradigm. In the first stage, they incorporate a trainable visual backbone while maintaining the pre-trained LLM frozen. On the other hand, LLMs primarily rely on the widely employed Transformer model, although the Mamba architecture [112] has also emerged in recent times. This proposes to make a State-Space Model (SSM) time-dependent, effectively creating a selective SSM with favorable properties: (i) inference costs and memory requirements that scale linearly with the sequence length, and (ii) efficient parallel training thanks to a smart GPU implementation of the algorithm. Similar to Transformers, Mamba models for language modeling are pre-trained using the next token prediction task. Recent studies propose MLLMs featuring Mamba as the language backbone [247, 414].

Vision-to-Language Adapters. The simultaneous presence of inputs from different modalities emphasizes the need to incorporate a module capable of delineating latent correspondences within these unimodal domains. These modules, termed as “adapters”, are intended to facilitate interoperability between the visual and textual domains. A spectrum of different adapters is used in common MLLMs,

ranging from elementary architectures such as linear layers or MLP to advanced methodologies such as Transformer-based solutions, exemplified by the Q-Former model, and conditioned cross-attention layers added to the LLM.

6.2 Visual Grounding

6.2.1 Methods

The visual grounding capabilities of an MLLM correspond to the ability to carry a dialogue with the user that includes the positioning of the content, also referred to as a referential dialogue [51]. In particular, You *et al.* [386] introduce *referring* as the ability to understand the content of an input region and can be evaluated on tasks such as region captioning and referring expression generation. Conversely, *grounding* is associated with localizing regions of a given textual description and corresponds to tasks such as referring expression comprehension (REC), referring expression segmentation (RES), phrase grounding, and grounded captioning. Two main components are required to equip MLLMs with these capabilities: a region-to-sequence method to process input regions and a sequence-to-region method to ground nouns and phrases. A summary of the MLLMs with visual grounding capabilities is reported in Table 6.1.

Region-as-Text. The most common way to output regions is to directly insert them into generated text as a series of coordinates, represented as numbers or as special tokens dedicated to location bins. Shikra [51], Kosmos-2 [241], MiniGPT-v2 [50], Ferret [386], CogVLM [333], SPHINX [192], Qwen-VL [12], and Griffon [400] convert bounding boxes into text by indicating two points. VisionLLM [335], VistaLLM [243], LLaFS [426], and ChatSpot [416] allow the MLLM to handle polygons by representing them as a series of points.

Embedding-as-Region. Another solution is to read input regions through region encoders and provide the output regions as embeddings extracted from the last layer of the MLLM to a decoder. For input regions, GLaMM [257], GPT4RoI [410], ASM [334] and ChatterBox [310] leverage features of the image encoder to perform ROI align on the bounding box, whereas PVIT [47] exploits RegionCLIP [419]. PixelLLM [365] and LLaVA-G [405] use the prompt encoder of SAM [154] and Semantic-SAM [174] respectively. For output regions, LISA [164], GLaMM, GSVA [359], NeXt-Chat [401], and LISA++ [377] send the embedding corresponding to special tokens to the mask decoder of SAM, LLaVA-G to OpenSeeD [404], Lenna [345] to Grounding-DINO [199], and PixelLM [263]

to a custom lightweight pixel decoder.

Differently, ContextDET [396] introduces a decoder that receives the latent embedding of the noun with learnable queries, performs a cross-attention with image features, and then uses a segmentation head. ChatterBox [310] combines features from the iTPN-B encoder [311] and the MLLM and provides them to the DINO detector [403]. GELLA [246] presents a fusion module in Mask2Former [58] to propose masks based on multi-modal image features and an association module to assign latent embeddings to them. PaLI-3 [54] converts embeddings into segmentation masks through a VQ-VAE [320] decoder.

Text-to-Grounding. Other approaches are based on open-vocabulary models that accept textual categories as input. DetGPT [242] generates a list of categories for Grounding-DINO. BuboGPT [417] leverages a combination of RAM, Grounding-DINO, and SAM and matches tags with nouns in the output sequence.

6.2.2 Self-Supervised Visual Encoders

Similar to the findings discussed in Chapters 3 and 4 regarding the improved localization properties of self-supervised models such as DINOv2 over vision-language models like CLIP in open-vocabulary segmentation, recent research has begun to explore how these models can serve as visual encoders within MLLMs. The advantages stem not only from their enhanced spatial awareness, but also from their flexibility in being aligned with text either via direct mapping or intermediate representations.

Jiang *et al.* [138] investigate this direction by replacing CLIP with self-supervised encoders like DINOv2 and employing a simple MLP as an adapter to the LLM. Their experiments show that DINOv2 consistently outperforms CLIP on fine-grained tasks while achieving comparable performance on image-level tasks. To capitalize on the complementary strengths of both encoders, they introduce the COMM module, which merges features from multiple intermediate layers of CLIP and DINOv2 to improve performance. GROUNDHOG [412] takes a region-centric approach, leveraging a class-agnostic region proposer to extract image regions and then pooling visual features from both DINOv2 and CLIP for each region. These region embeddings are subsequently passed to the language model, enabling fine-grained alignment and grounding. A different line of work by Tong *et al.* [313] uncovers a phenomenon they term *CLIP-Blind pairs*, instances where CLIP assigns high similarity to visually distinct images, leading to degradation in the performance of the MLLM. In contrast, DINOv2 embeddings remain more discriminative in such cases. To address this, they propose two Mixture

Model	LLM	Visual Encoder	Supporting Model	Main Tasks & Capabilities
ContextDET [396]	OPT-6.7B★	Swin-B	-	Visual Dialogue, VQA, Captioning, Detection, REC, RES
DetGPT [242]	Vicuna-13B★	EVA ViT-g	G-DINO★	Visual Dialogue, Detection
VisionLLM [335]	Alpaca-7B▲	Intern-H	Deformable-DETR▲	VQA, Captioning, Detection, Segmentation, REC
BuboGPT [417]	Vicuna-7B★	EVA ViT-g	RAM, G-DINO, SAM★	Visual Dialogue, Audio Understanding, Captioning, GroundCap
ChatSpot [416]	Vicuna-7B◆	CLIP ViT-L	-	Visual Dialogue, VQA, Captioning, Referring
GPT4RoI [410]	LLaVA-7B◆	OpenCLIP ViT-H	-	Visual Dialogue, VQA, Captioning, Referring
ASM [334]	Husky-7B▲	EVA ViT-g	-	VQA, Captioning, Referring
LISA [164]	LLaVA-13B▲	CLIP ViT-L	SAM◆	Visual Dialogue, Captioning, RES
PVIT [47]	LLaVA-7B◆	CLIP ViT-L	RegionCLIP★	Visual Dialogue, VQA, Captioning, Referring
GLaMM [257]	Vicuna-7B▲	OpenCLIP ViT-H	SAM◆	Visual Dialogue, Captioning, Referring, REC, RES, GroundCap
Griffon [400]	LLaVA-13B◆	CLIP ViT-L	-	REC, Detection, Phrase Grounding
LLaFS [426]	CodeLLaMA-7B▲	CLIP RN50	-	Few-Shot Segmentation
NExT-Chat [401]	Vicuna-7B◆	CLIP ViT-L	SAM◆	Visual Dialogue, Captioning, Referring, REC, RES, GroundCap
GSVA [359]	LLaVA-13B▲	CLIP ViT-L	SAM◆	VQA, Segmentation, REC, RES
Lenna [345]	LLaVA-7B▲	CLIP ViT-L	G-DINO◆	VQA, Captioning, REC
LISA++ [377]	LLaVA-13B▲	CLIP ViT-L	SAM◆	Visual Dialogue, Captioning, RES
LLaVA-G [405]	Vicuna-13B◆	CLIP ViT-L	OpenSeeD, S-SAM◆	Visual Dialogue, REC, RES, Grounding
PixelLLM [365]	FlanT5-XL-3B▲	EVA ViT-L	SAM★	Referring, REC, RES, GroundCap
PixelLM [263]	LLaVA-7B▲	CLIP ViT-L	-	Visual Dialogue, RES
VistaLLM [243]	Vicuna-13B◆	EVA	-	Visual Dialogue, VQA, Referring, REC, RES, GroundCap
ChatterBox [310]	LLaVA-13B▲	CLIP ViT-L	iTPN-B★, DINO◆	Visual Dialogue, Referring, REC, GroundCap
GELLA [246]	LLaVA-13B▲	CLIP ViT-L	Mask2Former◆	Segmentation, RES, GroundCap
PaLI-3 [54]	UL2-3B◆	SigLIP ViT-g	VQ-VAE◆	VQA, Captioning, Retrieval, RES

Table 6.1: Summary of MLLMs with components specifically designed for visual grounding and region-level understanding. For each model, we indicate the LLM used in its best configuration, in some cases initialized with the weights of a pre-trained MLLM, and any supporting models used to perform the task (◆: fine-tuning; ▲: fine-tuning with PEFT techniques; ★: frozen). Gray color indicates models not publicly available.

of Features (MoF) strategies: Additive-MoF, which linearly combines DINOv2 and CLIP embeddings, and Interleaved-MoF, which interleaves and separately adapts token sequences from both encoders before feeding them to the LLM. The Cambrian-1 [312] model further demonstrates the synergy between self-supervised and vision-language features, even in tasks where self-supervised models alone are less effective, such as OCR. It introduces the Spatial Vision Aggregator, a lightweight adapter using learnable queries that interact with encoder-specific features within spatial windows, enabling localized information extraction across modalities. Lastly, BRAVE [145] systematically evaluates a range of visual encoders, including DINOv2 and SILC, and proposes the MEQ-Former, a fusion module that combines multiple visual encoders with a fixed-length output and linear complexity. This allows scalable integration of complementary feature sources while preserving efficiency.

Collectively, these studies highlight a growing recognition of self-supervised encoders as valuable complements or alternatives to CLIP within MLLMs, particularly in tasks requiring precise spatial grounding.

6.2.3 Training

To enable visual grounding, MLLMs can be trained directly on task-specific data using predetermined instruction templates. For instance, CoinIt [243] is a unified set of 14 benchmarks converted into an instruction-tuning format, spanning from single-image coarse-level to multi-image region-level tasks. An additional training step is usually performed on an instruction-tuning dataset, such as LLaVa-Instruct [197], to preserve the conversational capabilities of the MLLM. However, some methods create their custom datasets to simultaneously improve the grounding and conversational capabilities. Specifically, Shikra [51], DetGPT [242], ChatSpot [416], and PVIT [47] leverage LLMs [3, 233] to combine regions and captions from datasets that present both annotations (*e.g.*, COCO). Differently, Kosmos-2 [241] and Ferret [386] exploit an open-vocabulary detector [179] to ground noun chunks parsed from captions and then reconstruct referring expressions. ASM [334], GLaMM [257], and LLaVA-G [405] propose automated pipelines comprising multiple steps based on off-the-shelf models for generating large corpora of conversations grounded in their corresponding images.

	RefCOCO			RefCOCO+			RefCOCOg	
Model	val	testA	testB	val	testA	testB	val(U)	test(U)
Kosmos-2 [241]	52.3	57.4	47.3	45.5	50.7	42.2	60.6	61.7
Shikra [51]	87.8	91.1	81.8	82.9	87.8	74.4	82.6	83.2
Qwen-VL [12]	88.6	92.3	84.5	82.8	88.6	76.8	86.0	86.3
Ferret [386]	89.5	92.4	84.4	82.8	88.1	75.2	85.8	86.3
MiniGPT-v2 [50]	88.7	91.7	85.3	80.0	85.1	74.5	84.4	84.7
CogVLM [333]	**92.8**	**94.8**	**89.0**	**88.7**	**92.9**	83.4	89.8	**90.8**
Griffon [400]	90.1	93.4	86.1	84.8	90.5	77.8	86.1	87.2
LION [48]	89.8	93.0	85.6	84.0	89.2	78.1	85.5	85.7
NExT-Chat [401]	85.5	90.0	77.9	77.2	84.5	68.0	80.1	79.8
SPHINX [192]	91.0	92.7	86.6	86.6	91.1	80.4	88.2	88.4
Lenna [345]	90.3	93.2	87.0	88.1	90.1	**84.0**	**90.3**	90.3
LLaVA-G [405]	89.2	-	-	81.7	-	-	84.8	-
Unified-IO 2 [207]	90.7	-	-	83.1	-	-	86.6	-
MM-Interleaved [309]	89.9	92.6	86.5	83.0	88.6	77.1	85.2	84.9
SPHINX-X [107]	90.6	93.7	86.9	85.5	90.5	79.9	88.3	88.5

Table 6.2: Performance analysis on the RefCOCO benchmarks for referring expression comprehension (REC). Best scores are in bold, second best are underlined.

6.2.4 Evaluation

The assessment of visual grounding capabilities of MLLMs comprises a variety of standard referring tasks, including region captioning, referring expression generation (REG), and region-level question answering, as well as grounding tasks like referring expression comprehension (REC), referring expression segmentation (RES) and grounded captioning. As regards evaluation metrics, for REC the accuracy is computed by assuming as correct predictions the ones that correspond to an intersection over union with the ground-truth above 0.5 (Acc@0.5). For referring expression segmentation the cumulative intersection over union (cIoU) is considered, while for region captioning METEOR [14] and CIDEr [324] are commonly used. However, few methods introduce their own benchmarks to evaluate the performance in more realistic scenarios, with grounded conversations that may involve multiple rounds. Quantitative results on the REC, RES, and region captioning tasks are respectively reported in Table 6.2, Table 6.3, and Table 6.4.

Below, we summarize the main datasets used across the literature, organized by the type of grounding task they support:

RefCOCO and RefCOCO+ [215] are collections of referring expressions

based on images from the COCO dataset. They were gathered through the ReferItGame [149], a two-player game where the first player examines an image featuring a segmented target object and formulates a natural language description referring to that object. The second player, who has access only to the image and the referring expression, selects the corresponding object. Players swap roles if they perform correctly, otherwise they receive a new object and image for description. The RefCOCO dataset has no constraints on the natural language and consists of 142,209 expressions for 50,000 objects across 19,994 images. Instead, in the RefCOCO+ players are disallowed from using location words in their referring expressions and it has 141,564 expressions for 49,856 objects in 19,992 images. Evaluation is performed on 1,500, 750, and 750 images corresponding to the validation, testA, and testB splits for both datasets.

RefCOCOg [391] was collected by a set of annotators who wrote natural language referring expressions for objects in COCO images, and another set of annotators who selected objects corresponding to given referring expressions. When a selected object was correct, the corresponding referring expression was inserted in the dataset. It consists of 85,474 referring expressions for 54,822 objects in 26,711 images. Evaluation is carried out on 1,300 and 2,600 images corresponding to the validation and test splits.

Visual Genome [161] connects structured image concepts to language and comprises 108,077 images along with detailed descriptions of all objects present in them, providing 5.4M region descriptions and 1.7M visual question-answer pairs. This dataset is typically used for region-level captioning and question-answering.

Visual7W [428] is a visual question-answering dataset that combines textual descriptions with image regions through object-level grounding. It comprises 328k question-answer pairs on 47k COCO images, together with 1.3M human-generated multiple-choice and more than 560k object groundings from 36,579 categories.

GRIT [241] is a large-scale dataset of grounded image-text pairs (*i.e.*, noun phrases or referring expressions associated with regions of the image) based on a subset of COYO-700M and LAION-2B. The construction pipeline consists of two steps: (i) extracting noun chunks from the captions and grounding them to bounding boxes with an open-vocabulary detector (*e.g.*, GLIP); (ii) expanding the noun chunks to referring expressions by exploiting their dependency relations in the original caption. The resulting dataset comprises 91M images, 115M text spans, and 137M associated bounding boxes.

ReasonSeg [164] is a benchmark introduced for the reasoning segmentation task, which consists of providing segmentation masks for complex and implicit query texts. Images are from OpenImages [163] and ScanNetv2 [74] and are

	RefCOCO			RefCOCO+			RefCOCOg	
Model	val	testA	testB	val	testA	testB	val(U)	test(U)
LISA [164]	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6
GLaMM [257]	**79.5**	**83.2**	**76.9**	**72.6**	**78.7**	**64.6**	74.2	74.9
NExT-Chat [401]	74.7	78.9	69.5	65.1	71.9	56.7	67.0	67.0
GSVA [359]	79.2	81.7	77.1	70.3	73.8	63.6	**75.7**	**77.0**
LLaVA-G [405]	77.1	-	-	68.8	-	-	71.5	-
PixelLLM [365]	76.9	78.5	74.4	69.2	72.1	64.5	70.7	72.4
GELLA [246]	76.7	80.5	73.6	67.0	73.2	60.6	70.4	71.5

Table 6.3: Performance analysis on the RefCOCO benchmarks for referring expression segmentation (RES). Best scores are in bold, second best are underlined.

	RefCOCO		Visual Genome	
Model	METEOR	CIDEr	METEOR	CIDEr
Kosmos-2 [241]	14.1	62.3	-	-
GPT4RoI [410]	-	-	17.4	145.2
ASM [334]	**20.8**	103.0	18.0	145.1
GLaMM [257]	16.2	**106.0**	19.7	**180.5**
NExT-Chat [401]	13.6	79.6	-	-
PixelLLM [365]	14.3	82.3	**19.9**	148.9

Table 6.4: Performance analysis on the RefCOCO and Visual Genome benchmarks for region captioning. Best scores are in bold, second best are underlined.

annotated with text instructions and corresponding segmentation masks. The resulting dataset comprises 1,218 image-instruction pairs. Evaluation metrics are the same as the RES standard benchmark. Two extended variants, ReasonDet [345] and ReasonSeg-Inst [377], are respectively introduced for reasoning detection and reasoning instance segmentation tasks.

Grounding-anything Dataset (GranD) [257] is a dataset designed for the grounded conversation generation (GCG) task, which aims to construct image-level captions with phrases associated with segmentation masks in the image. This dataset was built with an automated annotation pipeline composed of four stages: (i) object localization with the corresponding semantic label, segmentation mask, attributes, and depth information, (ii) extracting relationships between detected objects, (iii) combining previously collected relations to produce dense captions, (iv) enriching captions with contextual information. It comprises annotations for 11M SAM [154] images. Another dataset, GranD$_f$, is introduced for further fine-tuning and evaluating over the GCG task. It was gathered by extending

Flickr30k [387], RefCOCOg, and PSG [376] through GPT-4 and by manually annotating a set of samples. It comprises 214k image-grounded text pairs with 2.5k validation and 5k test samples. Evaluation metrics include METEOR and CIDEr for captioning, class-agnostic mask AP for grounding, intersection over union for segmentation, and mask recall for grounded captioning.

Grounded-Bench [405] is a benchmark introduced to assess the capabilities of an MLLM in carrying a grounded visual chat. It is built on top of the LLaVA-Bench [197], comprising conversational data generated with GPT-4 and instance annotations from COCO. It is expanded using 1,000 images with 7,000 entities from COCO annotated through an automated pipeline that involves GPT-4 to associate noun phrases from captions to ground-truth instances.

MUSE [263] is a multi-target reasoning segmentation dataset. It was created with an automated pipeline on top of 910k instance segmentation masks from the LVIS dataset [115] by exploiting GPT-4V to combine instance categories with natural language descriptions. The resulting dataset comprises 246k question-answer pairs, averaging 3.7 targets per answer.

These datasets reflect a shift toward more comprehensive benchmarks that evaluate grounding beyond static single-step tasks. By incorporating elements such as dialogue, complex reasoning, and multi-object scenarios, they better align with real-world use cases for MLLMs. Collectively, they reveal both the remarkable progress and the ongoing limitations of current models, underscoring the need for further improvements in fine-grained perception, spatial reasoning, and context-aware interaction.

Chapter 7

Conclusions

This dissertation has explored the integration of self-supervised vision models and vision–language representations for open-vocabulary semantic understanding, with a particular focus on segmentation, visual grounding, and embodied perception. The foundational insight guiding this work is that self-supervised visual backbones, trained without explicit human annotations, develop rich internal representations capable of capturing both spatial structure and semantic content. While these representations are not natively aligned with language, we demonstrate that they can be effectively connected to linguistic concepts through mechanisms such as visual prototypes and contrastive learning. This latent semantic capacity makes self-supervised features a powerful and versatile foundation for scalable open-world perception.

In bridging self-supervised learning and language grounding, we investigated not only how to align these modalities at the feature level but also how to apply this integration to practical tasks that demand fine-grained semantic understanding. These include open-vocabulary semantic segmentation, personalized instance-based navigation, and multimodal language–vision reasoning. Across these domains, we showed that self-supervised backbones, particularly DINOv2 [234], possess the necessary structure to support open-vocabulary tasks when appropriately guided by external signals or auxiliary models. This capacity enables a new class of perceptual systems that are general, adaptable, and capable of operating across domains, tasks, and modalities.

We summarize below the main contributions of this dissertation:

Open-Vocabulary Segmentation via Prototypes. We proposed a family of

methods for open-vocabulary segmentation based on visual prototypes. These methods avoid the need for direct alignment between visual and textual encoders by constructing representative visual embeddings for each category using weak supervision. We introduced VOCSeg, which builds a visual vocabulary by mining image–caption datasets and aligning segments using open-vocabulary models. We then extended this approach with FOSSIL, a training-free method that equips DINOv2 with multimodal segmentation capabilities via prototype matching, using synthetic supervision from Stable Diffusion [266]. Finally, FreeDA enhanced this strategy with fine-grained superpixels and contextualized CLIP [249] features, demonstrating that self-supervised vision backbones can achieve strong open-vocabulary segmentation without further training, while maintaining explainability properties through visual references.

Open-Vocabulary Segmentation via Contrastive Learning. We investigated whether DINOv2 can be aligned with text encoders through contrastive learning while keeping the vision backbone frozen. Motivated by the prototype-based results of the previous chapter, we proposed Talk2DINO, a framework that maps text embeddings into the DINOv2 space using a contrastive objective. Crucially, Talk2DINO introduces a novel training strategy that selects the most semantically relevant self-attention head and improves object–background separation through a custom attention-based refinement module. These results confirm that DINOv2, when exposed to even lightweight supervision, can become an effective open-vocabulary segmenter, and that vision–language alignment does not necessarily require retraining or fine-tuning of large vision models.

Personalized Instance-based Navigation. We extended the vision–language integration explored in open-vocabulary settings to embodied scenarios, where agents must perceive and act within physical simulated environments. We introduced Personalized Instance-based Navigation (PIN), a new benchmark in which a robot must navigate to a specific object instance defined by images or text descriptions, even in the presence of visually similar distractors. In this context, we demonstrated the utility of DINOv2 for instance-level matching: using its patch-level similarity structure, we showed that DINOv2 can effectively localize the same object instance across different scenes. This complements its semantic grounding capabilities and highlights its value for tasks requiring fine-grained perceptual grounding in realistic environments.

Multimodal Large Language Models for Visual Grounding. We reviewed the current landscape of Multimodal Large Language Models (MLLMs) with a focus on visual grounding and fine-grained reasoning. We surveyed the architectural

strategies, training regimes, and evaluation benchmarks that enable MLLMs to handle vision and language jointly, and in particular to reason about spatial relationships and referential expressions. We highlighted how recent MLLMs incorporate self-supervised backbones like DINOv2 to improve their localization capabilities and integrate pixel-level understanding. This integration supports the broader thesis vision of general-purpose models that combine structured perception with high-level reasoning.

7.1 Future Works and Open Challenges

While this dissertation has demonstrated the effectiveness of bridging self-supervised visual models with language for open-vocabulary perception, it also opens several avenues for future investigation. The integration of unsupervised representation learning with multimodal understanding remains a rich and evolving research area, with significant potential for expanding both theoretical insights and practical capabilities. Below, we highlight key directions for future work:

Integrating Supervised and Self-Supervised Learning for Dense Prediction. The presented methods for open-vocabulary segmentation were developed under the assumption of no manual pixel-level supervision. However, future work could explore hybrid training regimes that incorporate weak or limited supervision, such as sparse annotations or coarse labels, to enhance the segmentation quality and granularity. In particular, combining the spatial attention mechanisms of DINOv2 with lightweight supervision could guide the model to produce multi-granularity, class-agnostic segmentation masks. This would follow the paradigm of two-stage frameworks that first generate high-quality masks and subsequently refine or classify them, potentially enabling more robust and interpretable segmentation pipelines.

Scaling Insights to DINOv3 and Larger Self-Supervised Models. The promising results obtained with DINOv2 encourage further investigation into more recent and larger self-supervised backbones. DINOv3 [291], which significantly scales both the model size and training data while retaining semantic correspondence properties, represents a compelling next step. It remains an open question whether the vision–language bridging techniques proposed in this thesis, such as prototype matching and contrastive mapping, extend effectively to the feature space of DINOv3, and how its larger capacity can be leveraged for finer-grained or more abstract visual concepts.

Adapting DINOv2 for Multimodal Large Language Models. The region selection strategies introduced in Talk2DINO, which enable alignment between textual descriptions and image regions via self-attention heads, could be further developed within the MLLM framework. In particular, these mechanisms may serve as a lightweight yet effective way to augment LLMs with fine-grained perceptual grounding. Understanding how to adapt the structured feature space of DINOv2 into the token-centric processing pipelines of MLLMs could support more interpretable and spatially aware reasoning capabilities.

Harnessing Multimodal Diffusion for Concept Localization and Supervision. The recent emergence of Stable Diffusion 3 [96], built upon the Multimodal Diffusion Transformer (MM-DiT) [287], introduces new possibilities for concept generation and localization. Unlike its U-Net-based predecessors, MM-DiT may enable more flexible attention-based localization techniques that can be harnessed to identify semantic regions corresponding to generated concepts. Future work could explore how to extract and localize these concepts to build stronger, higher-resolution visual prototypes or even use them as direct supervision signals for training segmentation models. This would alleviate the need for large retrieval collections at inference time and move toward more scalable generative supervision paradigms.

Chapter 8

List of publications

The following list of publications includes all papers published during my Ph.D. period. Content and experimental results published in some of these papers have been included in the previous chapters, with explicit permission given by the other authors.

[1] Luca Barsellotti, Roberto Amoroso, Lorenzo Baraldi, and Rita Cucchiara. Enhancing Open-Vocabulary Semantic Segmentation with Prototype Retrieval. In *Proceedings of the International Conference on Image Analysis and Processing*, 2023.

[2] Luca Barsellotti, Roberto Amoroso, Lorenzo Baraldi, and Rita Cucchiara. FOSSIL: Free Open-Vocabulary Semantic Segmentation Through Synthetic References Retrieval. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2024.

[3] Luca Barsellotti, Roberto Amoroso, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Training-Free Open-Vocabulary Segmentation with Offline Diffusion-Augmented Prototype Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 89

[4] Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. The Revolution of Multimodal Large Language Models: A Survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

[5] Luca Barsellotti, Roberto Bigazzi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Personalized Instance-based Navigation Toward User-Specific Objects in Realistic Environments. *Advances in Neural Information Processing Systems*, 2024.

[6] Luca Barsellotti, Lorenzo Bianchi, Nicola Messina, Fabio Carrara, Marcella Cornia, Lorenzo Baraldi, Fabrizio Falchi, and Rita Cucchiara. Talking to DINO: Bridging Self-Supervised Vision Backbones with Language for Open-Vocabulary Segmentation. In *Proceedings of the International Conference on Computer Vision*, 2025.

Bibliography

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC Superpixels. Technical report, EPFL Technical Report, 2010. 16

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 16, 71

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023. 141, 142, 148

[4] Ziad Al-Halah, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Zero Experience Required: Plug & Play Modular Transfer Learning for Semantic Visual Navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 35

[5] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a Visual Language Model for Few-Shot Learning. In *Advances in Neural Information Processing Systems*, 2022. 143

[6] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On Evaluation of Embodied Navigation Agents. *arXiv:1807.06757*, 2018. 34, 112, 115

[7] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 34

[8] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805*, 2023. 141

[9] Nikita Araslanov and Stefan Roth. Single-Stage Semantic Segmentation from Image Labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 67, 68, 71, 90

[10] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 18, 19

[11] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 142

[12] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966*, 2023. 145, 149

[13] Sule Bai, Yong Liu, Yifei Han, Haoji Zhang, Yansong Tang, Jie Zhou, and Jiwen Lu. Self-Calibrated CLIP for Training-Free Open-Vocabulary Segmentation. *IEEE Transactions on Image Processing*, 2025. 30, 37

[14] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics Workshops*, 2005. 149

[15] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT Pre-Training of Image Transformers. In *Proceedings of the International Conference on Learning Representations*, 2022. 22

[16] Luca Barsellotti, Roberto Amoroso, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Training-Free Open-Vocabulary Segmentation with Offline Diffusion-Augmented Prototype Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 89

[17] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects. *arXiv:2006.13171*, 2020. 34, 112, 115, 131

[18] Wanda Benesova and Michal Kottman. Fast Superpixel Segmentation Using Morphological Processing. In *Proceedings of the Conference on Machine Vision and Machine Learning*, 2014. 16

[19] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. 17

[20] Lorenzo Bianchi, Fabio Carrara, Nicola Messina, Claudio Gennaro, and Fabrizio Falchi. The devil is in the fine-grained details: Evaluating open-vocabulary object detectors for fine-grained understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 136, 138

[21] Roberto Bigazzi, Lorenzo Baraldi, Shreyas Kousik, Rita Cucchiara, and Marco Pavone. Mapping High-level Semantic Regions in Indoor Environments without Object Recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2024. 113

[22] Roberto Bigazzi, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. Embodied Agents for Efficient Exploration and Smart Scene Description. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2023. 113

[23] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception Encoder: The best visual embeddings are not at the output of the network. In *Advances in Neural Information Processing Systems*, 2025. 33

[24] Eran Borenstein and Shimon Ullman. Learning to Segment. In *Proceedings of the European Conference on Computer Vision*, pages 315–328, 2004. 15

[25] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding Everything: Emerging Localization Properties in Vision-Language Transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 30, 37

[26] Yuri Boykov, Olga Veksler, and Ramin Zabih. Markov random fields with efficient approximations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998. 15

[27] Maria A Bravo, Sudhanshu Mittal, Simon Ging, and Thomas Brox. Open-Vocabulary Attribute Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 136, 138

[28] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020. 141, 142

[29] Pierre Buyssens, Isabelle Gardin, and Su Ruan. Eikonal based region growing for superpixels generation: Application to semi-supervised real time organ segmentation in CT images. *Innovation and Research in BioMedical Engineering*, 35(1):20–26, 2014. 16

[30] Pierre Buyssens, Matthieu Toutain, Abderrahim Elmoataz, and Olivier Lézoray. Eikonal-based vertices growing and iterative seeding for efficient graph-based segmentation. In *Proceedings of the IEEE International Conference on Image Processing*, 2014. 16

[31] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and Stuff Classes in Context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 64, 66, 69, 76, 86, 88, 89, 100

[32] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research. In *Proceedings of the IEEE International Conference on Advanced Robotics*, 2015. 35

[33] John F Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. 13

[34] Liangliang Cao and Li Fei-Fei. Spatially Coherent Latent Topic Model for Concurrent Segmentation and Classification of Objects and Scenes. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007. 15

[35] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment Anything with Concepts. In *Proceedings of the International Conference on Learning Representations*, 2026. 33

[36] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection With Transformers. In *Proceedings of the European Conference on Computer Vision*, 2020. 21, 33

[37] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the International Conference on Computer Vision*, 2021. 23, 27, 38, 51, 54, 56, 69, 81, 128, 130, 132, 134

[38] Niccolò Cavagnero, Gabriele Rosi, Claudia Cuttano, Francesca Pistilli, Marco Ciccone, Giuseppe Averta, and Fabio Cermelli. PEM: Prototype-based Efficient MaskFormer for Image Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 37

[39] Junbum Cha, Jonghwan Mun, and Byungseok Roh. Learning To Generate Text-Grounded Mask for Open-World Semantic Segmentation From Only Image-Text Pairs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 29, 37, 48, 53, 54, 59, 65, 66, 67, 69, 71, 79, 89, 90, 96, 99, 100

[40] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang.

Matterport3D: Learning from RGB-D Data in Indoor Environments. In *Proceedings of the International Conference on 3D Vision*, 2017. 34, 113, 114

[41] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 53

[42] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object Goal Navigation using Goal-Oriented Semantic Exploration. In *Advances in Neural Information Processing Systems*, 2020. 35, 126

[43] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural Topological SLAM for Visual Navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 35, 128

[44] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *IEEE Visual Communications and Image Processing*, 2017. 18

[45] Jeff Cheeger. A Lower Bound for the Smallest Eigenvalue of the Laplacian, 1970. 15

[46] Hila Chefer, Shir Gur, and Lior Wolf. Transformer Interpretability Beyond Attention Visualization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 128

[47] Chi Chen, Ruoyu Qin, Fuwen Luo, Xiaoyue Mi, Peng Li, Maosong Sun, and Yang Liu. Position-Enhanced Visual Instruction Tuning for Multimodal Large Language Models. *arXiv preprint arXiv:2308.13437*, 2023. 145, 147, 148

[48] Gongwei Chen, Leyang Shen, Rui Shao, Xiang Deng, and Liqiang Nie. LION: Empowering Multimodal Large Language Model with Dual-Level Visual Knowledge. *arXiv preprint arXiv:2311.11860*, 2023. 149

[49] Huafeng Chen, Pengxu Wei, Guangqian Guo, and Shan Gao. SAM-COD+: SAM-guided Unified Framework for Weakly-Supervised Camouflaged

Object Detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 33

[50] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. MiniGPT-v2: Large Language Model As a Unified Interface for Vision-Language Multi-task Learning. *arXiv preprint arXiv:2310.09478*, 2023. 145, 149

[51] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic. *arXiv preprint arXiv:2306.15195*, 2023. 144, 145, 148, 149

[52] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017. 18, 19

[53] Shizhe Chen, Thomas Chabal, Ivan Laptev, and Cordelia Schmid. Object Goal Navigation with Recursive Implicit Maps. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2023. 35, 128, 131, 132, 133, 138

[54] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. PaLI-3 Vision Language Models: Smaller, Faster, Stronger. *arXiv preprint arXiv:2310.09199*, 2023. 146, 147

[55] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*, 2015. 43, 52, 53, 65

[56] Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. DRESS: Instructing Large Vision-Language Models to Align and Interact with Humans via Natural Language Feedback. *arXiv preprint arXiv:2311.10081*, 2023. 142

[57] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision Transformer Adapter for Dense Predictions. In

Proceedings of the International Conference on Learning Representations, 2023. 20

[58] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-Attention Mask Transformer for Universal Image Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 21, 28, 31, 146

[59] Bowen Cheng, Alexander Schwing, and Alexander Kirillov. Per-Pixel Classification Is Not All You Need for Semantic Segmentation. In *Advances in Neural Information Processing Systems*, 2021. 21, 41, 43

[60] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality, 2023. 143

[61] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. CAT-Seg: Cost Aggregation for Open-Vocabulary Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 28

[62] Yunho Choi and Songhwai Oh. Image-Goal Navigation via Keypoint-Based Reinforcement Learning. In *Proceedings of the IEEE International Conference on Ubiquitous Robots*, 2021. 34, 113, 115

[63] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 142

[64] Fan RK Chung. *Spectral Graph Theory*, volume 92. American Mathematical Society, 1997. 15

[65] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling Instruction-Finetuned Language Models. *arXiv preprint arXiv:2210.11416*, 2022. 142, 143

[66] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. In *Advances in Neural Information Processing Systems*, 2012. 17

[67] Dan C Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, High Performance Convolutional Neural Networks for Image Classification. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2011. 17

[68] D Comaneci and Peter Mer Mean Shift. Arobust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:5, 2002. 16

[69] Christian Conrad, Matthias Mertz, and Rudolf Mester. Contour-Relaxed Superpixels. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2013. 16

[70] MMSegmentation Contributors. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020. 54

[71] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 64, 66, 70, 76, 88, 89, 100

[72] Claudia Cuttano, Gabriele Trivigno, Giuseppe Averta, and Carlo Masone. SANSA: Unleashing the Hidden Semantics in SAM2 for Few-Shot Segmentation. In *Advances in Neural Information Processing Systems*, 2025. 33

[73] Claudia Cuttano, Gabriele Trivigno, Gabriele Rosi, Carlo Masone, and Giuseppe Averta. SAMWISE: Infusing Wisdom in SAM2 for Text-Driven Video Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 33

[74] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 150

[75] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards Diverse and Natural Image Descriptions via a Conditional GAN. In *Proceedings of the International Conference on Computer Vision*, 2017. 16

[76] Yinpei Dai, Run Peng, Sikai Li, and Joyce Chai. Think, Act, and Ask: Open-World Interactive Personalized Robot Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2024. 35, 113, 114

[77] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers. In *Proceedings of the International Conference on Learning Representations*, 2024. 23, 81, 88, 93

[78] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. RoboTHOR: An Open Simulation-to-Real Embodied AI Platform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 34, 113, 114

[79] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-XL: A Universe of 10M+ 3D Objects. In *Advances in Neural Information Processing Systems*, 2023. 113, 118

[80] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *Advances in Neural Information Processing Systems*, 2022. 34, 113, 116

[81] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 17, 129, 130

[82] Haluk Derin, Howard Elliott, Roberto Cristi, and Donald Geman. Bayes Smoothing Algorithms for Segmentation of Binary Images Modeled by Markov Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):707–720, 1984. 15

[83] Haluk Derin, Howard Elliott, Roberto Cristi, and Donald Geman. Bayes Smoothing Algorithms for Segmentation of Images Modeled by Markov Random Fields. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 682–685, 1984. 15

[84] Karan Desai, Gaurav Kaul, Zubin Trivadi Aysola, and Justin Johnson. Redcaps: Web-curated image-text data created by the people, for the people. In *Advances in Neural Information Processing Systems*, 2021. 53

[85] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 35

[86] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, 2024. 142

[87] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, 2018. 141

[88] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 27, 37, 38, 46, 47

[89] William E Donath and Alan J Hoffman. Lower Bounds for the Partitioning of Graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973. 15

[90] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations*, 2021. 19

[91] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In

Proceedings of the International Conference on Learning Representations, 2021. 23, 127

[92] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations*, 2021. 54

[93] Fabio Drucker and John MacCormick. Fast Superpixels for Video Analysis. In *Workshop on Motion and Video Computing*, pages 1–8, 2009. 16

[94] Raphael Druon, Yusuke Yoshiyasu, Asako Kanezaki, and Alassane Watt. Visual Object Search by Learning Spatial Context. *IEEE Robotics and Automation Letters*, 2020. 35

[95] Heming Du, Xin Yu, and Liang Zheng. Learning Object Relation Graph and Tentative Policy for Visual Navigation. In *Proceedings of the European Conference on Computer Vision*, 2020. 35

[96] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *Proceedings of the International Conference on Machine Learning*, 2024. 156

[97] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, 2012. 64, 66, 69, 70, 76, 86, 88, 89, 100

[98] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA-02: A Visual Representation for Neon Genesis. *Image and Vision Computing*, 149:105171, 2024. 26, 27

[99] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA: Exploring the Limits of Masked Visual Representation Learning at Scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 26, 27, 144

[100] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning Hierarchical Features for Scene Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2012. 18

[101] Clément Farabet, Camille Couprie, Laurent Najman, and Yann Lecun. Scene Parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers. In *Proceedings of the International Conference on Machine Learning*, pages 1–8, 2012. 18

[102] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59:167–181, 2004. 16, 63, 65, 71, 73

[103] Miroslav Fiedler. A Property of Eigenvectors of Nonnegative Symmetric Matrices and its Application to Graph Theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975. 15

[104] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. DataComp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems*, 2023. 26

[105] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. CoWs on Pasture: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 127, 128, 130, 132, 133, 138

[106] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model. *arXiv preprint arXiv:2304.15010*, 2023. 144

[107] Peng Gao, Renrui Zhang, Chris Liu, Longtian Qiu, Siyuan Huang, Weifeng Lin, Shitian Zhao, Shijie Geng, Ziyi Lin, Peng Jin, Kaipeng Zhang, Wenqi Shao, Chao Xu, Conghui He, Junjun He, Hao Shao, Pan Lu, Hongsheng Li, and Yu Qiao. SPHINX-X: Scaling Data and Parameters for a Family of Multi-modal Large Language Models. *arXiv preprint arXiv:2402.05935*, 2024. 144, 149

[108] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *Proceedings of the European Conference on Computer Vision*, 2022. 28

[109] Gene H Golub and Charles F Van Loan. *Matrix Computations*. John Hopkins University Press, 2013. 14

[110] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Matthijs Douze, and Hervé Jégou. LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference. In *Proceedings of the International Conference on Computer Vision*, 2021. 21

[111] David Grangier, Léon Bottou, and Ronan Collobert. Deep Convolutional Networks for Scene Parsing. In *Proceedings of the International Conference on Machine Learning Workshops*, 2009. 17

[112] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv preprint arXiv:2312.00752*, 2023. 144

[113] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. In *Proceedings of the International Conference on Learning Representations*, 2022. 43

[114] Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A Survey on Self-Supervised Learning: Algorithms, Applications, and Future Trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9052–9071, 2024. 22

[115] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 152

[116] Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. Pay Attention to Your Neighbours: Training-Free Open-Vocabulary Semantic Segmentation. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2025. 81, 89, 90, 100

[117] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. Warp: Word-level adversarial reprogramming. *arXiv preprint arXiv:2101.00121*, 2021. 142

[118] Cong Han, Yujie Zhong, Dengjie Li, Kai Han, and Lin Ma. Open-Vocabulary Semantic Segmentation with Decoupled One-Pass Network. In *Proceedings of the International Conference on Computer Vision*, 2023. 28

[119] Xumeng Han, Longhui Wei, Xuehui Yu, Zhiyang Dou, Xin He, Kuiran Wang, Yingfei Sun, Zhenjun Han, and Qi Tian. Boosting Segment Anything Model Towards Open-Vocabulary Learning. In *Proceedings of the Conference on Artificial Intelligence*, 2025. 33

[120] Robert M Haralick and Linda G Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–132, 1985. 12, 13

[121] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for Object Segmentation and Fine-grained Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 18

[122] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 22, 56

[123] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 18

[124] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015. 19

[125] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 17, 18, 23

[126] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 142

[127] Steven L Horowitz and Theodosios Pavlidis. Picture Segmentation by a Tree Traversal Algorithm. *Journal of the ACM (JACM)*, 23(2):368–388, 1976. 12

[128] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the International Conference on Learning Representations*, 2021. 142

[129] Zhongwen Hu, Qin Zou, and Qingquan Li. Watershed Superpixel. In *Proceedings of the IEEE International Conference on Image Processing*, 2015. 16, 71

[130] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. Language Is Not All You Need: Aligning Perception with Language Models. *arXiv preprint arXiv:2302.14045*, 2023. 144

[131] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. CCNet: Criss-Cross Attention for Semantic Segmentation. In *Proceedings of the International Conference on Computer Vision*, 2019. 19

[132] Ahmad Humayun, Fuxin Li, and James M Rehg. The Middle Child Problem: Revisiting Parametric Min-Cut and Seeds for Object Proposals. In *Proceedings of the International Conference on Computer Vision*, 2015. 16

[133] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021. 43

[134] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the International Conference on Machine Learning*, 2015. 17

[135] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer To Rule Universal Image Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 21

[136] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *Proceedings of the International Conference on Machine Learning*, 2021. 26, 46, 47

[137] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of Experts. *arXiv preprint arXiv:2401.04088*, 2024. 142

[138] Dongsheng Jiang, Yuchen Liu, Songlin Liu, Jin'e Zhao, Hao Zhang, Zhen Gao, Xiaopeng Zhang, Jin Li, and Hongkai Xiong. From CLIP to DINO: Visual Encoders Shout in Multi-modal Large Language Models. *arXiv preprint arXiv:2310.08825*, 2023. 146

[139] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Trans. on Big Data*, 7(3):535–547, 2019. 54, 65

[140] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, et al. Dinov2 meets text: A unified framework for image-and pixel-level vision-language alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24905–24916, 2025. 89, 90

[141] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? *IEEE Robotics and Automation Letters*, 2020. 113

[142] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR - Modulated Detection for End-to-End Multi-Modal Understanding. In *Proceedings of the International Conference on Computer Vision*, 2021. 28

[143] Dahyun Kang and Minsu Cho. In Defense of Lazy Visual Grounding for Open-Vocabulary Semantic Segmentation. In *Proceedings of the European Conference on Computer Vision*, 2024. 89, 90

[144] Jagat Narain Kapur, Prasanna K Sahoo, and Andrew KC Wong. A New Method for Gray-Level Picture ThresholdingUusing the Entropy of the Histogram. *Computer Vision, Graphics, and Image Processing*, 29(3):273–285, 1985. 13

[145] Oğuzhan Fatih Kar, Alessio Tonioni, Petra Poklukar, Achin Kulshrestha, Amir Zamir, and Federico Tombari. BRAVE: Broadening the Visual Encoding of Vision-Language Models. In *Proceedings of the European Conference on Computer Vision*, 2025. 148

[146] Sandra Kara, Hejer Ammar, Florian Chabot, and Quoc-Cuong Pham. Image Segmentation-Based Unsupervised Multiple Objects Discovery. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2023. 24, 25

[147] Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion Models for Zero-Shot Open-Vocabulary Segmentation. In *Proceedings of the European Conference on Computer Vision*, 2024. 32, 37, 53, 54, 66, 86, 90

[148] Yasufumi Kawano and Yoshimitsu Aoki. MaskDiffusion: Exploiting Pre-Trained Diffusion Models for Semantic Segmentation. *IEEE Access*, 12:127283–127293, 2024. 32

[149] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to Objects in Photographs of Natural Scenes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014. 150

[150] Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. GOAT-Bench: A Benchmark for Multi-Modal Lifelong Navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 35, 113, 114, 115

[151] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*, 2015. 17

[152] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 18

[153] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *Proceedings of the International Conference on Computer Vision*, 2023. 133

[154] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment Anything. In *Proceedings of the International Conference on Computer Vision*, 2023. 25, 32, 145, 151

[155] Kurt Koffka. Principles of Gestalt Psychology. *Harcourt, Brace, and Company*, 1935. 12

[156] Wolfgang Köhler. *Gestalt Psychology*. Horace Liveright, 1929. 12

[157] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv:1712.05474*, 2017. 34, 113, 114, 116

[158] Philipp Krähenbühl and Vladlen Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *Advances in Neural Information Processing Systems*, 2011. 15, 54, 91

[159] Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to Objects Specified by Images. In *Proceedings of the International Conference on Computer Vision*, 2023. 114, 127, 132, 133

[160] Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-Specific Image Goal Navigation: Training Embodied Agents to Find Object Instances. *arXiv:2211.15876*, 2022. 34, 113, 115, 119

[161] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 123:32–73, 2017. 130, 150

[162] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 2012. 17

[163] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *International Journal of Computer Vision*, 128:1956–1981, 2020. 130, 150

[164] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: Reasoning Segmentation via Large Language Model. *arXiv preprint arXiv:2308.00692*, 2023. 145, 147, 150, 151

[165] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. ClearCLIP: Decomposing CLIP Representations for Dense Vision-Language Inference. In *Proceedings of the European Conference on Computer Vision*, 2024. 30, 48, 89, 90

[166] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. ProxyCLIP: Proxy Attention Improves CLIP for Open-Vocabulary Segmentation. In *Proceedings of the European Conference on Computer Vision*, 2024. 30, 48, 81, 89, 90, 91, 94, 96, 100, 106, 109

[167] Federico Landi, Roberto Bigazzi, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. Spot the Difference: A Novel Task for Embodied Agents in Changing Environments. In *Proceedings of the International Conference on Pattern Recognition*, 2022. 35

[168] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 17

[169] Minhyeok Lee, Suhwan Cho, Jungho Lee, Sunghun Yang, Heeseung Choi, Ig-Jae Kim, and Sangyoun Lee. Effective SAM Combination for Open-Vocabulary Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 33

[170] Xiaohan Lei, Min Wang, Wengang Zhou, Li Li, and Houqiang Li. Instance-aware Exploration-Verification-Exploitation for Instance ImageGoal Navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 35, 127, 132

[171] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 142

[172] Alex Levinshtein, Adrian Stere, Kiriakos N Kutulakos, David J Fleet, Sven J Dickinson, and Kaleem Siddiqi. TurboPixels: Fast Superpixels Using Geometric Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2290–2297, 2009. 16

[173] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *Proceedings of the International Conference on Learning Representations*, 2022. 27

[174] F Li, H Zhang, P Sun, X Zou, S Liu, J Yang, C Li, L Zhang, and J Gao. Semantic-SAM: Segment and Recognize Anything at Any Granularity. In *Proceedings of the European Conference on Computer Vision*, 2024. 33, 145

[175] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask DINO: Towards a Unified Transformer-Based Framework for Object Detection and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 21

[176] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *Proceedings of the International Conference on Machine Learning*, 2023. 26, 27, 144

[177] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *Proceedings of the International Conference on Machine Learning*, 2022. 26, 27

[178] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv preprint arXiv:1908.03557*, 2019. 141

[179] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded Language-Image Pre-Training. In *Proceedings of*

the IEEE Conference on Computer Vision and Pattern Recognition, 2022. 27, 28, 81, 148

[180] Weijie Li, Xinhang Song, Yubing Bai, Sixian Zhang, and Shuqiang Jiang. ION: Instance-level Object Navigation. In *Proceedings of the ACM International Conference on Multimedia*, 2021. 34, 113, 114

[181] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 142

[182] Xiangtai Li, Li Zhang, Ansheng You, Maoke Yang, Kuiyuan Yang, and Yunhai Tong. Global Aggregation then Local Distribution in Fully Convolutional Networks. In *Proceedings of the British Machine Vision Conference*, 2019. 19

[183] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *Proceedings of the European Conference on Computer Vision*, 2020. 26

[184] Yi Li, Hualiang Wang, Yiqun Duan, Jiheng Zhang, and Xiaomeng Li. A Closer Look at the Explainability of Contrastive Language-Image Pre-training. *Pattern Recognition*, 162:111409, 2025. 30, 37

[185] Zhengqin Li and Jiansheng Chen. Superpixel Segmentation using Linear Spectral Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 16

[186] Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Open-vocabulary object segmentation with diffusion models. In *Proceedings of the International Conference on Computer Vision*, 2023. 31

[187] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 27, 37, 38, 41, 43, 46, 47

[188] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 19

[189] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 19

[190] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision*, 2014. 17, 27, 65, 90, 103

[191] Yuqi Lin, Minghao Chen, Kaipeng Zhang, Hengjia Li, Mingming Li, Zheng Yang, Dongqin Lv, Binbin Lin, Haifeng Liu, and Deng Cai. TagCLIP: A Local-to-Global Framework to Enhance Open-Vocabulary Multi-Label Classification of CLIP without Training. In *Proceedings of the Conference on Artificial Intelligence*, 2024. 30, 37

[192] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. SPHINX: The Joint Mixing of Weights, Tasks, and Visual Embeddings for Multi-modal Large Language Models. *arXiv preprint arXiv:2311.07575*, 2023. 144, 145, 149

[193] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *Proceedings of the International Conference on Computer Vision*, 2023. 127

[194] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems*, 2023. 143

[195] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy Rate Superpixel Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 16

[196] Quande Liu, Youpeng Wen, Jianhua Han, Chunjing Xu, Hang Xu, and Xiaodan Liang. Open-World Semantic Segmentation via Contrasting and Clustering Vision-Language Embedding. In *Proceedings of the European Conference on Computer Vision*, 2022. 29

[197] Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. LLaVA-Plus:

Learning to Use Tools for Creating Multimodal Agents. *arXiv preprint arXiv:2311.05437*, 2023. 148, 152

[198] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with Grounded Pre-training for Open-Set Object Detection. In *Proceedings of the European Conference on Computer Vision*, 2024. 28

[199] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. In *Proceedings of the European Conference on Computer Vision*, 2024. 33, 145

[200] Wei Liu, Andrew Rabinovich, and Alexander C Berg. ParseNet: Looking Wider to See Better. In *Proceedings of the International Conference on Learning Representations Workshops*, 2016. 18

[201] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT understands, too. *AI Open*, 2023. 142

[202] Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment Anything with One Shot Using All-Purpose Feature Matching. In *Proceedings of the International Conference on Learning Representations*, 2024. 33

[203] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the International Conference on Computer Vision*, 2021. 21, 43

[204] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 18

[205] Edward Loper and Steven Bird. NLTK: The Natural Language Toolkit. *arXiv preprint cs/0205028*, 2002. 61

[206] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *Advances in Neural Information Processing Systems*, 2019. 141

[207] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-IO 2: Scaling Autoregressive Multimodal Models with Vision, Language, Audio, and Action. *arXiv preprint arXiv:2312.17172*, 2023. 149

[208] Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A Strong Baseline for Indoor Object Navigation. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2022. 35

[209] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. SegCLIP: Patch Aggregation with Learnable Centers for Open-Vocabulary Semantic Segmentation. In *Proceedings of the International Conference on Machine Learning*, 2023. 29, 69

[210] Vaïa Machairas, Etienne Decencière, and Thomas Walter. Waterpixels: Superpixels based on the watershed transformation. In *Proceedings of the IEEE International Conference on Image Processing*, 2014. 16

[211] Vaïa Machairas, Matthieu Faessel, David Cárdenas-Peña, Théodore Chabardes, Thomas Walter, and Etienne Decenciere. Waterpixels. *IEEE Transactions on Image Processing*, 24(11):3707–3716, 2015. 16

[212] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. ZSON: Zero-Shot Object-Goal Navigation using Multimodal Goal Embeddings. In *Advances in Neural Information Processing Systems*, 2022. 35, 113, 114, 128, 131, 132, 133

[213] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. THDA: Treasure Hunt Data Augmentation for Semantic Navigation. In *Proceedings of the International Conference on Computer Vision*, 2021. 35, 113, 114

[214] Yu A Malkov and Dmitry A Yashunin. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2018. 72

[215] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and Comprehension of Unambiguous

Object Descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 149

[216] Bar Mayo, Tamir Hazan, and Ayellet Tal. Visual Navigation with Spatial Attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 35

[217] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Finding an Unsupervised Image Segmenter in Each of Your Deep Generative Models. *arXiv preprint arXiv:2105.08127*, 2021. 24

[218] Rudolf Mester, Christian Conrad, and Alvaro Guevara. Multichannel Segmentation Using Contour Relaxation: Fast Super-Pixels and Temporal Propagation. In *Proceedings of the Scandinavian Conference on Image Analysis*, 2011. 16

[219] Fernand Meyer. Color Image Segmentation. In *International Conference on Image Processing and its Applications*, 1992. 16

[220] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling Open-Vocabulary Object Detection. *Advances in Neural Information Processing Systems*, 2024. 81

[221] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple Open-Vocabulary Object Detection. In *Proceedings of the European Conference on Computer Vision*, 2022. 27, 128, 130, 132, 133, 138

[222] MosaicML. Introducing MPT-7B: A New Standard for Open-Source, Commercially Usable LLMs, 2023. 143

[223] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward Semantic Segmentation With Zoom-Out Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3376–3385, 2015. 18

[224] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 64, 66, 69, 76, 88, 89, 100

[225] Arsalan Mousavian, Alexander Toshev, Marek Fišer, Jana Košecká, Ayzaan Wahid, and James Davidson. Visual Representations for Semantic Target Driven Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2019. 35

[226] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. SLIP: Self-supervision Meets Language-Image Pre-training. In *Proceedings of the European Conference on Computer Vision*, 2022. 27

[227] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim. Open Vocabulary Semantic Segmentation With Patch Aligned Contrastive Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 23, 29, 37, 48

[228] Muhammad Ferjad Naeem, Yongqin Xian, Xiaohua Zhai, Lukas Hoyer, Luc Van Gool, and Federico Tombari. SILC: Improving Vision Language Pretraining with Self-Distillation. In *Proceedings of the European Conference on Computer Vision*, 2024. 29, 89, 90

[229] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning*, 2010. 17

[230] Peer Neubert and Peter Protzel. Compact Watershed and Preemptive SLIC: On improving trade-offs of superpixel segmentation algorithms. In *Proceedings of the International Conference on Pattern Recognition*, 2014. 16

[231] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning Deconvolution Network for Semantic Segmentation. In *Proceedings of the International Conference on Computer Vision*, 2015. 18

[232] Ron Ohlander, Keith Price, and D Raj Reddy. Picture Segmentation Using a Recursive Region Splitting Method. *Computer Graphics and Image Processing*, 8(3):313–333, 1978. 12

[233] OpenAI. Introducing ChatGPT, 2022. 143, 148

[234] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa,

Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research Journal*, 2023. 23, 38, 54, 56, 58, 62, 65, 69, 81, 84, 128, 130, 132, 133, 134, 153

[235] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. 13

[236] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022. 142

[237] Anwesan Pal, Yiding Qiu, and Henrik Christensen. Learning hierarchical relationships for object-goal navigation. In *Proceedings of the Conference on Robot Learning*, 2021. 35

[238] Nikhil R Pal and Sankar K Pal. A Review on Image Segmentation Techniques. *Pattern Recognition*, 26(9):1277–1294, 1993. 12, 13

[239] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 16

[240] Theodosios Pavlidis and Steven L Horowitz. Segmentation of Plane Curves. *IEEE Transactions on Computers*, 100(8):860–870, 1974. 12

[241] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding Multimodal Large Language Models to the World. *arXiv preprint arXiv:2306.14824*, 2023. 145, 148, 149, 150, 151

[242] Renjie Pi, Jiahui Gao, Shizhe Diao, Rui Pan, Hanze Dong, Jipeng Zhang, Lewei Yao, Jianhua Han, Hang Xu, and Lingpeng Kong Tong Zhang. DetGPT: Detect What You Need via Reasoning. *arXiv preprint arXiv:2305.14167*, 2023. 146, 147, 148

[243] Shraman Pramanick, Guangxing Han, Rui Hou, Sayan Nag, Ser-Nam Lim, Nicolas Ballas, Qifan Wang, Rama Chellappa, and Amjad Almahairi. Jack

of All Tasks, Master of Many: Designing General-purpose Coarse-to-Fine Vision-Language Model. *arXiv preprint arXiv:2312.12423*, 2023. 145, 147, 148

[244] Judith MS Prewitt. Object Enhancement and Extraction. *Picture processing and Psychopictorics*, 10(1):15–19, 1970. 13

[245] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots. In *Proceedings of the International Conference on Learning Representations*, 2024. 34

[246] Lu Qi, Yi-Wen Chen, Lehan Yang, Tiancheng Shen, Xiangtai Li, Weidong Guo, Yu Xu, and Ming-Hsuan Yang. Generalizable Entity Grounding via Assistance of Large Language Model. *arXiv preprint arXiv:2402.02555*, 2024. 146, 147, 151

[247] Yanyuan Qiao, Zheng Yu, Longteng Guo, Sihan Chen, Zijia Zhao, Mingzhen Sun, Qi Wu, and Jing Liu. VL-Mamba: Exploring State Space Models for Multimodal Learning. *arXiv preprint arXiv:2403.13600*, 2024. 144

[248] Jie Qin, Jie Wu, Pengxiang Yan, Ming Li, Ren Yuxi, Xuefeng Xiao, Yitong Wang, Rui Wang, Shilei Wen, Xin Pan, et al. FreeSeg: Unified, Universal and Open-Vocabulary Image Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 28

[249] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning*, 2021. 26, 38, 43, 54, 64, 65, 67, 69, 81, 84, 98, 128, 129, 130, 132, 133, 134, 138, 141, 144, 154

[250] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 143

[251] Rajat Raina, Anand Madhavan, and Andrew Y Ng. Large-Scale Deep Unsupervised Learning Using Graphics Processors. In *Proceedings of the International Conference on Machine Learning*, 2009. 17

[252] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. PONI: Potential Functions for ObjectGoal Navigation with Interaction-free Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 35

[253] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-Matterport 3D Dataset (HM3D): 1000 Large-scale 3D Environments for Embodied AI. In *Advances in Neural Information Processing Systems*, 2021. 34, 113, 114, 120, 125

[254] Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. PIRLNav: Pretraining With Imitation and RL Finetuning for ObjectNav. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 35

[255] Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual Grouping in Contrastive Vision-Language Models. In *Proceedings of the International Conference on Computer Vision*, 2023. 29, 37

[256] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision Transformers for Dense Prediction. In *Proceedings of the International Conference on Computer Vision*, 2021. 20

[257] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Erix Xing, Ming-Hsuan Yang, and Fahad S Khan. GLaMM : Pixel Grounding Large Multimodal Model. *arXiv preprint arXiv:2311.03356*, 2023. 145, 147, 148, 151

[258] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment Anything in Images and Videos. In *Proceedings of the International Conference on Learning Representations*, 2025. 32

[259] Niyati Rawal, Roberto Bigazzi, Lorenzo Baraldi, and Rita Cucchiara. AIGeN: An Adversarial Approach for Instruction Generation in VLN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2024. 34

[260] Pengzhen Ren, Changlin Li, Hang Xu, Yi Zhu, Guangrun Wang, Jianzhuang Liu, Xiaojun Chang, and Xiaodan Liang. ViewCo: Discovering Text-Supervised Segmentation Masks via Multi-View Semantic Consistency. In *Proceedings of the International Conference on Learning Representations*, 2023. 29, 69

[261] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks. *arXiv:2401.14159*, 2024. 33, 127

[262] Xiaofeng Ren and Jitendra Malik. Learning a Classification Model for Segmentation. In *Proceedings of the International Conference on Computer Vision*, 2003. 15, 16

[263] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. PixelLM: Pixel Reasoning with Large Multimodal Model. *arXiv preprint arXiv:2312.02228*, 2023. 145, 147, 152

[264] Thomas Wilhelm Ridler, S Calvard, et al. Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(8):630–632, 1978. 13

[265] Lawrence G Roberts. Machine Perception of Three-Dimensional Solids. In *MIT Press*, 1963. 13

[266] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 25, 31, 48, 52, 60, 61, 65, 154

[267] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*, 2015. 18, 19, 31, 60

[268] Azriel Rosenfeld. *Digital Picture Processing*. Academic Press, 1976. 12

[269] Azriel Rosenfeld and Pilar De La Torre. Histogram Concavity Analysis as an Aid in Threshold Selection. *IEEE Transactions on Systems, Man, and Cybernetics*, (2):231–235, 1983. 13

[270] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 53

[271] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A Hierarchical Vision Transformer without the Bells-and-Whistles. In *Proceedings of the International Conference on Machine Learning*, 2023. 32

[272] Prasanna K Sahoo, SAKC Soltani, and Andrew KC Wong. A Survey of Thresholding Techniques. *Computer Vision, Graphics, and Image Processing*, 41(2):233–260, 1988. 13

[273] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching With Graph Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 35, 127, 129, 132

[274] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 34, 113

[275] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, 2022. 26, 43

[276] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. In *Proceedings of Neural Information Processing Systems Workshops*, 2021. 26

[277] James A Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. In *Proceedings of the National Academy of Sciences*, 1996. 128

[278] Mehmet Sezgin and Bu¨ lent Sankur. Survey Over Image Thresholding Techniques and Quantitative Performance Evaluation. *Journal of Electronic Imaging*, 13(1):146–168, 2004. 13

[279] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A Large-Scale, High-Quality Dataset for Object Detection. In *Proceedings of the International Conference on Computer Vision*, 2019. 130

[280] Tong Shao, Zhuotao Tian, Hang Zhao, and Jingyong Su. Explore the Potential of CLIP for Training-Free Open Vocabulary Semantic Segmentation. In *Proceedings of the European Conference on Computer Vision*, 2024. 30, 37

[281] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2018. 26, 53

[282] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, et al. iGibson 1.0: a Simulation Environment for Interactive Tasks in Large Realistic Scenes. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2021. 34, 113

[283] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997. 14, 24

[284] Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 14, 24, 50

[285] Yuheng Shi, Minjing Dong, and Chang Xu. Harnessing Vision Foundation Models for High-Performance, Training-Free Open Vocabulary Segmentation. In *Proceedings of the International Conference on Computer Vision*, 2025. 33

[286] Gyungin Shin, Weidi Xie, and Samuel Albanie. ReCo: Retrieve and Co-segment for Zero-shot Transfer. In *Advances in Neural Information Processing Systems*, 2022. 37, 53, 54, 66, 69, 89, 90, 96

[287] Joonghyuk Shin, Alchan Hwang, Yujin Kim, Daneul Kim, and Jaesik Park. Exploring Multimodal Diffusion Transformers for Enhanced Prompt-based Image Editing. In *Proceedings of the International Conference on Computer Vision*, 2025. 156

[288] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 34

[289] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing Objects with Self-Supervised Transformers and no Labels. In *Proceedings of the British Machine Vision Conference*, 2021. 24, 81, 84

[290] Oriane Siméoni, Chloé Sekkat, Gilles Puy, Antonín Vobeckỳ, Éloi Zablocki, and Patrick Pérez. Unsupervised Object Localization: Observing the Background To Discover Objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 81, 84

[291] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. 23, 155

[292] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of the International Conference on Learning Representations*, 2015. 17, 18

[293] Irwin Sobel and Gary Feldman. A 3x3 Isotropic Gradient Operator for Image Processing. *Stanford Artificial Intelligence Project*, 1968. 13

[294] Robert J Sternberg, Karin Sternberg, and Jeff Mio. *Cognitive Psychology*. Wadsworth Publishing, 2009. 12

[295] Johann Strassburg, Rene Grzeszick, Leonard Rothacker, and Gernot A Fink. On the Influence of Superpixel Methods for Image Parsing. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, volume 2, pages 518–527, 2015. 17

[296] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for Semantic Segmentation. In *Proceedings of the International Conference on Computer Vision*, 2021. 21

[297] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An Evaluation of the State-of-the-Art. *Computer Vision and Image Understanding*, 166:1–27, 2018. 16

[298] Lin Sun, Jiale Cao, Jin Xie, Xiaoheng Jiang, and Yanwei Pang. CLIPer: Hierarchically Improving Spatial Representation of CLIP for Open-Vocabulary Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 32

[299] Lin Sun, Jiale Cao, Jin Xie, Fahad Shahbaz Khan, and Yanwei Pang. iSeg: An Iterative Refinement-based Framework for Training-free Segmentation. *arXiv preprint arXiv:2409.03209*, 2024. 32

[300] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: Improved Training Techniques for CLIP at Scale. *arXiv preprint arXiv:2303.15389*, 2023. 26

[301] Shuyang Sun, Runjia Li, Philip Torr, Xiuye Gu, and Siyang Li. CLIP as RNN: Segment Countless Visual Concepts without Training Endeavor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 33

[302] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 17, 18

[303] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training Home Assistants to Rearrange their Habitat. In *Advances in Neural Information Processing Systems*, 2021. 34

[304] Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the DAAM: Interpreting Stable Diffusion Using Cross Attention. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2022. 31, 49, 54, 60

[305] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford Alpaca: An Instruction-Following LLaMA Model, 2023. 143

[306] H Emrah Tasli, Cevahir Cigla, and A Aydin Alatan. Convexity constrained efficient superpixel and supervoxel extraction. *Signal Processing: Image Communication*, 33:71–85, 2015. 16

[307] H Emrah Tasli, Cevahir Cigla, Theo Gevers, and A Aydin Alatan. Super Pixel Extraction via Convexity Induced Boundary Adaptation. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2013. 16

[308] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. Unifying Language Learning Paradigms. *arXiv preprint arXiv:2205.05131*, 2022. 142

[309] Changyao Tian, Xizhou Zhu, Yuwen Xiong, Weiyun Wang, Zhe Chen, Wenhai Wang, Yuntao Chen, Lewei Lu, Tong Lu, Jie Zhou, et al. MM-Interleaved: Interleaved Image-Text Generative Modeling via Multi-modal Feature Synchronizer. *arXiv preprint arXiv:2401.10208*, 2024. 149

[310] Yunjie Tian, Tianren Ma, Lingxi Xie, Jihao Qiu, Xi Tang, Yuan Zhang, Jianbin Jiao, Qi Tian, and Qixiang Ye. ChatterBox: Multi-round Multimodal Referring and Grounding. *arXiv preprint arXiv:2401.13307*, 2024. 145, 146, 147

[311] Yunjie Tian, Lingxi Xie, Zhaozhi Wang, Longhui Wei, Xiaopeng Zhang, Jianbin Jiao, Yaowei Wang, Qi Tian, and Qixiang Ye. Integrally Pre-Trained Transformer Pyramid Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 146

[312] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer,

Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A Fully Open, Vision-Centric Exploration of Multimodal LLMs. In *Advances in Neural Information Processing Systems*, 2024. 148

[313] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes Wide Shut? Exploring the Visual Shortcomings of Multimodal LLMs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 146

[314] Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: Revenge of the ViT. In *Proceedings of the European Conference on Computer Vision*, 2022. 69

[315] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*, 2023. 143

[316] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 143

[317] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*, 2025. 27

[318] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. In *Advances in Neural Information Processing Systems*, 2020. 127

[319] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin De Capitani, and Luc Van Gool. SEEDS: Superpixels Extracted via Energy-Driven Sampling. In *Proceedings of the European Conference on Computer Vision*, 2012. 16, 71

[320] Aaron Van Den Oord, Oriol Vinyals, et al. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*, 2017. 146

[321] Wouter Van Gansbeke, Simon Vandenhende, and Luc Van Gool. Discovering Object Masks with Transformers for Unsupervised Semantic Segmentation. *arxiv preprint arXiv:2206.06363*, 2022. 24

[322] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, 2017. 19, 26, 129, 141

[323] Andrea Vedaldi and Stefano Soatto. Quick Shift and Kernel Methods for Mode Seeking. In *Proceedings of the European Conference on Computer Vision*, 2008. 16

[324] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-Based Image Description Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 149

[325] Olga Veksler, Yuri Boykov, and Paria Mehrani. Superpixels and Supervoxels in an Energy Optimization Framework. In *Proceedings of the European Conference on Computer Vision*, 2010. 16

[326] Luc Vincent and Pierre Soille. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(06):583–598, 1991. 16

[327] Vibashan VS, Shubhankar Borse, Hyojin Park, Debasmit Das, Vishal Patel, Munawar Hayat, and Fatih Porikli. PosSAM: Panoptic Open-vocabulary Segment Anything. *arXiv preprint arXiv:2403.09620*, 2024. 33

[328] Feng Wang, Jieru Mei, and Alan Yuille. SCLIP: Rethinking Self-Attention for Dense Vision-Language Inference. In *Proceedings of the European Conference on Computer Vision*, 2024. 30, 37, 48, 81, 89, 90, 99, 100

[329] Hongyu Wang, Shuming Ma, Shaohan Huang, Li Dong, Wenhui Wang, Zhiliang Peng, Yu Wu, Payal Bajaj, Saksham Singhal, Alon Benhaim, et al. Magneto: A Foundation Transformer. In *Proceedings of the International Conference on Machine Learning*, 2023. 143

[330] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-End Panoptic Segmentation With Mask Transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 21

[331] Jie Wang and Xiaoqiang Wang. VCells: Simple and Efficient Superpixels Using Edge-Weighted Centroidal Voronoi Tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1241–1247, 2012. 16

[332] Jinglong Wang, Xiawei Li, Jing Zhang, Qingyuan Xu, Qin Zhou, Qian Yu, Lu Sheng, and Dong Xu. Diffusion Model is Secretly a Training-Free Open Vocabulary Semantic Segmenter. *IEEE Transactions on Image Processing*, 2025. 32

[333] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. CogVLM: Visual Expert for Pretrained Language Models. *arXiv preprint arXiv:2311.03079*, 2023. 145, 149

[334] Weiyun Wang, Min Shi, Qingyun Li, Wenhai Wang, Zhenhang Huang, Linjie Xing, Zhe Chen, Hao Li, Xizhou Zhu, Zhiguo Cao, et al. The All-Seeing Project: Towards Panoptic Visual Recognition and Understanding of the Open World. *arXiv preprint arXiv:2308.01907*, 2023. 145, 147, 148, 151

[335] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. VisionLLM: Large Language Model is also an Open-Ended Decoder for Vision-Centric Tasks. *arXiv preprint arXiv:2305.11175*, 2023. 145, 147

[336] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. InternImage: Exploring Large-Scale Vision Foundation Models With Deformable Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 127

[337] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a Foreign Language: BEiT Pretraining for Vision and Vision-Language Tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 27

[338] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-Local Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 19

[339] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and Learn for Unsupervised Object Detection and Instance Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 24, 48, 51, 54, 81

[340] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L Crowley, and Dominique Vaufreydaz. Self-Supervised Transformers for Unsupervised Object Discovery Using Normalized Cut. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 81

[341] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. TokenCut: Segmenting Objects in Images and Videos With Self-Supervised Transformer and Normalized Cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15790–15801, 2023. 24, 81, 84

[342] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022. 142

[343] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. *arXiv preprint arXiv:2204.07705*, 2022. 142

[344] Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. Multi-ON: Benchmarking Semantic Map Memory using Multi-Object Navigation. In *Advances in Neural Information Processing Systems*, 2020. 34, 112, 113, 114

[345] Fei Wei, Xinyu Zhang, Ailing Zhang, Bo Zhang, and Xiangxiang Chu. Lenna: Language enhanced reasoning detection assistant. *arXiv preprint arXiv:2312.02433*, 2023. 145, 147, 149, 151

[346] David Weikersdorfer, Alexander Schick, and Daniel Cremers. Depth-Adaptive Supervoxels for RGB-D Video Segmentation. In *Proceedings of the IEEE International Conference on Image Processing*, 2013. 16

[347] Max Wertheimer. Experimentelle Studien über das Sehen von Bewegung. *Zeitschrift für Psychologie*, 61:161–265, 1912. 12

[348] Max Wertheimer. Laws of Organization in Perceptual Forms. *A source book of Gestalt psychology*, pages 71–88, 1938. 12

[349] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust Fine-Tuning of Zero-Shot Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 144

[350] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CvT: Introducing Convolutions to Vision Transformers. In *Proceedings of the International Conference on Computer Vision*, 2021. 21

[351] Ji-Jia Wu, Andy Chia-Hao Chang, Chieh-Yu Chuang, Chun-Pei Chen, Yu-Lun Liu, Min-Hung Chen, Hou-Ning Hu, Yung-Yu Chuang, and Yen-Yu Lin. Image-Text Co-Decomposition for Text-Supervised Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 29, 37

[352] Junfeng Wu, Yi Jiang, Qihao Liu, Zehuan Yuan, Xiang Bai, and Song Bai. General Object Foundation Model for Images and Videos at Scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 28, 37

[353] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Change Loy. CLIPSelf: Vision Transformer Distills Itself for Open-Vocabulary Dense Prediction. *arXiv preprint arXiv:2310.01403*, 2023. 29, 30

[354] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. DiffuMask: Synthesizing Images with Pixel-level Annotations for Semantic Segmentation Using Diffusion Models. *arXiv preprint arXiv:2303.11681*, 2023. 31

[355] Zhenyu Wu and Richard Leahy. An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993. 14, 24

[356] Monika Wysoczańska, Michaël Ramamonjisoa, Tomasz Trzciński, and Oriane Siméoni. CLIP-DIY: CLIP Dense Inference Yields Open-Vocabulary Semantic Segmentation For-Free. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2024. 86, 89, 90

[357] Monika Wysoczanska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzcinski, and Patrick Pérez. CLIP-DINOiser: Teaching CLIP a few DINO tricks for open-vocabulary semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, volume 3, 2024. 29, 86, 89, 90, 91, 94, 106, 109

[358] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-World Perception for Embodied Agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 34, 113, 114

[359] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. GSVA: Generalized Segmentation via Multimodal Large Language Models. *arXiv preprint arXiv:2312.10103*, 2023. 145, 147, 151

[360] Shiting Xiao, Rishabh Kabra, Yuhang Li, Donghyun Lee, Joao Carreira, and Priyadarshini Panda. OpenWorldSAM: Extending SAM2 for Universal Image Segmentation with Language Prompts. In *Advances in Neural Information Processing Systems*, 2025. 33

[361] Enze Xie, Wenhai Wang, Zhiding Yu, Animashree Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation With Transformers. In *Advances in Neural Information Processing Systems*, 2021. 21

[362] Yun Xing, Jian Kang, Aoran Xiao, Jiahao Nie, Ling Shao, and Shijian Lu. Rewrite Caption Semantics: Bridging Semantic Gaps for Language-Supervised Semantic Segmentation. In *Advances in Neural Information Processing Systems*, 2023. 29

[363] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. GroupViT: Semantic Segmentation Emerges From Text Supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 29, 46, 47, 53, 54, 66, 68, 69, 89, 90, 96, 99

[364] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 31

[365] Jiarui Xu, Xingyi Zhou, Shen Yan, Xiuye Gu, Anurag Arnab, Chen Sun, Xiaolong Wang, and Cordelia Schmid. Pixel Aligned Language Models. *arXiv preprint arXiv:2312.09237*, 2023. 145, 147, 151

[366] Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Yi Wang, Yu Qiao, and Weidi Xie. Learning Open-Vocabulary Semantic Segmentation Models From Natural Language Supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 29, 69

[367] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side Adapter Network for Open-Vocabulary Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 28

[368] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A Simple Baseline for Open-Vocabulary Semantic Segmentation with Pre-trained Vision-Language Model. In *Proceedings of the European Conference on Computer Vision*, 2022. 27, 28, 37, 38, 41, 46, 47

[369] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-Scale Conv-Attentional Image Transformers. In *Proceedings of the International Conference on Computer Vision*, 2021. 21

[370] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020. 143

[371] Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. OVRL-V2: A simple state-of-art baseline for ImageNav and ObjectNav. *arXiv:2303.07798*, 2023. 35

[372] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline Visual Representation Learning for Embodied Navigation. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023. 35

[373] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-Matterport 3D Semantics Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 113, 120

[374] Brian Yamauchi. A Frontier-Based Approach for Autonomous Exploration. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997. 127

[375] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal Instance Perception As Object Discovery and Retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 28

[376] Jingkang Yang, Yi Zhe Ang, Zujin Guo, Kaiyang Zhou, Wayne Zhang, and Ziwei Liu. Panoptic scene graph generation. In *Proceedings of the European Conference on Computer Vision*, 2022. 152

[377] Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. LISA++: An Improved Baseline for Reasoning Segmentation with Large Language Model. *arXiv preprint arXiv:2312.17240*, 2023. 145, 147, 151

[378] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual Semantic Navigation using Scene Priors. In *Proceedings of the International Conference on Learning Representations*, 2019. 35

[379] Yuhang Yang, Jinhong Deng, Wen Li, and Lixin Duan. ResCLIP: Residual Attention for Training-free Dense Vision-language Inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 30, 37, 48

[380] Jian Yao, Marko Boben, Sanja Fidler, and Raquel Urtasun. Real-Time Coarse-to-Fine Topologically Preserving Segmentation. In *Proceedings of*

the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 16

[381] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary Tasks and Exploration Enable ObjectNav. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 35

[382] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mPLUG-Owl: Modularization Empowers Large Language Models with Multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 144

[383] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mPLUG-Owl2: Revolutionizing Multi-modal Large Language Model with Modality Collaboration. *arXiv preprint arXiv:2311.04257*, 2023. 144

[384] Muyang Yi, Quan Cui, Hao Wu, Cheng Yang, Osamu Yoshie, and Hongtao Lu. A Simple Framework for Text-Supervised Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 29, 37

[385] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 2014. 17

[386] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and Ground Anything Anywhere at Any Granularity. *arXiv preprint arXiv:2310.07704*, 2023. 145, 148, 149

[387] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 152

[388] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *Proceedings of the International Conference on Learning Representations*, 2016. 18

[389] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations*, 2016. 19

[390] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive Captioners are Image-Text Foundation Models. *Transactions on Machine Learning Research*, 2022. 27

[391] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling Context in Referring Expressions. In *Proceedings of the European Conference on Computer Vision*, 2016. 150

[392] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions Die Hard: Open-Vocabulary Segmentation with Single Frozen Convolutional CLIP. In *Advances in Neural Information Processing Systems*, 2023. 28, 37

[393] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A New Foundation Model for Computer Vision. *arXiv preprint arXiv:2111.11432*, 2021. 26, 27

[394] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-Contextual Representations for Semantic Segmentation. In *Proceedings of the European Conference on Computer Vision*, 2020. 19

[395] Charles T Zahn. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, 20(1):68–86, 1971. 12, 14, 24

[396] Yuhang Zang, Wei Li, Jun Han, Kaiyang Zhou, and Chen Change Loy. Contextual Object Detection with Multimodal Large Language Models. *arXiv preprint arXiv:2305.18279*, 2023. 146, 147

[397] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *Proceedings of the European Conference on Computer Vision*, 2014. 17

[398] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *Proceedings of the International Conference on Computer Vision*, 2023. 26, 27, 29

[399] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. LiT: Zero-Shot Transfer With Locked-Image Text Tuning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 27, 130

[400] Yufei Zhan, Yousong Zhu, Zhiyang Chen, Fan Yang, Ming Tang, and Jinqiao Wang. Griffon: Spelling out All Object Locations at Any Granularity with Large Language Models. *arXiv preprint arXiv:2311.14552*, 2023. 145, 147, 149

[401] Ao Zhang, Liming Zhao, Chen-Wei Xie, Yun Zheng, Wei Ji, and Tat-Seng Chua. NExT-Chat: An LMM for Chat, Detection and Segmentation. *arXiv preprint arXiv:2311.04498*, 2023. 145, 147, 149, 151

[402] Fei Zhang, Tianfei Zhou, Boyang Li, Hao He, Chaofan Ma, Tianjiao Zhang, Jiangchao Yao, Ya Zhang, and Yanfeng Wang. Uncovering Prototypical Knowledge for Weakly Open-Vocabulary Semantic Segmentation. In *Advances in Neural Information Processing Systems*, 2023. 29

[403] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. *arXiv preprint arXiv:2203.03605*, 2022. 146

[404] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A Simple Framework for Open-Vocabulary Segmentation and Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 28, 145

[405] Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Lei Zhang, Chunyuan Li, et al. LLaVA-Grounding: Grounded Visual Chat with Large Multimodal Models. *arXiv preprint arXiv:2312.02949*, 2023. 145, 147, 148, 149, 151, 152

[406] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A Tale of Two Features: Stable Diffusion Complements DINO for Zero-Shot Semantic Correspondence. In *Advances in Neural Information Processing Systems*, 2023. 63

[407] L Zhang, X Li, A Arnab, K Yang, Y Tong, and P Torr. Dual Graph Convolutional Network for Semantic Segmentation. In *Proceedings of the British Machine Vision Conference*, 2019. 19

[408] Li Zhang, Dan Xu, Anurag Arnab, and Philip HS Torr. Dynamic Graph Message Passing Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 19

[409] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Hao Dong, Peng Gao, and Hongsheng Li. Personalize Segment Anything Model with One Shot. In *Proceedings of the International Conference on Learning Representations*, 2024. 33, 128, 130, 132

[410] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. GPT4RoI: Instruction Tuning Large Language Model on Region-of-Interest. *arXiv preprint arXiv:2307.03601*, 2023. 145, 147, 151

[411] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 143

[412] Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. GROUNDHOG: Grounding Large Language Models to Holistic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 146

[413] Yuhang Zhang, Richard Hartley, John Mashford, and Stewart Burn. Superpixels via Pseudo-Boolean Optimization. In *Proceedings of the International Conference on Computer Vision*, 2011. 16

[414] Han Zhao, Min Zhang, Wei Zhao, Pengxiang Ding, Siteng Huang, and Donglin Wang. Cobra: Extending Mamba to Multi-Modal Large Language Model for Efficient Inference. *arXiv preprint arXiv:2403.14520*, 2024. 144

[415] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 18, 19

[416] Liang Zhao, En Yu, Zheng Ge, Jinrong Yang, Haoran Wei, Hongyu Zhou, Jianjian Sun, Yuang Peng, Runpei Dong, Chunrui Han, et al. ChatSpot: Bootstrapping Multimodal LLMs via Precise Referring Instruction Tuning. *arXiv preprint arXiv:2307.09474*, 2023. 145, 147, 148

[417] Yang Zhao, Zhijie Lin, Daquan Zhou, Zilong Huang, Jiashi Feng, and Bingyi Kang. BuboGPT: Enabling Visual Grounding in Multi-Modal LLMs. *arXiv preprint arXiv:2307.08581*, 2023. 146, 147

[418] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabuo Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking Semantic Segmentation From a Sequence-to-Sequence Perspective With Transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 20

[419] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. RegionCLIP: Region-based Language-Image Pretraining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 27, 81, 145

[420] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing Through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 64, 66, 70, 73, 76, 88, 89, 100

[421] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic Understanding of Scenes Through the ADE20K Dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 64, 66, 70, 73, 76, 88, 89, 100

[422] Chong Zhou, Chen Change Loy, and Bo Dai. Extract Free Dense Labels from CLIP. In *Proceedings of the European Conference on Computer Vision*, 2022. 30, 37, 48, 53, 54, 66, 69, 81, 89, 90, 96

[423] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT Pre-Training with Online Tokenizer. *Proceedings of the International Conference on Learning Representations*, 2022. 23

[424] Nan Zhou, Ke Zou, Kai Ren, Mengting Luo, Linchao He, Meng Wang, Yidi Chen, Yi Zhang, Hu Chen, and Huazhu Fu. MedSAM-U: Uncertainty-Guided Auto Multi-Prompt Adaptation for Reliable MedSAM. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 33

[425] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting Twenty-Thousand Classes Using Image-Level Supervision. In *Proceedings of the European Conference on Computer Vision*, 2022. 81

[426] Lanyun Zhu, Tianrun Chen, Deyi Ji, Jieping Ye, and Jun Liu. LLaFS: When Large-Language Models Meet Few-Shot Segmentation. *arXiv preprint arXiv:2311.16926*, 2023. 145, 147

[427] Minzhao Zhu, Binglei Zhao, and Tao Kong. Navigating to Objects in Unseen Environments by Distance Prediction. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2022. 35

[428] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7W: Grounded Question Answering in Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 150

[429] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2017. 34, 113

[430] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized Decoding for Pixel, Image, and Language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 28

[431] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment Everything Everywhere All at Once. In *Advances in Neural Information Processing Systems*, 2023. 28