University of Modena and Reggio Emilia
XXXVII cycle of the International Doctorate School in
Information and Communication Technologies
Doctor of Philosophy dissertation in
Computer Engineering and Science

Multimodal Attentive Deep Learning Architectures for Visual-Semantic Understanding

A Multimodal Bridge from Pixels to Reasoning

Roberto Amoroso

Supervisor: Prof. Rita Cucchiara
Co-Supervisor: Prof. Lorenzo Baraldi
PhD Course Coordinator: Prof. Luigi Rovati

Modena, 2025

Review committee composed of:

Prof. Volker Tresp, *Ludwig-Maximilians-Universität München (LMU)*
Dr. Fabrizio Falchi, *National Research Council (ISTI-CNR)*

Contents

Chapter 1

Introduction

Computer Vision has experienced rapid advancements in recent years, driven by the emergence of attentive and Transformer-based models. These architectures have revolutionized the field, enabling complex data interactions and pushing the boundaries of Artificial Intelligence (AI). Central to this evolution is attention modeling, which has become a cornerstone of research in vision, facilitating a sophisticated and nuanced understanding of diverse data types, such as text, images, and videos. The integration of these data types has given rise to Multimodal Deep Learning, which aims to emulate human-like perception and reasoning across multiple modalities. This fusion of modalities enhances model performance and expands the applicability of AI systems across various domains, from autonomous vehicles to healthcare diagnostics.

The research presented in this thesis investigates critical challenges associated with multimodal attentive architectures, including the integration of Vision Transformers, multimodal learning, and visual-semantic understanding to enhance the ability of AI systems to interpret and reason about visual content.

Organization and Summary. Vision Transformers (ViTs) have emerged as a groundbreaking architecture that challenges the long-standing dominance of Convolutional Neural Networks (CNNs) in Computer Vision. ViTs have demonstrated remarkable capabilities in modeling long-range dependencies and capturing global relationships in visual data by adopting the self-attention mechanism originally designed for Natural Language Processing (NLP). However, the success of ViTs comes with significant computational costs and demands for large-scale training data, presenting challenges that need to be addressed for practical applications. In

Chapter 3, we address these issues by introducing bidimensional downsampling techniques that optimize ViTs for enhanced hierarchical feature extraction while mitigating computational inefficiencies. Another crucial aspect of developing effective ViT models lies in their pre-training strategies. Self-supervised learning has emerged as a powerful paradigm for harnessing unlabeled data to learn meaningful visual representations. Chapter 4 introduces novel pre-training objectives and tokenization strategies that enhance the efficiency and effectiveness of self-supervised learning for vision tasks.

Despite the success of attention-based architectures, classical Computer Vision techniques continue to offer valuable insights and complementary strengths. Chapter 5 explores the integration of traditional perception-based approaches, such as superpixels, with modern Transformer-based segmentation architectures. This fusion demonstrates how established Computer Vision principles can enhance boundary awareness and spatial reasoning capabilities in attention-based models, particularly in dense prediction tasks like semantic segmentation.

The potential of Transformers extends beyond pure visual tasks into the realm of multimodal learning. By bridging vision and language, Transformers enable sophisticated understanding across modalities, from open-vocabulary segmentation to deepfake detection and video question answering. This thesis investigates various aspects of multimodal integration. In Chapter 6, we leverage vision-language models to perform open vocabulary semantic segmentation, a cutting-edge multimodal task that enables the segmentation of arbitrary categories expressed in textual form. We introduce innovative approaches such as synthetic references generation and prototype retrieval to bridge the gap between global features and pixel-level semantics. These methods effectively address the domain shift problem and enable open-vocabulary segmentation capabilities without relying on extensive training or large annotated datasets.

Throughout our investigation, we recognize the importance of comprehensive evaluation methodologies for segmentation models. Traditional metrics, while valuable, often fail to capture the nuanced aspects of model performance across different applications. Chapter 7 proposes a novel fine-grained error analysis method that provides deeper insights into model behavior and facilitates more targeted improvements in segmentation algorithms.

The broader implications of multimodal learning are further explored through a systematic study on multimodal deepfake detection, presented in Chapter 8. By leveraging contrastive-based disentangling strategies, we analyze the interplay between textual semantics and low-level visual cues in the context of advanced diffusion models. In the realm of multimodal video understanding, Chapter 9 presents

a text-guided temporal querying Transformer for video question answering. This component effectively bridges frame-wise visual perception with the reasoning capabilities of Large Language Models (LLMs), advancing the state-of-the-art in multimodal video comprehension.

The research presented in this thesis spans a wide spectrum of Computer Vision challenges, from low-level semantic segmentation to high-level deepfake detection and video reasoning. By developing novel methodologies and architectures, we contribute to expanding the possibilities of artificial visual intelligence. Our findings have implications for various applications, including augmented reality, personalized robotics, medical imaging, and autonomous driving, paving the way for future research in visual-semantic understanding.

Sintesi in Lingua Italiana

Negli ultimi anni, la Visione Artificiale ha compiuto progressi straordinari grazie all'introduzione di modelli basati su Transformer. Queste architetture hanno rivoluzionato il settore, consentendo interazioni complesse tra dati e ampliando i confini dell'Intelligenza Artificiale (IA). Il fulcro di questa trasformazione è la modellazione dell'attenzione, che consente una comprensione sofisticata di diversi tipi di dati, come testi, immagini e video. L'integrazione di questi dati ha dato vita al Deep Learning Multimodale, che mira a replicare la percezione e il ragionamento umano su più modalità, migliorando le prestazioni e ampliando le applicazioni dell'IA in settori come la sanità e la guida autonoma.

La ricerca presentata in questa tesi indaga sfide critiche associate alle architetture multimodali attentive, inclusa l'integrazione dei Vision Transformers, l'apprendimento multimodale e la comprensione visivo-semantica per migliorare la capacità dei sistemi IA di interpretare e ragionare sui contenuti visivi.

I Vision Transformers (ViTs) sono emersi come un'architettura innovativa che sfida il dominio di lunga data delle Reti Neurali Convoluzionali (CNN) nella Visione Artificiale. I ViT hanno dimostrato notevoli capacità nel modellare dipendenze a lungo raggio e catturare relazioni globali nei dati visivi adottando il meccanismo di auto-attenzione originariamente progettato per l'Elaborazione del Linguaggio Naturale (NLP). Tuttavia, il successo dei ViT comporta costi computazionali significativi e richiede grandi quantità di dati per l'addestramento, presentando sfide che devono essere affrontate per applicazioni pratiche. Nel Capitolo 3, affrontiamo queste problematiche introducendo tecniche di campionamento bidimensionale che ottimizzano i ViT per un'estrazione gerarchica delle features visuali, riducendo al contempo le inefficienze computazionali.

Un altro aspetto cruciale dello sviluppo di modelli ViT efficaci risiede nelle loro strategie di pre-addestramento. L'apprendimento auto-supervisionato è emerso come un potente paradigma per sfruttare i dati non annotati al fine di apprendere migliori rappresentazioni visive. Il Capitolo 4 introduce nuovi obiettivi di pre-addestramento e strategie di tokenizzazione che migliorano l'efficienza e l'efficacia dell'apprendimento auto-supervisionato per task visivi.

Nonostante il successo delle architetture basate sull'attenzione, le tecniche classiche della Visione Artificiale continuano a offrire intuizioni preziose e punti di forza complementari. Il Capitolo 5 esplora l'integrazione degli approcci tradizionali basati sulla percezione, come i superpixel, con le moderne architetture di segmentazione basate su Transformer. Questa fusione dimostra come i principi consolidati della Visione Artificiale possano migliorare la consapevolezza dei confini e le capacità di ragionamento spaziale nei modelli basati sull'attenzione, particolarmente nei compiti di previsione densa come la segmentazione semantica.

Il potenziale dei Transformers si estende oltre i compiti puramente visivi nel regno dell'apprendimento multimodale. Collegando visione e linguaggio, i Transformers consentono una comprensione sofisticata tra modalità diverse, dalla segmentazione open-vocabulary al rilevamento dei deepfake e alla risposta a domande sui video. Questa tesi indaga vari aspetti dell'integrazione multimodale. Nel Capitolo 6, sfruttiamo modelli vision-language per eseguire segmentazioni semantiche open-vocabulary, un compito multimodale all'avanguardia che consente la segmentazione di categorie arbitrarie espresse in forma testuale. Introduciamo approcci innovativi come la generazione di riferimenti sintetici e il recupero di prototipi per colmare il divario tra caratteristiche globali e semantica a livello pixel. Questi metodi affrontano efficacemente il problema dello spostamento del dominio e consentono capacità di segmentazione open-vocabulary senza fare affidamento su ampi set di dati annotati o addestramenti estensivi.

Nel corso di questa indagine, riconosciamo l'importanza delle metodologie di valutazione complete per i modelli di segmentazione. Le metriche tradizionali, spesso non riescono a cogliere gli aspetti sfumati delle prestazioni del modello in diverse applicazioni. Il Capitolo 7 propone un nuovo metodo di analisi degli errori fine-grained che fornisce informazioni più approfondite sul comportamento del modello e facilita miglioramenti più mirati negli algoritmi di segmentazione.

Le implicazioni più ampie del deep learning multimodale sono ulteriormente esplorate attraverso uno studio sistematico sul rilevamento multimodale dei deepfake, presentato nel Capitolo 8. Sfruttando strategie contrastive basate sul disaccoppiamento, analizziamo l'interazione tra semantica testuale e indizi visivi a basso livello nel contesto dei modelli avanzati di diffusione.

Nel dominio della comprensione video multimodale, il Capitolo 9 presenta un Transformer temporale guidato dal testo per la risposta a domande sui video. Questo componente collega efficacemente la percezione visiva fotogramma per fotogramma con le capacità di ragionamento dei grandi modelli linguistici, avanzando lo stato dell'arte nella comprensione video multimodale.

La ricerca presentata in questa tesi copre un ampio spettro di sfide della Visione Artificiale, dalla segmentazione semantica a basso livello al rilevamento avanzato dei deepfake e al ragionamento video. Sviluppando metodologie e architetture innovative, contribuiamo ad ampliare le possibilità dell'intelligenza visiva artificiale. I nostri risultati hanno implicazioni per varie applicazioni, tra cui realtà aumentata, robotica personalizzata, imaging medico e guida autonoma, aprendo la strada a future ricerche nella comprensione visivo-semantica.

1.1 Activities Carried Out During the Ph.D.

Besides the research activities described in this thesis, I took part in several conferences and journals throughout the three years of my PhD, both as a reviewer and presenter. A list of the main additional activities is reported below, while the complete list of my publications is reported in Appendix A.

Internships

- NVIDIA, Munich, Germany: Conducted research on multimodal video understanding for autonomous vehicles, under the co-supervision of Prof. Laura Leal-Taixé and Dr. Elmar Haussmann. Developed a large-scale text-video retrieval system, leading to an oral presentation at NTECH 2024.
- ELLIS PhD Internship at LMU, Munich, Germany: Research activity focused on multimodal video understanding with large language models and open-vocabulary segmentation, under the co-supervision of Prof. Volker Tresp. The internship resulted in two research papers presented at WACV 2024 and WACV 2025.

Participations to National and International Projects

- ELLIS PhD Student: My research activity in the European Laboratory for Learning and Intelligent Systems (ELLIS) focuses on multimodal attentive models for visual semantic understanding.

- European Lighthouse on Secure and Safe AI (ELSA) Project: Developed a deepfake detection method for images generated through diffusion models, including the creation of the COCOFake dataset.
- ROAd Digital Sustainable Twins in Emilia-Romagna (Roadster) Project: Developed a multimodal open-vocabulary segmentation network for urban environments.
- Multidisciplinary AI Italian Skills (MAIIS) Project: Created an information platform for data collection and reporting on AI technology development in Italy.

Collaborations with Companies

- Worked with Memooria s.r.l. on creating a print-resistant adversarial watermarking system for artwork protection.
- Collaborated with Maticad s.r.l. to develop advanced semantic segmentation models for indoor environments.
- Interaction with CINECA to access high-performance computing resources.

Conferences and Journals Reviewing

- Computer Vision and Pattern Recognition (CVPR)
- International Conference on Pattern Recognition (ICCV)
- European Conference on Computer Vision Workshops (ECCV)
- IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)
- Association for the Advancement of Artificial Intelligence (AAAI)
- IEEE Transactions on Multimedia (TMM)
- Pattern Recognition Letters (PRL)
- ACM Multimedia (ACMMM)
- International Conference on Pattern Recognition (ICPR)

Conferences Attended

- WACV 2025, Winter Conference on Applications of Computer Vision, Tucson, USA.
- ECCV 2024, European Conference on Computer Vision, Milan, Italy.
- CVPR 2024, Conference on Computer Vision and Pattern Recognition, Seattle, USA.
- WACV 2024, Winter Conference on Applications of Computer Vision, Hawaii, USA.
- BMVC 2023, British Machine Vision Conference, Aberdeen, UK.
- ICIAP 2023, International Conference on Image Analysis and Processing, Udine, Italy.
- CCNC 2023, Consumer Communications and Networking Conference, Las Vegas, USA.
- ICIAP 2021, International Conference on Image Analysis and Processing, Lecce, Italy.
- CAIP 2021, International Conference on Computer Analysis of Images and Patterns, Virtual Conference.
- IWANN 2021, International Work-Conference on Artificial Neural Networks, Virtual Conference.
- CoNEXT 2020, International Conference on Emerging Networking EXperiments and Technologies, Barcelona, Spain.

Schools and Symposia Attended

- ELLIS Summer School on Large-Scale AI 2023, Modena, Italy.
- International Computer Vision Summer School (ICVSS) 2022, Sicily, Italy.
- ELLIS Doctoral Symposium 2023, Helsinki, Finland.
- ELLIS Doctoral Symposium 2022, Alicante, Spain.
- ELLIS Doctoral Symposium 2021, Tubingen, Germany.

Master Thesis Co-advising

- Brandon Willy Viglianisi, "Revisiting token mixing in Transformer architectures", 2022.
- Paolo Bruno, "Investigating Bidimensional Downsampling in Vision Transformer Models", 2021.

Tutoring and Teaching

- Computer Vision and Cognitive Systems Master Course.
- AI for Automotive Master Course.
- School in AI: Deep Learning, Vision and Language for Industry.
- CINECA AI Academy: AI, Computer vision, and HPC.
- Python Programming, Machine Learning, and Data Analysis Course.

Awards and Honors

- Outstanding Reviewer Award at ECCV 2024.
- Best Project Award at the ELLIS Summer School on Large-Scale AI 2023.
- Reading Group Competition Award at ICVSS 2022.
- Best Paper Award at ICIAP 2021.
- Best Poster Award at CoNEXT 2020.

Seminars Attended

- "From Machine Learning to Autonomous Intelligence", Prof. Yann LeCun, 2023.
- "Thoughts on Artificial Intelligence", Prof. Jürgen Schmidhuber, 2023.
- "Graph-based Tracking", Prof. Laura Leal-Taixé, 2023.
- "From Images to Text: New Forms of Human-AI Interaction", Prof. Lorenzo Baraldi, 2023.

- "From Handcrafted to End-to-End Learning, and Back: a Journey far Multi-Object Tracking", Prof. Laura Leal-Taixé, 2022.
- "Human behavior understanding in large-scale visual data", Prof. Rita Cucchiara, 2022.
- "Applied Deep Learning in the Industry", Dr. Alessandro Nicolosi, 2022.
- "Webinar Series on Transformer Architectures", NVIDIA AI Technology Center, 2022.
- "Artificial Intelligence on HPC", EuroCC Italy, 2022.
- "Research in videogames: use of deep learning for saliency estimation and cheating prevention", Dr. Iuri Frosio, 2021.

Courses attended

- "Retrieval Optimization: Tokenization to Vector Quantization", Dr. Kacper Łukawski, 2024.
- "Multimodal RAG: Chat with Videos", Dr. Vasudev Lal, 2024.
- "Finetuning Large Language Models", Dr. Sharon Zhou, 2023.
- "Multimodal Machine Learning", Prof. Cigdem Beyan and Prof. Wei Wang, 2022.
- "High-Performance Deep Learning with GPUs", Dr. Giuseppe Fiameni, 2022.

Chapter 2

Literature Survey

This chapter provides an overview of the literature from which our research has been inspired. We first review the most important works related to Vision Transformer (ViT) architectures and focus on their inherent challenges, such as computational efficiency and the need for effective pre-training strategies. In Section 2.1, we explore key advancements aimed at addressing these limitations, including innovations in self-attention mechanisms, hybrid architectures, and self-supervised pre-training techniques tailored for ViTs.

The discussion then shifts to the evolution of semantic segmentation, in Section 2.2, highlighting the transition from traditional convolutional methods to modern attention-based architectures. Additionally, we revisit classical Computer Vision techniques like superpixels and discuss their integration into modern pipelines to enhance segmentation precision. We also examine how multimodal approaches have enabled breakthroughs such as open-vocabulary segmentation. The limitations of existing segmentation evaluation metrics are also addressed, alongside the introduction of fine-grained error analysis methodologies for more nuanced performance assessment.

Next, we explore multimodal deepfake detection, an increasingly critical research area given the rise of synthetic media and its associated risks of disinformation and malicious misuse. Section 2.3 reviews deepfake detection methodologies, spanning traditional approaches for GAN-generated content to recent innovations targeting diffusion-based deepfakes. The role of datasets in driving progress is emphasized, with a focus on the need for diverse and challenging benchmarks that better reflect real-world scenarios.

Finally, Section 2.4 traces the evolution of video Question Answering (video QA) from early sequential neural network-based approaches to modern Transformer-based encoder-decoder architectures. We examine methods for efficient temporal modeling, which are crucial for handling the dense and dynamic nature of video data, and investigate how pre-trained Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) are leveraged to address the unique challenges inherent in video QA tasks.

2.1 Vision Transformer Architectures

The emergence of ViTs represents a paradigm shift in Computer Vision, introducing a novel architecture that challenges traditional convolutional approaches. Originally designed for Natural Language Processing (NLP), Transformer models have demonstrated remarkable capabilities in visual tasks by leveraging self-attention mechanisms to capture long-range dependencies and global relationships in visual data. However, the adoption of Vision Transformers in practical applications faces two primary challenges: computational efficiency and the need for effective pre-training strategies.

In the following sections, we explore the key developments and innovations in Vision Transformer architectures, with a particular focus on approaches that improve their efficiency and enhanced pre-training strategies. Specifically, we first delve into methods aimed at reducing the computational demands of ViTs while maintaining their strong performance. These include innovations in self-attention mechanisms, sequence length reduction, and hybrid approaches that combine the strengths of Transformers and CNNs. Subsequently, we examine the growing body of work on self-supervised learning techniques tailored for ViTs. These methods leverage unlabeled data to pre-train models effectively, with an emphasis on the development of visual targets as supervisory signals. These approaches not only make Vision Transformers more practical for real-world applications but also provide insights into the fundamental trade-offs between model capacity, computational efficiency, pre-training strategies, and performance.

2.1.1 Enhancing Efficiency in Attentive Models

Despite having been initially proposed for machine translation and Natural Language Processing, Transformer-based architectures [60, 192, 230] have recently demonstrated their effectiveness also in many Computer Vision tasks [122, 215],

either combining self-attention with convolutional layers or using pure attention-based solutions. Approaches based on these architectures reached state-of-the-art results in several tasks, including image classification [67, 224], object detection [19, 310], semantic segmentation [107, 299], image and video captioning [49, 50, 287], and image generation [114].

Although many improvements of convolutional neural networks for image classification are inspired by Transformers (*e.g*. squeeze and excitation [104], selective kernel [143], and split-attention networks [284]), the first work that transfers a pure Transformer-based architecture to the classification task has been proposed by Dosovitskiy *et al*. [67] with the introduction of the Vision Transformer model. This architecture takes as input a sequence of square image patches and directly applies Transformer layers over raw pixels. While it has achieved promising results on ImageNet [205] bridging the gap with state-of-the-art convolutional neural networks, a pre-training stage on a large amount of data is required to achieve these remarkable performances. To solve this issue and to manage the high computational requirements typical of Transformer-based models, many different solutions have been proposed including the use of low-bit quantization [289], network pruning [177, 220], and knowledge distillation [224].

Other solutions, more specific to Transformer models, tackle the quadratic complexity of the self-attention operator. For example, Child *et al*. [44] proposed to factorize the self-attention matrix into smaller and faster attention operators, thus obtaining a $\mathcal{O}(n\sqrt{n})$ complexity. Further, linear complexity can be obtained via a kernel-based formulation of self-attention, as proposed in [121], or by performing the self-attention operator on non-overlapping local windows, as proposed in [162].

On a different line, some works go in the direction of limiting the length of the input sequence to process [86, 186, 237, 278]. For example, the approach proposed by Yuan *et al*. [278] consists in structuring the image into tokens for capturing local patterns. Other strategies consist of downsampling the sequence length, either merging 2D patches, as done in [237], or applying 1D pooling to the intermediate tokens, as done in [186]. In Chapter 3, we follow this direction and propose to apply bidimensional downsampling to visual tokens at the intermediate blocks of the Vision Transformer architecture, which is closer to what is commonly done with 2D max-pooling layers in convolutional neural networks.

2.1.2 Self-Supervised Pre-training of Vision Transformers

Self-supervised learning is a machine learning paradigm where models are trained using unlabeled data by generating supervisory signals from the data itself. Unlike

supervised learning, which relies on manually annotated labels, and unsupervised learning, which aims to discover inherent structures or patterns in data without any explicit supervision, self-supervised learning creates pseudo-labels or pretext tasks that allow models to learn meaningful representations of the data. This approach bridges the gap between supervised and unsupervised learning by leveraging inherent structures or relationships within the input data to formulate training objectives. For instance, in Computer Vision, self-supervised tasks might involve predicting missing parts of an image, reconstructing corrupted inputs, or identifying transformations applied to the data. This approach has gained significant traction in recent years due to its ability to leverage large-scale unlabeled datasets and produce representations that generalize well across diverse downstream tasks.

Driven by its effectiveness in learning generalizable representations, numerous innovative methods have emerged in recent years to effectively pre-train vision-based architectures using self-supervised strategies. Early approaches primarily focused on designing diverse pretext tasks [64, 83, 183, 240, 288] whereas later methods increasingly leveraged contrastive learning paradigms [32, 36, 89, 97, 184]. The advent of Vision Transformer models [67] has pushed towards the introduction of new increasingly sophisticated self-supervised pre-training strategies [5, 20, 31, 37]. In this context, motivated by the great success in NLP, some attempts have been made to effectively adapt the Masked Language Modeling paradigm [60, 160] and its auto-regressive variant [12] in the Computer Vision domain, either by directly predicting pixels [31], image patches [67], or discrete visual tokens [5]. In particular, the recently proposed BEiT approach [5] effectively performs pre-training via image patch masking and predicts the discretized labels [194] of masked patches. Since the introduction of BEiT, many subsequent methods based on similar pre-training strategies have been presented [34, 66, 70, 96, 244, 245, 256]. Some of these methods [34, 70, 96] have introduced an encoder-decoder architecture to separate uncorrupted encoded information from masked tokens, which are employed directly as input to the decoder. Inspired by findings from NLP literature [214, 269], in Chpater 4 we propose a method to overcome the limitations of Masked Image Modeling in self-supervised pre-training.

Visual Targets. In the domain of self-supervised learning, visual targets are a set of specific objectives that are used as the supervisory signal to pre-train visual models. Based on the type of signal, the targets can be categorized into low-level visual features, hand-crafted features, and visual tokens. Some recent studies [73, 96, 106, 161, 256] have utilized pixel information as the low-level supervisory signal for self-supervised pre-training. In contrast, [244] have used

HOG hand-crafted features to reconstruct the masked visual input. More recently, visual targets based on CLIP [191] have been used with remarkable success, either by directly employing CLIP features [103, 245, 290] or training a discrete tokenizer [189] to reconstruct the semantic features encoded by CLIP. In Chapter 4, we propose a discrete visual tokenizer based on CLIP features which offers a novel approach to visual pre-training without the need for specific training over a particular dataset.

2.2 Multimodal Semantic Segmentation

Semantic segmentation is a fundamental task in Computer Vision that involves assigning a semantic label to every pixel in an image, enabling detailed scene understanding and object delineation. Formally, for a given input image the goal is to predict a label map where each pixel corresponds to a semantic class from a predefined set of categories. In recent years, the field has witnessed remarkable progress, driven by advancements in both traditional Convolutional Neural Networks and emerging attention-based architectures like Vision Transformers. However, even these models struggle with tasks requiring fine-grained understanding or adaptation to unseen categories.

The concept of multimodal semantic segmentation builds upon these limitations by integrating additional modalities to enhance segmentation performance. By leveraging the complementary nature of visual and textual data, multimodal approaches enable models to generalize beyond predefined class sets and perform open-vocabulary segmentation. For example, vision-language models like CLIP [191] utilize textual descriptions to guide pixel-level predictions, allowing for fine-grained region understanding based on semantic cues. This paradigm shift not only addresses challenges in recognizing novel objects but also improves contextual understanding in complex scenes.

This section begins with an overview of traditional convolutional methods, tracing the evolution of deep learning techniques that have shaped semantic segmentation. We then explore the application of Transformer-based models to dense tasks, emphasizing their advantages over convolutional backbones. Furthermore, we revisit classical Computer Vision techniques like superpixels and discuss their integration into modern segmentation pipelines to enhance boundary precision and spatial reasoning. The discussion extends to multimodal approaches to segmentation, emphasizing the ability of open-vocabulary methods to leverage synergies between visual and textual modalities for enhanced performance. Finally, we

address the limitations of existing evaluation metrics and motivate the need for fine-grained error analysis methodologies to better assess model performance across diverse scenarios.

2.2.1 Convolutional Semantic Segmentation Models

Semantic segmentation is a cornerstone visual task, counting numerous applications, such as object detection, scene understanding, and image editing. Over the years, several deep learning-based approaches have been proposed for semantic image segmentation. One of the most successful approaches is the fully convolutional network (FCN) [163], which enabled outstanding progress in semantic segmentation, performing pixel-to-pixel classification in an end-to-end manner. Many recent efforts have been focused on improving different aspects of FCN. Several works [27, 28, 29] enlarged the receptive field by adopting dilated convolutions, while others introduced context modeling [95, 152, 187, 272, 282], boundary information refinement [9, 26, 273] and multi-scale feature aggregation [233, 295] to obtain fine-grained predictions. Recently, attention-based models [79, 107, 142, 239, 296] have been employed to learn long-range context information. While these approaches adopt convolutional encoding backbones [99] for feature extraction, more recent methods have proved the effectiveness of employing Transformer-based [230] backbones for semantic segmentation.

2.2.2 Attentive Models for Dense Prediction

Transformer-based architectures for dense prediction tasks, specifically semantic segmentation, have initially been tackled by adding convolutional or MLP decoders on top of ViT-based backbones. One key advantage of ViTs over CNNs is their ability to process high-resolution images without downscaling, allowing the retention of more detailed information about the objects in an image and leading to improved segmentation accuracy. Ranftl *et al*. [195] propose to reshape tokens from different layers of a ViT-like backbone into an image representation with decreasing resolution and increasing channels.

Other approaches focus on reducing memory and model parameters, by downsampling the encoder sequence length in subsequent layers, as seen in Pyramid Vision Transformers [237], or by employing lightweight MLP decoders like SegFormer [255]. Some methods compute self-attention locally within groups of patches [45, 162], while others, such as XCiT [71], transpose query-key interactions to achieve linear complexity with respect to the number of patches.

More recently, Zhang *et al*. [280] propose a plain Transformer encoder-decoder architecture that employs a novel attention-to-mask mechanism that generates segmentation masks by evaluating the similarity between learnable class tokens and multi-level ViT feature maps.

2.2.3 Superpixels for Semantic Segmentation

Observing that pixels are not natural and meaningful entities to represent images, the first reference to superpixels as a preprocessing step can be found in [199]. Several alternatives for superpixel extraction followed, which can be classified based on their high-level approach, according to the categorization reported in [1] and [217]. Besides watershed-based algorithms [7, 171, 180], methods based on clustering techniques [1, 149, 232, 246], such as k-means, use pixel color, and spatial information and allow to specify the desired number of superpixels and their compactness. Other strategies instead treat images like a graph and partition their edges based on color similarities [75, 87, 157, 199]. Deep learning-based approaches have recently been proposed for superpixel computation [111, 267], and superpixels have been exploited for preserving edges and improving semantic segmentation [77, 264]. More recently, [292] applies a superpixel algorithm to the input image to extract semantically homogeneous regions and segment the image by per-region prediction using a sequence-to-sequence Transformer. For a comprehensive description and evaluation of state-of-the-art superpixel algorithms, we refer the reader to [217]. In Chapter 5, we exploit superpixels' position, shape, and edge priors to develop positional encoding techniques that boost the performance of ViT-based semantic segmentation models.

2.2.4 Open-Vocabulary Semantic Segmentation

Building upon the success of large-scale vision-language models in zero-shot classification [112, 191], several works on open-vocabulary segmentation have investigated strategies to transfer the multimodal image-text alignment toward finer granularity (*i.e*., region or pixel level) [62, 82, 151, 258, 261].

Supervised Methods. A group of literature has been focusing on the supervision provided by dense annotations, available for a limited set of categories, to generalize on unseen classes. OpenSeg [82] decouples the task in a region proposer and a grounder that aligns regions to words from captions. Similarly, OVSeg [151] employs a two-stage method, in which class-agnostic regions are masked and provided to a CLIP encoder with learnable visual prompts. SAN [261] combines

a side network with CLIP to propose regions while recognizing their corresponding semantic category. However, these approaches are affected by performance gaps between seen and unseen categories [62, 151] and, due to the costs of dense annotations, can be applied in limited domains.

Unsupervised Methods. Other works have instead exploited contrastive training over a large set of image-text pairs, without dense annotations. GroupViT [258] proposes a Transformer architecture that learns to group image regions progressively. MaskCLIP [304] adapts a frozen CLIP for dense predictions through modifications in the last attention layer. TCL [21] presents a grounding mechanism that learns to associate text to regions during contrastive learning. OVSegmentor [260] introduces a module based on slot attention to group tokens of a Transformer and aligns them to captions.

Diffusion Models for Segmentation. Diffusion models [201] have proven state-of-the-art performance in image generation. Few works tackle the task of localizing the concepts mentioned in the conditioning captions during the generation. DAAM [219] proposes exploiting the cross-attention mechanism that Stable Diffusion uses to extract attribution maps for the words mentioned in the prompt. DiffuMask [251] leverages the advances of DAAM to generate ground truth segmentation masks without human annotation and train a segmentation model on them. GroundedDiffusion [150] implements a grounding module to align textual and visual embeddings during the diffusion process. Some works have investigated the usage of diffusion models for open-vocabulary segmentation. ODISE [259] employs Stable Diffusion as a feature extractor for its mask generator. OVDiff [117] generates a set of visual references at prediction time to support the semantic segmentation process.

In Chapter 6, we present a novel unsupervised open-vocabulary segmentation approach that leverages image generation during an offline stage to collect visual prototypes, effectively minimizing computational overhead at prediction time. To enhance efficiency further, it employs superpixels to segment the image into class-agnostic regions, enabling the efficient computation of local visual similarities.

2.2.5 Segmentation Error Analysis

The primary metric for evaluating semantic segmentation models is the mean Intersection over Union ($mIoU$), though other metrics like the *F1-measure/Dice score* and *Pixel Accuracy*, are occasionally considered in addition to $mIoU$ [39]. To better align with human perception, the Boundary F1 score (BF score) [55] was

introduced, offering a per-image evaluation of segmentation quality. Additionally, *Trimap IoU* [27, 128] focuses on the IoU within a narrow band around ground-truth mask boundaries, making it less sensitive to errors far from these boundaries and favoring larger predictions. Another boundary-focused metric, *Boundary IoU* [39], computes the intersection over union only on pixels that lie close to the boundaries of predicted or ground-truth foreground masks. Restricting the set of pixels for IoU computation to the boundary pixels has the advantage of mitigating IoU's bias toward large objects. However, at the same time, it introduces an insensitivity toward errors far away from predicted and ground-truth mask boundaries. For a detailed comparison of these metrics, refer to [39].

While these metrics provide an overall assessment of model performance, they offer limited insight into specific error types. To shed some light on the errors of a model, one can consider *Precision* and *Recall*, providing rather limited information as errors are only distinguished in terms of false positives and false negatives. Alternatively, one can resort to a qualitative visual inspection. However, a qualitative analysis can be time-consuming and subject to a relatively high variance due to the limited number of samples that can be examined.

Therefore, a more detailed and quantitative method for error analysis would significantly benefit researchers and practitioners by enabling fine-grained model evaluation. For the task of object detection, tools like TIDE [10] have proven effective by categorizing detection errors and quantifying their impact on *Average Precision (AP)*, guiding several effective model design choices [30, 38, 145, 218, 297]. To address this gap in semantic segmentation, a fine-grained quantitative evaluation framework is introduced in Chapter 7. This framework enables detailed error analysis across diverse scenarios, offering researchers the tools to better understand and improve segmentation models.

2.3 Multimodal DeepFake Detection

The rapid advancements in generative models, particularly diffusion-based architectures, have led to the proliferation of synthetic media generated from textual prompts in natural language, commonly referred to as deepfakes. While these technologies have enabled numerous creative and practical applications, they also pose significant challenges in terms of misuse, including the spread of disinformation, identity theft, and other malicious activities. As a result, the detection of deepfakes has become a critical area of research in Computer Vision and multimodal learning.

Unlike unimodal approaches, which rely solely on visual cues, multimodal deepfake detection leverages complementary information from multiple modalities, such as visual and textual signals, to significantly enhance detection accuracy and robustness. This capability makes multimodal methods particularly effective for identifying sophisticated deepfakes characterized by subtle semantic inconsistencies between textual prompts and their corresponding generated images.

The following sections provide a comprehensive exploration of deepfake detection methodologies. First, we review both traditional and modern approaches for identifying deepfakes generated by GANs and other visual forgery tools, with a particular emphasis on methods that enhance generalization capabilities and leverage multimodal integration. Next, we examine recent advancements in detecting deepfakes produced by diffusion models, which represent the cutting edge of generative technology. Finally, we analyze the role that dataset characteristics play in driving progress in this domain.

2.3.1 General Deepfake Detection

In recent years, with the growth and diffusion of generative models, several research efforts [54, 231] have been made to effectively detect synthetic images generated by GANs [85, 119, 120, 178, 309] and other deep learning-based architectures [126, 227]. While initial works did not concentrate on the generalization capabilities of deepfake detectors [175, 204], subsequent approaches [23, 52, 84, 88, 174, 236] focused instead on the development of generic detectors that can be applied to different generators, thus avoiding the need to have a specific detector for each generative model. On the same line, different solutions [68, 78, 291] proposed to detect deepfakes based on the spectrum of GAN-generated images. In fact, CNN-based generative models usually leave a distinguishable fingerprint over generated images, due to transposed convolutions [68, 291], up-sampling operations [78, 24], and the spectral bias of convolution layers [69, 123]. Some works in similar directions also focused on associating fake images to the corresponding generator among several known GANs [116, 274] or extending deepfake detection to the video domain [53, 91, 92, 93, 268]. In the latter case, deepfakes are usually generated by partially manipulating original videos with existing tools for face swapping and other sophisticated algorithms for audio manipulation. Research efforts in this domain have mainly been dedicated to improving deepfake detection performance with the integration of multiple modalities, such as spatial rich model filters [92, 169] and audio traces [42, 268] in both cases combined with RGB features.

2.3.2 Diffusion-based Deepfakes Detection

While all aforementioned methods are tailored for detecting deepfakes generated by GANs or other visual forgery tools, a few works extended the analysis to deepfake images coming from diffusion models [61, 181, 193, 201, 206]. Among them, [247] proposed to detect fake images based on their wavelet-packet representations taking into account features from the pixel and frequency space. [200] evaluate the performance of state-of-the-art detectors and also tackle the frequency domain, analyzing different factors that influence the spectral properties of these images, discovering that GANs and diffusion models produce images with different characteristics that require adaptation of existing classifiers to ensure reliable detection. Similarly, [51] introduced an analysis of the forensics traces left by common diffusion models and investigated whether deepfake detectors tailored for GANs can also distinguish images generated by diffusion models. Finally, [211] analyzed and compared deepfakes generated by different text-to-image diffusion models, investigating the possibility of correctly attributing deepfake images to the diffusion model that generated them. Overall, these studies highlight the need for developing detection methods that can effectively detect deepfakes generated by various types of generative models, including diffusion models. In Chapter 8, we investigate deepfake detection for images generated by advanced diffusion models, analyzing the effectiveness of contrastive and classification-based visual features. Additionally, we propose a multimodal framework to assess detection strategies, introducing a novel contrastive disentangling method to investigate the interplay between textual semantics and perceptual cues.

2.3.3 Datasets for Deepfake Detection

The availability of large datasets has played a crucial role in the development of deepfake detection techniques. One of the most widely used datasets is FaceForensics++ [204], which contains videos of real and fake faces generated using several generative models. The dataset provides both raw and manipulated videos with different compression rates and resolutions, allowing the evaluation of deepfake detection methods under different scenarios. Another popular dataset is Celeb-DF [148], which contains videos of celebrities manipulated using different techniques including GANs and face swapping. Celeb-DF also provides several levels of difficulty, ranging from low-quality to high-quality forgeries, making it suitable for evaluating both traditional and advanced deepfake detection methods. Other datasets have been proposed, such as DeeperForensics-1.0 [113], which

contains manipulated videos generated using multiple GAN-based models, and DFDC [65], composed of thousands of videos of real and fake faces.

Despite the availability of these datasets, there is still a need for more diverse and challenging datasets that reflect the increasing sophistication of deepfake generation methods. In particular, while current datasets mainly focus on faces, there is a lack of datasets for detecting deepfakes in other types of images, such as natural scenes. In Chapter 8, we introduced a novel large-scale dataset designed to address this limitation. This dataset includes natural images paired with their corresponding synthetic counterparts generated using diffusion models, as well as natural language captions that establish semantic links between them. This allows for the evaluation of deepfake detection methods in a more complex and diverse context and also enables the development of methods that can identify semantic inconsistencies between natural and synthetic images.

2.4 Multimodal Video Question Answering

The rapid advancements in multimodal learning and the integration of vision and language have significantly expanded the scope of video understanding tasks. Among these, video Question Answering has emerged as a fundamental and challenging problem, requiring models to reason over both the spatial and temporal dimensions of video content while aligning this understanding with textual queries. Video QA is inherently complex due to the dynamic, dense, and multi-event nature of videos. Unlike static images, videos require models to capture temporal dependencies, contextual relationships, and causal reasoning across sequences of frames. The introduction of Transformer-based architectures, pre-trained LLMs, and MLLMs has revolutionized this field by enabling more effective integration of visual and textual modalities.

In the following sections, we delve into key research areas in video QA. First, we trace the evolution of video QA from early approaches based on sequential neural networks to modern Transformer-based encoder-decoder architectures, highlighting the critical role of benchmark datasets in driving advancements. Next, we examine methods for efficient temporal feature extraction, including sparse sampling techniques and adaptive temporal selection strategies, which are crucial for handling the vast temporal information in videos. Finally, we investigate how LLMs and MLLMs are leveraged to address the unique challenges of video QA, focusing on their ability to reason across temporal sequences while effectively aligning visual features with textual queries.

2.4.1 Video QA in the Pre-LLM Era

Video Question Answering is a core task in video-language understanding that requires assigning meaningful answers to questions based on video content [300]. Early approaches to video language modeling primarily relied on sequential neural networks, such as Long Short-Term Memory networks [132], 3D Convolutional Neural Networks [144], and Graph Neural Networks [188, 253]. These models were effective in capturing temporal and spatial information but faced limitations in handling complex video data and contextual reasoning. With the advent of Transformer-based visual encoders and novel paradigms for video-language pre-training [4, 14, 134, 191, 234, 241, 265, 266], encoder-decoder models have achieved significant advancements in the video QA task. These models benefit from the Transformers' ability to capture long-range dependencies and contextual information, leading to improved performance in understanding and reasoning about video content. The development of video QA has been driven by diverse benchmark datasets spanning multiple domains, including movies and TV shows [136, 137, 223, 257], as well as wild, online videos [90, 249, 252, 257, 277]. Notably, datasets like NeXT-QA [252] and STAR [249] have introduced more challenging aspects of video understanding, emphasizing temporal reasoning and causal relationships.

2.4.2 Temporal Modeling for Video QA

Temporal modeling is essential for video understanding due to the dynamic nature of video data, which often contains redundant or irrelevant frames alongside critical events. Several approaches have been proposed to address this challenge by focusing on efficient temporal feature extraction. [14, 134, 135] propose sparse sampling based on dataset biases. While these methods can be effective, they often struggle with complex videos containing multiple events or subtle temporal dependencies, limiting their applicability in real-world scenarios. Several works also adopt a "first temporal, then spatial" strategy to enhance video understanding by training a temporal selection module. TranSTR [146] uses Adaptive Temporal Rationalization to select critical frames, whereas MIST [81] leverages a temporal attention layer to aggregate frame-wise features. These approaches demonstrate the importance of selective temporal processing but often involve complex architectural designs. SeViLa [275], as the pioneering work to leverage LLMs for video QA, proposes a self-refined vision-language model for keyframe selection. However, the optimal approach to bridging video input with LLMs for video question

answering is still an open challenge. Given that the computational overhead of LLMs scales quadratically with the sequence length of inputs, there is a pressing need for more efficient temporal modeling methods. In Chapter 9, we tackle this challenge and propose a question-guided temporal querying Transformer that achieves competitive performance while mitigating the computational overhead.

2.4.3 Video QA with LLMs

Pre-trained LLMs have drawn significant attention in recent years owing to their emergent capabilities of instruction following [46, 222, 298] and human-like commonsense reasoning [129, 154, 270]. In the field of visual understanding, MLLMs attempt to merge the capabilities of LLMs with visual data to address complex vision tasks, such as video QA [140]. Their ability to perform visual reasoning by leveraging the underlying power of LLMs has been demonstrated to surpass traditional models, setting a new baseline in visual comprehension. The application of LLMs to video QA presents both promising opportunities and unique challenges. The inherently multi-event nature of video content introduces complexities not present in static images. Videos require models to navigate and reason across temporal sequences, a task that demands more than just the integration of visual and textual modalities. To address these challenges, the Q-Former emerges as a powerful component to compress spatial visual features [57, 140, 283], thus facilitating the alignment of these features with the reasoning capabilities of LLMs. By adopting and further developing the Q-Former architecture, the method proposed in Chapter 9 follows this direction of utilizing an LLM as a powerful agent for reasoning on video data and explores how to extract visual features from video data and align them to MLLMs.

Chapter 3

Bidimensional Downsampling in Vision Transformer Models

Computer vision tasks such as image classification [99, 213], object detection [196, 198], or semantic segmentation [27, 98] have been tackled for years by employing convolutional neural networks (CNNs). These architectures combine the use of a local operator (*i.e*. the convolution) and strategies for building hierarchical representations through spatial downsampling, which is usually carried out either via pooling layers or by adopting strided convolutions. In recent years, the Transformer architecture [230] has received a relevant interest from the Natural Language Processing (NLP) community [60, 192] and has been adopted to solve Computer Vision tasks as well [19, 67, 224, 299]. In the case of a Transformer, the architecture features an infinite receptive field, which is achieved through the computation of content-based pairwise similarities and, differently from CNNs, does not feature a hierarchical structure.

While a Transformer can achieve non-locality and infinite receptive field without requiring a significant increase in the number of parameters, its overall architecture usually features an increased computational cost in terms of multiply-add operations [67, 224]. This can be explained by the fact that Transformers maintain a full-length sequence across all layers so that the computational cost of

This chapter is related to the publication "Paolo Bruno, Roberto Amoroso, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. Investigating Bidimensional Downsampling in Vision Transformer Models, ICIAP 2022". **Winner of the Best Paper Award**.

running a single layer does not decrease with the layer depth. While this appears to be an issue from a computational point of view, it also hinders the fact that Transformers lacks a structure that is specifically designed for images.

Contributions. In this chapter, we address both issues and investigate the use of bidimensional pooling in Vision Transformers. Our proposal is in line with recent literature which has tackled either the optimization of the Vision Transformer model [44, 121, 220] or the application of one-dimensional pooling [186]. In our approach, the sequence of visual tokens is re-arranged into its original spatial configuration at each architectural block, and bidimensional pooling is then applied to connect different patches together and downsample the sequence length. In this way, we both decrease the computational requirements of the architecture and create a hierarchy in the feature extraction process that resembles that of a CNN. Experiments are carried out on both a small-scale scenario, that of CIFAR-100 [131], and in a larger-scale setup, that of ImageNet [205]. By comparing with both a baseline Vision Transformer and with the usage of one-dimensional pooling, we demonstrate the effectiveness of our proposal, both in terms of accuracy and reduction of the number of operations.

The rest of the chapter is organized as follows: Section 3.1 introduces key preliminary concepts, alongside a detailed description of the proposed model's architecture and methodology. In Section 3.2, we present the adopted datasets, evaluation metrics, implementation setup, and training details. This section also provides a comprehensive analysis of our experimental results.

3.1 Proposed Method

We first revisit the Vision Transformer (ViT) architecture [67] and, in the following sections, introduce our proposal which applies 2D downsampling in ViT models.

3.1.1 Preliminaries

The ViT model [67] has shown that attention and feed-forward mechanisms can be employed to solve image classification tasks. Given an input image $I \in \mathbb{R}^{H \times W \times C}$, where H, W and C denote height, width, and the number of channels, this is transformed into a sequence of N square patches $[\boldsymbol{x}_1^p; \boldsymbol{x}_2^p; ..., \boldsymbol{x}_N^p]$, where $\boldsymbol{x}_i^p \in \mathbb{R}^{P^2 C}$ is the i-th patch of the input image. Being $P \times P$ the resolution of square patches, H and W usually are multiple of P and the number of patches is $N = (H \cdot W)/P^2$. A linear layer is then applied to each flattened patch to project

it to the input dimensionality of the model so that patches can be employed as input tokens for a Transformer encoder.

An additional classification token $\boldsymbol{x}_{\text{class}}$ is usually added to the sequence of patch embeddings. This is implemented as a trainable vector that goes through the Transformer layers and is then projected with a linear layer to predict the final output class. Positional embeddings are then added to the patch embeddings to inject information about the position of patches inside the image. Formally, the model input can be written as:

$$\boldsymbol{z}_0 = [\boldsymbol{x}_{\text{class}}, \boldsymbol{x}_1^p \boldsymbol{E}, \boldsymbol{x}_2^p \boldsymbol{E}, ..., \boldsymbol{x}_N^p \boldsymbol{E}] + \boldsymbol{E}_{\text{pos}}, \tag{3.1}$$

where $\boldsymbol{E}$ indicates the token embedding matrix and $\boldsymbol{E}_{\text{pos}}$ the positional embedding matrix. The Transformer encoder [230] implemented in ViT consists of L identical layers, each being composed of a multi-head self-attention layer (MSA) and a multi-layer perceptron (MLP). Layer normalization (LN) is applied before every layer [235] and residual connections are applied after every MSA and MLP. Given the input sequence $\boldsymbol{z}_0$, the classification output of the model can be written as:

$$\boldsymbol{z}'_l = \mathsf{MSA}(\mathsf{LN}(\boldsymbol{z}_{l-1})) + \boldsymbol{z}_{l-1} \qquad l = 1, ..., L \tag{3.2}$$

$$\boldsymbol{z}_l = \mathsf{MLP}(\mathsf{LN}(\boldsymbol{z}'_l)) + \boldsymbol{z}'_l \qquad l = 1, ..., L \tag{3.3}$$

$$\boldsymbol{y} = \mathsf{LN}(\boldsymbol{z}_L^0), \tag{3.4}$$

where $\boldsymbol{z}_L^0$ is the first output of the last encoder layer, corresponding to $\boldsymbol{x}_{\text{class}}$.

Scaled Dot-Product Attention. The attention function performed in MSA layers can be seen as a mapping between queries and a set of key-value pairs with an output. Queries and keys have size d_k, the values dimension is d_v, and all are obtained as linear projections of the input sequence. The dot-product between the queries and all keys is applied, the resulting vector is divided by $\sqrt{d_k}$ and a softmax function is computed over the sequence. The softmax outputs are then employed to perform a weighted sum of the values. Given the matrix of queries $\boldsymbol{Q}$, and those of keys and values $\boldsymbol{K}$ and $\boldsymbol{V}$, the output of the attention operator is:

$$\mathsf{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\intercal}{\sqrt{d_k}}\right)\boldsymbol{V}. \tag{3.5}$$

Multi-Head Attention. The above-defined attention function is computed for h different sets of keys, values, and queries, each obtained from separate and learned

linear projections. The results after the h parallel operations are concatenated and projected, as follows:

$$\mathsf{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\text{head}_1, ...\text{head}_h)\boldsymbol{W}^O. \tag{3.6}$$

Each head is defined by the following equation:

$$\text{head}_i = \mathsf{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V), \tag{3.7}$$

where $\boldsymbol{W}_i^*$ indicates the parameters of each attention head.

3.1.2 How Pooling Layers Can Help Vision Transformers

The ViT model maintains a fixed-length sequence that passes through all the layers of the network. This choice, although simple, neglects two considerations: (*i*) each layer contributes differently to the accuracy and efficiency of the model, and (*ii*) using a sequence with a fixed length can introduce excessive redundancy, with a consequent increase in memory consumption and FLOPs without a corresponding benefit in performance. A multi-level hierarchical representation that would solve both issues is, indeed, missing. CNNs achieve this through intensive use of the pooling layer (or of the stride mechanism) to reduce the spatial dimension of the inputs [99, 213] and, at the same time, significantly reduce the computational cost and increase the scalability of the model.

Bidimensional Downsampling. Inspired by Pan *et al*. [186], who investigated the usage of one-dimensional pooling in ViT-like structures, we propose to apply *bidimensional* pooling to shrink the patch embeddings sequence and create a hierarchical representation.

Without loss of generality, a max-pooling operation is considered for all the experiments. Clearly, while a 1D max-pooling can only collapse adjacent tokens, in a 2D configuration the kernel window includes all the neighboring elements with respect to the application point and considers the spatial arrangement of tokens in the input image. Our pooling strategy is thus capable of summarizing intermediate features in a spatially-aware manner. The result is a better localization of relevant features inside the feature map. While the rest of the architecture is left unchanged, we also replace the class token by average pooling the entire sequence after the last encoder layer [186].

To perform a 2D operation over a mono-dimensional sequence of intermediate activations, we firstly re-arrange the sequence of activations in matrix form, thus recovering the original spatial arrangement, and then apply the 2D max-pooling

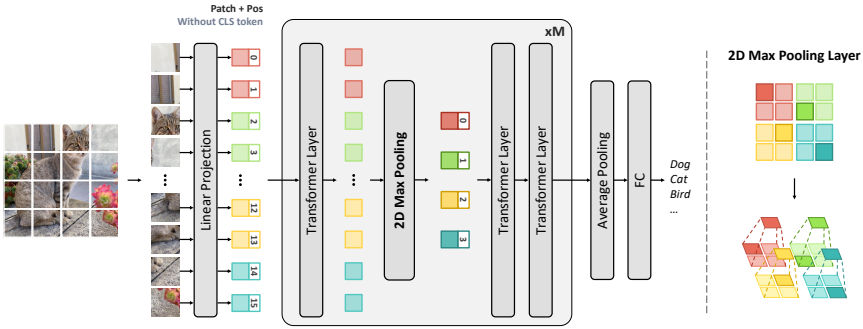


Figure 3.1: Overview of the proposed VT2D architecture. To reduce redundancy and computational complexity, we progressively shrink the patches sequence length through 2D max-pooling. To this aim, we divide the ViT [67] layers into M stages. Each stage is composed of a Transformer layer, a 2D max-pooling, and a variable number of Transformer layers after the pooling operator. Instead of using the CLS token, the output of the last stage is average pooled and given as input to an FC layer to compute the final prediction.

and flatten the result back into a sequence of vectors. The spatial dimensions (H_{out}, W_{out}) obtained after the application of the 2D pooling operation, and before flattening, are thus:

$$H_{out} = \left\lfloor \frac{H_{in} - K}{S} + 1 \right\rfloor \qquad W_{out} = \left\lfloor \frac{W_{in} - K}{S} + 1 \right\rfloor, \tag{3.8}$$

where K indicates the kernel size and S the stride. We do not apply any padding or dilation. Considering that the pooling operation alters the relative spatial positions of the activations, positional embeddings are re-computed and added after each pooling stage.

3.1.3 Overall Architecture

To build our architecture, we conceptually divide the encoder layers into M stages, as shown in Figure 3.1. Before the first stage, the input is arranged in flattened patches, linearly projected into a sequence of tokens. A learnable positional encoding, initialized as in DeiT [67], is also added to inject information about the positions of the patches. Each stage is composed of a Transformer layer, a

max-pooling 2D, and a variable number of Transformer layers. Note that the sequence length is reduced only after the pooling layer. The first Transformer layer input is described by the following equation:

$$\boldsymbol{z}_0 = [\boldsymbol{x}_1^p \boldsymbol{E}, \boldsymbol{x}_2^p \boldsymbol{E}, ..., \boldsymbol{x}_N^p \boldsymbol{E}] + \boldsymbol{E}_{\text{pos}}, \tag{3.9}$$

where $\boldsymbol{E}_{\text{pos}}$ represents the learnable position embedding. We define the output after a layer b which precedes a max-pooling layer, with also the addition of a new positional encoding, as:

$$\hat{\boldsymbol{z}}_b = \mathsf{MaxPool2D}\left(\boldsymbol{z}_b\right) + \boldsymbol{E}_b, \tag{3.10}$$

where $\boldsymbol{z}_b$ is defined as in Equation 3.2 and $\mathbf{E}_b$ is the new learnable positional embeddings applied after the application of the 2D max-pooling. An average pooling is applied after the last Transformer layer of the last stage, then a fully-connected layer (FC) is used to make the final predictions. Formally, predictions can be formulated as:

$$\mathbf{y} = \mathsf{FC}(\mathsf{AvgPool}(\mathsf{LN}(\boldsymbol{z}_L))). \tag{3.11}$$

3.2 Experiments

To evaluate the effectiveness of our proposal, we perform experiments on two image classification datasets and compare our approach with different variants and baselines. In the following, we first provide implementation and experimental details and then describe our experimental results.

3.2.1 Experimental Setting

In the following experiments, we consider different configurations of our ViT-based model equipped with 2D-pooling (which we refer to as VT2D) and compare against a ViT model with no pooling and a ViT-based model with 1D-pooling [186]. In the following, we refer to these baselines as VT and VT1D, respectively.

Datasets and Evaluation Protocol. We perform experiments on the CIFAR-100 dataset [131], which contains 100 classes and 60k images, and the ILSVRC-2012 ImageNet dataset [205], which has $1,000$ classes and 1.3M images. All trained models are compared in terms of FLOPs and number of parameters and evaluated in terms of top-1 and top-5 accuracy on the considered datasets.

Implementation Details. Following recent literature on ViT-based models [67, 224], we devise two model configurations varying the model dimensionality d and the number of attention heads h: Tiny ($d = 192$, $h = 3$) and Small ($d = 384$, $h = 6$). Regardless of the model configuration, we always employ 12 layers, divided in $M = 4$ stages with 3 Transformer layers each.

For the experiments on the CIFAR-100 dataset, we use a batch size of 128 and an initial learning rate of $1.25 \cdot 10^{-4}$, while for the ImageNet dataset, the batch size is set to 1024 and the initial learning rate is equal to $5 \cdot 10^{-4}$. The input image resolution is set to 224×224 for both datasets. For training, we use the AdamW optimizer [165], with momentum and weight decay set to 0.9 and 0.25 respectively, and train the models for 300 epochs on both the datasets. Note that, during the training phase, we use a cosine scheduler, so that the initial learning rate is reached only after 5 warm-up epochs, and a stochastic depth of 0.1 to facilitate convergence. In all experiments, model weights are initialized with a truncated normal distribution.

To obtain the required amount of data to train Transformer-based models, we follow the same data augmentation strategy used to train the DeiT model [224]. In particular, we apply rand-augment [56] and random erasing [301], together with mixup [285] and cutmix [279]. The magnitude and standard deviation of rand-augment are set to 9 and 0.5, respectively. Random erasing is applied with a probability equal to 0.25. We also employ repeated augmentation [8, 101, 224]. We run our experiments on 4 RTX 2080 GPUs.

3.2.2 Experimental Results

Experiments on CIFAR-100. To identify the best strategy to apply the 2D pooling for reducing the model complexity while maintaining competitive performance, we conduct an ablation study in which we vary the kernel size and stride of the pooling layers and the network configuration. The configurations considered differ in terms of the stages in which the pooling layer is applied. In particular, we vary the number of stages, from 1 to 4, and the depth of the stage in which 2D pooling is performed in the model. For all VT2D configurations, we use 2D pooling with stride 2 except when applying the 2D pooling in all four stages of the model, where we use stride equal to 1. As already mentioned, as our baselines, we also consider the VT1D approach, in which 1D pooling with stride 2 is applied at all four stages of the model, and the VT model, which has no pooling layers for downsampling. Results on the CIFAR-100 dataset are reported in Table 3.1, using Tiny and Small configurations. A noteworthy aspect that emerges from the experimental results is

	Pooling Stages	Kernel Size	Params (M)	FLOPs (G)	Top-1 Acc. (%)	Top-5 Acc. (%)
VT-Ti (no pooling)	-	-	5.54	1.25	72.92	92.88
VT1D-Ti-4	0,1,2,3	3	5.58	0.38	72.76	92.67
VT2D-Ti-1	0	3×3	5.55	0.31	71.86	92.03
VT2D-Ti-1	1	3×3	5.55	0.57	73.39	92.94
VT2D-Ti-1	2	3×3	5.55	0.82	73.04	92.78
VT2D-Ti-1	3	3×3	5.55	1.08	71.49	92.53
VT2D-Ti-2	0,2	3×3	5.55	**0.24**	70.31	91.27
VT2D-Ti-2	0,2	2×2	5.55	0.29	70.92	91.82
VT2D-Ti-2	1,3	3×3	5.55	0.54	72.25	92.32
VT2D-Ti-2	1,3	2×2	5.55	0.58	72.17	92.36
VT2D-Ti-4	0,1,2,3	3×3	5.61	0.61	72.87	92.61
VT2D-Ti-4	0,1,2,3	2×2	5.65	0.88	**75.31**	**93.47**
VT-S (no pooling)	-	-	21.70	4.58	75.62	93.01
VT1D-S-4	0,1,2,3	3	21.77	1.39	76.09	93.43
VT2D-S-1	0	3×3	21.71	1.15	75.31	92.32
VT2D-S-1	1	3×3	21.71	2.08	76.59	93.16
VT2D-S-1	2	3×3	21.71	3.02	76.18	93.35
VT2D-S-1	3	3×3	21.71	3.95	75.13	93.34
VT2D-S-2	0,2	3×3	21.71	**0.86**	73.31	91.65
VT2D-S-2	0,2	2×2	21.73	1.04	74.44	92.02
VT2D-S-2	1,3	3×3	21.71	1.97	76.26	92.91
VT2D-S-2	1,3	2×2	21.73	2.13	75.51	93.13
VT2D-S-4	0,1,2,3	3×3	21.83	2.28	75.68	92.26
VT2D-S-4	0,1,2,3	2×2	21.91	3.26	**77.61**	**93.57**

Table 3.1: Experimental results on the CIFAR-100 dataset [131]. For each experiment, we indicate the indexes of network stages in which we perform 1D or 2D pooling and the max-pooling kernel size.

that applying downsampling at early stages brings the most noticeable saving in terms of computational complexity. Moreover, performing a finer-grained pooling by applying kernels of size 2 compared to 3 benefits the most the performance, at the cost of a slightly higher computational complexity.

Our approach with 2D pooling applied at all four stages obtains better performance compared to both the model without pooling and the 1D pooling version,

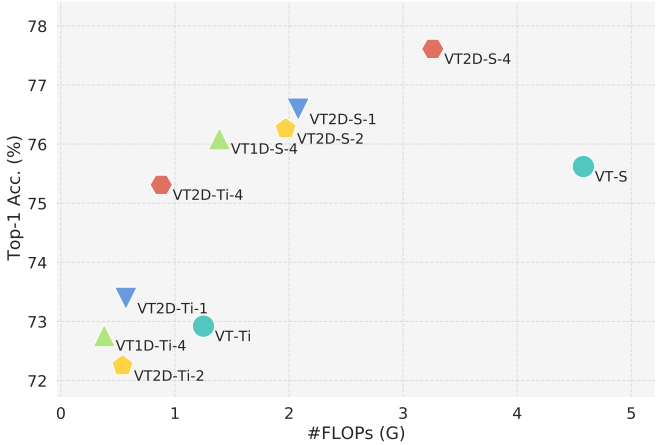


Figure 3.2: Performance comparison in terms of top-1 accuracy and FLOPs on the CIFAR-100 dataset [131].

with a significant reduction of the computational complexity, especially when compared to the VT model without pooling layers. Specifically, it can be noticed that some of the tested configurations with bidimensional pooling in one or two stages perform on par with the VT model in terms of accuracy but require on average 50% fewer FLOPs. The best-performing variant, the VT2D with four stages featuring pooling, brings to an accuracy increase of 2.39% and 1.99% for the Tiny and Small configurations, respectively, while reducing the FLOPs by one third. When comparing with the VT1D version, instead, it can be seen that the best configuration of the VT2D model is always significantly better in terms of accuracy but brings to an increase of the number of FLOPs. However, considering the variants with a single pooling stage, we can notice a computational complexity similar to the 1D pooling version, but with better performance in terms of accuracy, thus further demonstrating the effectiveness of our approach.

In Figure 3.2, we show the performance comparison of our approach and the considered baselines in terms of top-1 accuracy and FLOPs, using both Tiny and Small configurations. From the graph, we can notice that our model obtains the best trade-off between overall performance and computational complexity, outperforming the VT and VT1D models in terms of accuracy while keeping FLOPs comparable or even reduced.

	Pooling Stages	Kernel Size	Params (M)	FLOPs (G)	Top-1 Acc. (%)	Top-5 Acc. (%)
VT1D-Ti-4	0,1,2,3	3	5.75	0.38	67.28	87.70
VT2D-Ti-1	0	3×3	5.72	**0.31**	65.56	86.47
VT2D-Ti-1	1	3×3	5.72	0.57	70.39	**89.84**
VT2D-Ti-4	0,1,2,3	2×2	5.82	0.88	**70.60**	89.83
VT1D-S-4	0,1,2,3	3	22.12	1.40	74.83	92.05
VT2D-S-1	0	3×3	22.06	**1.15**	73.92	91.17
VT2D-S-1	1	3×3	22.06	2.08	**78.19**	**93.87**
VT2D-S-4	0,1,2,3	2×2	22.26	3.26	78.02	93.66

Table 3.2: Experimental results on the ImageNet dataset [205].

Experiments on ImageNet. As a further analysis, we explore the effects of applying 2D pooling in the case of a bigger and more complex dataset than CIFAR-100, and consider the ImageNet dataset. In this analysis, we include a subset of variants previously described, *i.e.* those with best accuracy/FLOPs trade-off, and the baseline model with 1D pooling. Again, for all VT2D configurations, we use 2D pooling with stride equal to 2, except for the model variant that applies bidimensional pooling in all four stages of the network. The results of this comparison are reported in Table 3.2.

Considering the Tiny configuration, our best model is VT2D-Ti-4 which applies a 2D pooling with kernel size 2 in all four stages, followed by VT2D-Ti-1 in which a single bidimensional downsampling is applied in the second stage of the network. Noticeably, VT2D-Ti-1 outperforms the one-dimensional pooling baseline with a slight increase in terms of FLOPs. Similar results can be also observed when turning to the Small configuration. Specifically, the VT2D-S-1 with two-dimensional pooling at the second stage of the network overcomes the VT1D baseline by 3.36% and 1.82% in terms of top-1 and top-5 accuracy, respectively.

Comparison with State-of-the-art Models. As a final analysis, in Table 3.3 we report the comparison of our best variants and other state-of-the-art models based on the Vision Transformer architecture that apply different strategies to achieve efficiency. For the competitors, we use the same notation as for our models. In particular, we consider the knowledge distillation-based approach proposed in [224] (DeiT), two variants of DeiT that additionally perform pruning (one following the strategy proposed in [220] - SCOP, the other the strategy proposed

		CIFAR-100		ImageNet	
	FLOPs (G)	**Top-1 Acc. (%)**	**Top-5 Acc. (%)**	**Top-1 Acc. (%)**	**Top-5 Acc. (%)**
DeiT-Ti [224]	1.25	-	-	72.20	91.10
DeiT-Ti+SCOP [220]	0.80	-	-	68.90	89.00
DeiT-Ti+PoWER [86]	0.80	-	-	69.40	89.20
HVT-Ti-1 [186]	0.64	-	-	69.64	89.40
HVT-Ti-4 [186]	0.38	69.51	91.78	-	-
VT2D-Ti-1	0.57	73.39	92.94	70.39	89.84
VT2D-Ti-4	0.88	75.31	93.47	70.60	89.83
DeiT-S [224]	4.60	-	-	79.80	95.00
DeiT-S+SCOP [220]	2.60	-	-	77.50	93.50
DeiT-S+PoWER [86]	2.70	-	-	78.30	94.00
HVT-S-1 [186]	2.40	74.27	93.07	78.00	93.83
HVT-S-4 [186]	1.39	75.43	93.56	75.23	92.30
VT2D-S-1	2.08	76.59	93.16	78.19	93.87
VT2D-S-4	3.26	77.61	93.57	78.02	93.66

Table 3.3: Comparison with state-of-the-art models on the CIFAR-100 [131] and ImageNet [205] datasets.

in [86] - PoWER), and the monodimensional pooling-based approach proposed in [186] (HVT). From the table, it can be observed that bidimensional pooling is comparable to the considered state-of-the-art approaches both in terms of FLOPs saving and accuracy.

Chapter 4

Self-supervised Learning for ViT Pre-Training

In Chapter 3, we examined the foundational principles of Vision Transformer (ViT) models, their impact on the field of Computer Vision, and strategies to enhance their efficiency. Despite their remarkable success in advancing state-of-the-art performance across various visual tasks, empirical studies reveal that ViTs typically require significantly more training data compared to convolutional neural networks (CNNs) [159]. To address this data-hungry limitation, self-supervised pre-training has emerged as a promising solution, enabling the effective utilization of large-scale unlabeled image datasets.

This chapter delves into self-supervised learning as a pre-training paradigm for ViTs. Self-supervised learning has revolutionized machine learning by enabling models to learn meaningful representations from unlabeled data, thereby reducing reliance on costly manual annotations. Inspired by its success in Natural Language Processing (NLP), where models like BERT [60] have set new benchmarks, researchers have adapted similar methodologies to Computer Vision.

One prominent adaptation is the Masked Image Modeling (MIM) pre-training objective [5], which draws inspiration from BERT [60]. MIM involves masking random image patches and reconstructing the corrupted visual input. While

This chapter is related to the publication "Lorenzo Baraldi*, Roberto Amoroso*, Marcella Cornia, Andrea Pilzer, Lorenzo Baraldi, and Rita Cucchiara, Learning to Mask and Permute Visual Tokens for Vision Transformer Pre-Training, CVIU 2025 (* Equal Contribution)".

several recent studies have refined the MIM approach [34, 96], there has been little exploration of alternative pre-training objectives in the visual domain. In contrast, in the field of NLP, several methods [214, 269] have surpassed the BERT pre-training objective with the introduction of advanced pre-training paradigms that address limitations of previous methods.

Contributions. Drawing inspiration from NLP, we investigate a permutation-based pre-training strategy, which we term Permuted Image Modeling (PIM). This approach autoregressively predicts permuted image patches maintaining contextual bi-directionality without corrupting any part of the input. Despite offering an improvement over the standard MIM-based objective, the autoregressive PIM technique reduces the amount of positional information available for each prediction. To tackle this issue, we propose a Masked and Permuted pre-training solution for Vision Transformers (MaPeT) which leverages auxiliary position information as input during pre-training, thus allowing the model to access the positional information of each input image patch. In addition to the pre-training objective, a crucial aspect of self-supervised vision pre-training is the design of visual targets, used as supervisory signals. While some works have employed low-level and hand-crafted visual features [106, 161, 244], the dominant approach is to employ discrete visual tokens to reconstruct the corrupted input [5, 34, 70, 189]. In this context, although BEiT [5] initially employed DALL-E [194] visual tokens, its performance has been surpassed by VQ-KD, proposed in BEiT v2 [189]. In particular, VQ-KD employs an encoder-decoder architecture that reconstructs CLIP features and is directly trained on ImageNet-1k [59], requiring retraining to achieve satisfactory results on other datasets. In contrast to previous works, we propose k-CLIP, a novel discrete tokenizer that generates visual tokens directly from CLIP features without requiring an ad hoc discrete autoencoder. This innovation simplifies the tokenization process while maintaining high-quality semantic representations.

To evaluate the effectiveness of the proposed pre-training model and tokenizer, we conduct experiments that provide a fair comparison between models, adhering to the same experimental settings. This approach allows to accurately measure the efficacy of each model under consistent conditions and effectively compare their relative strengths and weaknesses. Experimental results demonstrate that our MaPeT model achieves competitive performance and surpasses both mask- and permutation-based image pre-training. Additionally, we show that the visual tokens extracted by our proposed k-CLIP tokenizer exhibit richer semantic information than competitors, outperforming both DALL-E and VQ-KD visual tokens when employed directly for image classification.

The rest of the chapter is organized as follows: Section 4.1 introduces the necessary preliminaries, including key concepts, notations, and an overview of the MIM and PIM pre-training objectives. In Section 4.2, we provide a detailed explanation of our proposed MaPeT pre-training paradigm, accompanied by the k-CLIP visual tokenizer. Finally, Section 4.3 describes the experimental setup and presents both quantitative and qualitative analyses of the results.

4.1 Preliminaries

In this section, we detail two pre-training strategies that are the starting point of our proposal, *i.e.*, Masked Image Modeling (MIM) and Permuted Image Modeling (PIM), and we introduce the terminology used in the rest of the chapter.

Image Patches. We adopt ViT [67] as the backbone network for our architecture. As discussed in Section 3.1.1, ViT splits an image into a sequence of 2D image patches, which are linearly projected to the model embedding space and elaborated through multiple attention blocks. Given an input image $I \in \mathbb{R}^{H \times W \times C}$, this is mapped into a sequence of N square patches $\{\boldsymbol{x}_i^p\}_{i=1}^N$, where $\boldsymbol{x}_i^p \in \mathbb{R}^{P \times P \times C}$ is the i-th patch of the input image. Subsequently, a linear layer is applied to each flattened patch to project it to the input dimensionality of the model D, outputting the patch embeddings $\{\boldsymbol{x}_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbb{R}^D$, which are then added to the learnable 1D positional embeddings $\boldsymbol{E}_{\text{pos}} \in \mathbb{R}^{N \times D}$. The ViT [67] encoder consists of L identical Transformer blocks layers, where the output embeddings of the last layer represent the encoded representations of the N input image patches. In our experiments, we consider an image size of 224×224 with a patch dimension P of 16×16, constituting an input sequence of $14 \times 14 = 196$ patch embeddings.

Visual Tokens. In self-supervised pre-training, learned supervisory signals are typically used to effectively pre-train the visual backbone. In this work, we employ visual tokens as supervisory signals during the pre-training phase. Specifically, we represent targets as a discrete token sequence, accomplished by utilizing a visual tokenizer $\mathcal{T}(I)$ on the input image I. The visual tokenizer maps the image pixels onto a visual codebook (or vocabulary), generating a sequence of tokens $\boldsymbol{v} = [v_1, \ldots, v_N] \in \mathcal{V}^{(H/P) \times (W/P)}$, where $\mathcal{V}$ represents the vocabulary containing discrete token indices. Our approach employs a 14×14 grid of visual tokens to represent each image, while the vocabulary size is set to $|\mathcal{V}| = 8192$.

Masked Image Modeling (MIM). Inspired by the Masked Language Modeling strategy utilized in BERT [60], the MIM paradigm is a pre-training technique

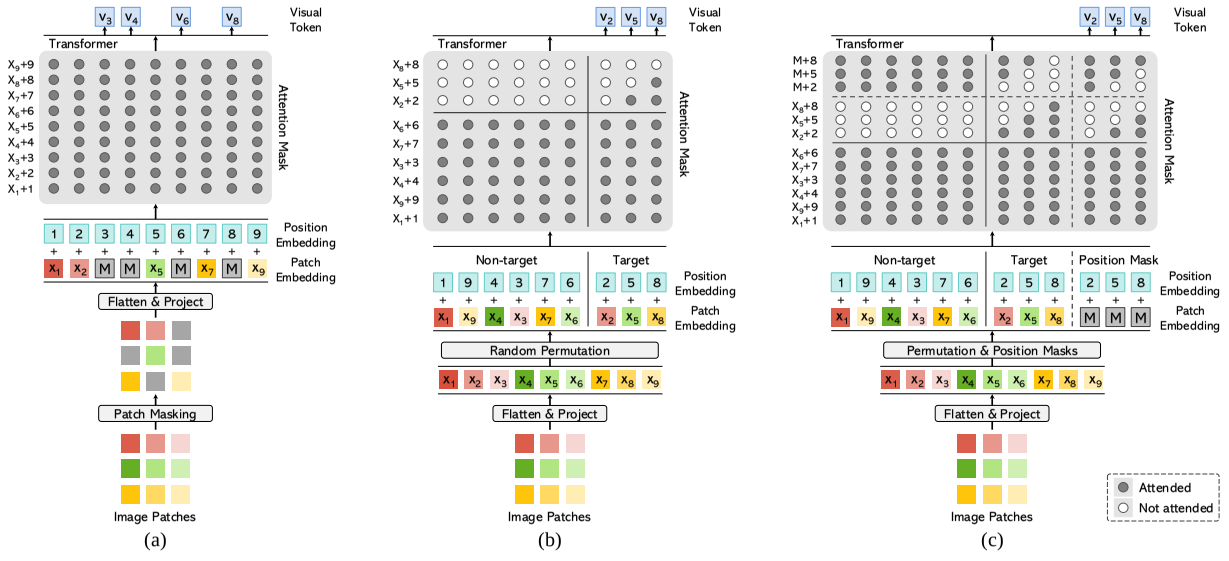


Figure 4.1: (a) Masked Image Modeling (MIM). (b) Permuted Image Modeling (PIM). (c) Masked and Permuted pre-training for Vision Transformers (MaPeT). While MIM reconstructs visual tokens from randomly masked image patches, PIM autoregressively predicts tokens associated with permuted image patches. MaPeT uses PIM to capture intra-patch dependency and takes auxiliary position information as input to ensure that the model sees a full sequence of patches at each target position.

for vision tasks that aims to recover visual information from a corrupted input image. This is achieved by randomly masking a portion of the image patches and predicting the visual tokens related to the corrupted region of the input, as depicted in Figure 4.1 (a). Like Masked Language Modeling, MIM has some inherent disadvantages. Firstly, a mask token M is introduced during pre-training and never used during fine-tuning, leading to a discrepancy in the pre-training and fine-tuning phases. Secondly, given the masked tokens $\bar{\boldsymbol{x}}$ and the uncorrupted context $\tilde{\boldsymbol{x}}$, the probability $p(\bar{\boldsymbol{x}} \mid \tilde{\boldsymbol{x}})$ is typically factorized, assuming the independence of reconstructed patches. To address these issues, the NLP literature has investigated a permutation-based variant [269] that can reduce the disadvantages of standard Masked Language Modeling. These results motivate us to investigate the application of this strategy to vision tasks.

Permuted Image Modeling (PIM). The PIM self-supervised pre-training objective differs from MIM in two key components: the use of patch permutations and attention masking to capture bidirectional contexts. Specifically, PIM permutes patch embeddings, splits them into non-target and target patches, and predicts the visual tokens associated with the target patches using an auto-regressive approach. Attention masking is then applied to reduce the visibility of patches in the attention process, allowing a target patch to not access its contextual information (*i.e.*, its content) during prediction while remaining visible to patches that come after it in the permuted order. A representation of the process is shown in Figure 4.1 (b).

Formally, given an input image I, we extract the patch embeddings $\{\boldsymbol{x}_i\}_{i=1}^{N}$, and tokenize it into N visual tokens $\{v_i\}_{i=1}^{N}$. We define $\mathcal{Z}_t$ as the set of all $N!$ possible permutations of the length-N index sequence $\{1, 2, \ldots, N\}$. Given a permutation $\boldsymbol{z}$, we use z_t and $\boldsymbol{z}_{<t}$ to denote the t-th element and the first $t-1$ elements of $\boldsymbol{z}$, respectively. After applying $\boldsymbol{z}$, the permuted patch embeddings are fed into an L-layer ViT backbone to extract the final hidden representations. For each input embedding $\boldsymbol{x}_{z_t}$ at position z_t in the considered permutation $\boldsymbol{z}$, we use a softmax classifier to predict the corresponding visual token. The goal of PIM is to maximize the following log-likelihood objective:

$$\max_{\theta} \sum_{I \in \mathcal{D}} \mathbb{E}_{\boldsymbol{z} \in \mathcal{Z}_t} \left[\sum_{t=c+1}^{N} \log p_\theta(v_{z_t} | \boldsymbol{x}_{z_{<t}}) \right], \tag{4.1}$$

where θ represents the model parameters, $\mathcal{D}$ denotes the training dataset, $\boldsymbol{x}_{z_{<t}}$ are the only patch embeddings that are visible at position z_t, and c is a cutting point applied to split the permutation $\boldsymbol{z}$ in a subset of non-target patch embeddings $\boldsymbol{x}_{z_{\leq c}}$ and target patch embeddings $\boldsymbol{x}_{z_{>c}}$. The aim of c is to reduce the number of visual

tokens to be predicted, thus mitigating optimization difficulties.

While MIM preserves full positional information of each image patch, during the prediction process, PIM can access the contextual and positional information only of the $t-1$ patches preceding z_t, as shown in Equation 4.1. Given this lack of full positional information, PIM introduces an input discrepancy between pre-training and fine-tuning, underscoring the need for a pre-training technique that combines the advantages of both MIM and PIM while mitigating their respective limitations.

4.2 Proposed Method

In this section, we present MaPeT, a novel pre-training paradigm that combines masked and permuted image modeling strategies. In addition, we also introduce the k-CLIP visual tokenizer, which exploits discretized CLIP features to produce visual tokens.

4.2.1 Masked and Permuted Pre-Training

Drawing inspiration from NLP literature [214, 269], we propose a novel pre-training methodology called **Ma**sked and **Pe**rmuted Vision **T**ransformer (MaPeT) which builds on the strengths of both MIM and PIM to enhance performance on vision tasks. In particular, our approach overcomes the independence assumption of reconstructed patches, thus capturing intra-patch dependencies more effectively. Moreover, MaPeT incorporates auxiliary position embeddings during pre-training, enabling the model to access position information for all patches, thereby resolving the pre-training fine-tuning discrepancy introduced by PIM. Figure 4.1 (c) shows an overview of the MaPeT approach.

Given a permutation $\boldsymbol{z}$ and a cutting point c, MaPeT can predict the visual token associated with an input patch embedding $\boldsymbol{x}_{z_t}$ by leveraging the content and position of the preceding $\boldsymbol{x}_{z_{<t}}$, as well as the position of the subsequent target embeddings $\boldsymbol{x}_{z_{>t}}$. To this end, we introduce the concept of learnable masked token $M \in \mathbb{R}^D$, which is used to express the positional information of $\boldsymbol{x}_{z_{>c}}$. By repeating $N-c$ times the token M, we obtain $\{M_i\}_{i=c+1}^{N}$ identical masked tokens, which are then summed to the positional embedding $\{E^i_{\text{pos}}\}_{i=z_{c+1}}^{z_N}$ of each target $\boldsymbol{x}_{z_{>c}}$. These resulting position-aware masked tokens $\boldsymbol{M}_{\text{pos}} = \{M_{c+1}+E^{z_{c+1}}_{\text{pos}}, \dots, M_N+E^{z_N}_{\text{pos}}\}$ are concatenated to the input sequence of patch embeddings $\boldsymbol{H}^0$ thus obtaining the augmented input $\boldsymbol{H}^0_M = [\boldsymbol{H}^0, \boldsymbol{M}_{\text{pos}}]$. Note

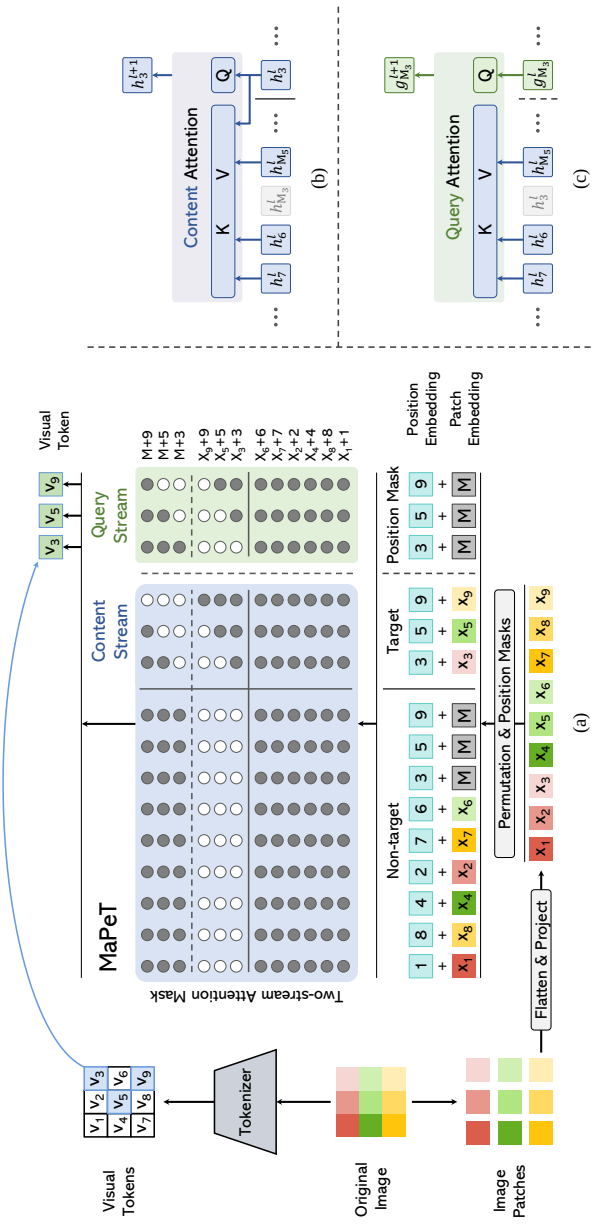


Figure 4.2: Overview of our MaPeT pre-training (a), with content stream attention (b), which follows the standard self-attention mechanism, and query stream attention (c), which lacks information about the content of the patch embedding $\boldsymbol{x}_{z_t}$ whose visual token v_{z_t} is to be predicted. The blue and green masks in (a) are the content and query attention masks employed in the two-stream self-attention. The query stream reuses the attention keys and values from the content stream. During pre-training, the output of the query stream serves as the model's output. At the fine-tuning step, we remove the query stream, and only the content stream is utilized, reverting to a standard ViT backbone.

that the positional embeddings are permuted according to the same permutation $\boldsymbol{z}$ applied to the patches. For convenience, we introduce $\boldsymbol{M}_{z_{\geq t}}$ which represents the subset $\{M^i_{\text{pos}}\}^{N-c}_{i=\max(1,\, t-c)}$. Intuitively, when $t \leq c$, we have that $\boldsymbol{M}_{z_{\geq t}}$ comprises all $\boldsymbol{M}_{\text{pos}}$. Conversely, when $t > c$, $\boldsymbol{M}_{z_{\geq t}}$ only includes the position-aware masked tokens related to $\boldsymbol{x}_{z_{\geq t}}$.

The training objective of our MaPeT model is to maximize the log-likelihood of predicting the visual token v_{z_t} associated with the patch embedding $\boldsymbol{x}_{z_t}$ given $\boldsymbol{x}_{z_{<t}}$:

$$\max_{\theta} \sum_{I \in \mathcal{D}} \mathbb{E}_{\boldsymbol{z} \in \mathcal{Z}_t} \left[\sum_{t=c+1}^{N} \log p_\theta(v_{z_t} | \boldsymbol{x}_{z_{<t}}, \boldsymbol{M}_{z_{\geq t}}) \right]. \tag{4.2}$$

By doing so, MaPeT allows the patch embedding $\boldsymbol{x}_{z_t}$ to attend to contextual information of the patch embeddings $\boldsymbol{x}_{z_{<t}}$ as well as the positional information of $\boldsymbol{x}_{z_{>t}}$. This approach compensates for the position discrepancy of PIM and provides the model with information about the position of the target patches whose visual tokens are to be predicted, as illustrated in Figure 4.2.

Two-stream Self-attention Pre-training. Since the target patch embeddings $\boldsymbol{x}_{z>c}$ follow the permuted order, the next predicted patch can occur in any position. As a consequence, masking the attention matrix of a ViT encoder, instead of corrupting the input like in MIM, makes the backbone architecture non-trivial. To implement PIM on a ViT backbone, we adopt the two-stream self-attention mechanism introduced by XLNet [269]. Specifically, our model consists of two attention streams: a query stream and a content stream. The query stream $g_\theta(\boldsymbol{x}_{z_{<t}}, \boldsymbol{M}_{z_{\geq t}})$ accesses the content of the previous patches in the permuted sequence at a given position t. However, it does not access the content of $\boldsymbol{x}_{z_t}$, only viewing the position $\boldsymbol{M}_{z_{\geq t}}$ of the subsequent patches. In contrast, the content stream $h_\theta(\boldsymbol{x}_{z_{\leq t}}, \boldsymbol{M}_{z_{>t}})$ encodes the content of both the previous elements in the sequence and the element in position z_t, with positional information $\boldsymbol{M}_{z_{>t}}$ of the remaining patches. The input of the first query stream layer $\boldsymbol{g}^{(0)}$ consists of the masked elements $\boldsymbol{M}_{\text{pos}}$, which encode the position of the target patch embeddings $\boldsymbol{x}_{z_{>c}}$. Instead, the input of the first content stream layer $\boldsymbol{h}^{(0)}$ is the augmented input sequence $\boldsymbol{H}^0_M$. Formally, for each Transformer layer l with $l = 1, \ldots, L$, the attention mechanism of both content and query streams can be defined as follows:

$$\begin{aligned} h^{(l)}_{z_t} &\leftarrow \mathsf{Attention}(\boldsymbol{Q} = h^{(l-1)}_{z_t}, \boldsymbol{KV} = (\boldsymbol{h}^{(l-1)}_{z_{\leq t}}, \boldsymbol{h}^{(l-1)}_{\boldsymbol{M}_{z_{>t}}}); \theta), \\ g^{(l)}_{z_t} &\leftarrow \mathsf{Attention}(\boldsymbol{Q} = g^{(l-1)}_{z_t}, \boldsymbol{KV} = (\boldsymbol{h}^{(l-1)}_{z_{<t}}, \boldsymbol{h}^{(l-1)}_{\boldsymbol{M}_{z_{\geq t}}}); \theta), \end{aligned} \tag{4.3}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are respectively queries, keys, and values of the attention operator. Both query and content streams use separate attention masks to limit the visible contextual and positional information for each patch. During pre-training, the output of the query stream is used as the model output. At the fine-tuning step, the query stream is dropped, and only the content stream is used, returning to the standard ViT backbone. Figure 4.2 shows an illustration of the content stream (b), the query stream (c), and how they are integrated into our MaPeT architecture.

Attention Masking. In order to limit the number of visible patches in the content and query attention operations, as described in Equation 4.3, MaPeT leverages attention masking. In particular, the permutation z influences the creation of two distinct attention masks: one for the content attention stream and one for the query attention stream. The content mask guarantees that only patch embeddings $\boldsymbol{x}_{z_{\leq t}}$ and positional tokens $\boldsymbol{M}_{z_{>t}}$ are visible to the patch embedding $\boldsymbol{x}_{z_t}$ in the content stream. On the other hand, the query mask ensures that only patch embeddings $\boldsymbol{x}_{z_{<t}}$ and positional tokens $\boldsymbol{M}_{z_{\geq t}}$ are visible to $\boldsymbol{x}_{z_t}$ in the query stream.

4.2.2 k-CLIP: Discretized CLIP-based Tokenizer

The role of visual tokenizers in pre-training pipelines is significant, as they provide crucial guidance for downstream fine-tuning outcomes. To possibly reduce the overhead of tokenizer retraining of previously proposed approaches [189], we explore the impact of directly utilizing discretized CLIP features. In particular, we propose a novel visual tokenizer, called k-CLIP, that employs discretized CLIP features as visual tokens. Our method does not rely on any pre-training or supervised data to create the visual tokens, thus enabling pre-training without access to large amounts of labeled data or a particular pre-training dataset. Furthermore, the use of CLIP features enables our method to capture high-level visual semantics that are more meaningful for downstream tasks, further improving the performance of the learned representations.

Specifically, we sample visual features from the ImageNet dataset [59] using the CLIP model [191] and cluster them using k-means to obtain $|\mathcal{V}| = 8192$ centroids. During pre-training, these centroids are indexed to retrieve the corresponding visual tokens for prediction. Formally, our visual tokenizer $\mathcal{T}(I)$ consists of a ResNet-based CLIP visual encoder and a k-means model. The visual encoder $f_v : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H_c \times W_c \times D_c}$ maps an input image I to a grid of visual features that correspond to the activations produced by the last convolutional layer of the CLIP backbone. The visual features are then reshaped to a $N_c \times D_c$ matrix, where $N_c = H_c \times W_c$. In our method, the CLIP visual encoder generates visual features

of shape $N_c \times D_c = 196 \times 4096$, which are subsequently indexed by k-means and mapped to a sequence of 196 discrete visual tokens $\boldsymbol{v} = [v_1, \ldots, v_N] \in \mathcal{I}$. The set of visual tokens $\boldsymbol{v} = \{v_i\}_{i=1}^{N}$ is defined by the k-means centroid indexes $\mathcal{I} = \{1, \ldots, |\mathcal{V}|\}$. To mitigate the computational complexity, we randomly sample 2% of the approximately 250 million D_c-dimensional CLIP features extracted from ImageNet. The sampled feature collection is then used to fit the k-means clustering model.

4.3 Experiments

This section presents a thorough experimental evaluation of the proposed methodology, focusing on its effectiveness and robustness across various tasks and benchmarks. The evaluation begins with a detailed description of the experimental setup. Subsequently, we delve into specific analyses, including comparisons of pre-training objectives, state-of-the-art benchmarks, semantic segmentation performance, cross-domain transfer learning, and ablation studies on critical model hyperparameters. Finally, we explore the impact of visual tokenizers on model performance and include qualitative insights into the semantic coherence of our k-CLIP tokenizer.

4.3.1 Experimental Setup

Self-supervised pre-training literature has introduced several training and fine-tuning procedures that encompass different tokenizers, hyperparameters, and visual backbones. To ensure a fair and unbiased comparison among these pre-training algorithms, we opt to evaluate our MaPeT method, as well as the related approaches in the literature, using identical experimental configurations. This strategy allows for the isolation of the algorithmic factor in the experiments and promotes unbiased comparisons of pre-training objectives. To assess the effectiveness of the pre-trained objective under consideration, we fine-tune our models on a downstream classification task. Additionally, following recent literature, we perform fine-tuning experiments for semantic segmentation to assess the generalization capability of the proposed pre-training objective for pixel-level predictions.

Pre-training Setup. We first investigate the influence of the visual tokenizer and evaluate the proposed k-CLIP against VQ-KD [189]. To minimize the computational effort, we pre-process the ImageNet-1k training dataset for both the tokenizers and store the visual tokens associated with each image. For this reason,

our pre-training augmentation policy only includes color jittering to preserve image patch positions corresponding to the pre-extracted visual tokens.

We compare MaPeT against different pre-training objectives. In particular, we employ a ViT-based backbone that is pre-trained according to the pure MIM objective formulation. Note that this differs from the pre-training strategy proposed by the BEiT approach [5] as it lacks the block-wise masking algorithm, which progressively extracts multiple blocks of patches until 40% of the positions are masked. In our MIM-based pre-training, we replace the block-wise masking strategy with a random patch masking approach, thus keeping it similar to a BERT-like solution [60] applied to Computer Vision tasks. Analogously, we also pre-train a ViT backbone through a double-stream architecture according to the PIM objective described in Section 4.1.

Moreover, we consider the standard BEiT model [5] (*i.e.*, a MIM-based pre-training with block-wise masking) and CAE [34]. The CAE method employs an encoder-decoder architecture where the encoder processes only visible image patches (50% of the entire image), while the remaining 50% is masked. A latent contextual regressor predicts the masked representation based on the encoder output, and a lightweight decoder processes the output of the regressor, which is then used to predict the visual token of the related masked patches.

Image Classification Setup. During the classification fine-tuning stage, the final hidden layer of the ViT-based backbone extracts features that are then combined via average pooling to generate a global image representation. This representation is subsequently fed into a softmax classifier. Following the linear probing experiment reported in [5], we also train a linear classifier head over the output representation produced by the frozen pre-trained backbone. We design three different model variants based on Vision Transformer [67], *i.e.*, ViT-Tiny (ViT-T), ViT-Small (ViT-S), and ViT-Base (ViT-B). Our MaPeT model is trained by setting the cutting point c to 50, 50, and 60, respectively. In Section 4.3.6, we propose a comprehensive analysis of the impact of varying cutting point values on model performance.

Hyperparameters and Linear Probing. In Table 4.1, we schematize the experimental settings adopted during the pre-training phase. The complete experimental configuration for fine-tuning our classification models is outlined in Table 4.2. Linear probing has been a widely considered proxy for assessing the effectiveness of self-supervised pre-training models. In accordance with the approach outlined in [5], we train a linear classifier head over the image-level representation output produced by the frozen pre-trained backbone. We use the class labels of the images to train the aforementioned classifier head. We train for 50 epochs using a batch

Hyperparameter	Tiny Size	Small Size	Base Size
Hidden size	192	384	768
FFN inner hidden size	768	1536	3072
Attention heads	3	6	12
Layers		12	
Attention head size		64	
Layer scale		0.1	
Patch size		16×16	
Training epochs		300	
Batch size		2048	
Optimizer		AdamW	
Adam ϵ		1e-8	
Adam β		(0.9, 0.999)	
Peak learning rate		1.5e-3	
Minimal learning rate		1e-5	
Warmup learning rate		1e-6	
Learning rate schedule		Cosine	
Warmup epochs		10	
Gradient clipping		3.0	
Dropout		✗	
Stoch. depth		0.1	
Weight decay		0.05	
Input resolution		224×224	
Color jitter		0.4	

Table 4.1: Hyperparameters for pre-training MaPeT on the ImageNet-1k [59] dataset.

size of 1024, AdamW [165] as optimizer, and a learning rate of $4e{-}3$ with cosine decay. The weight decay is set to $1e{-}4$. Our pre-training augmentation strategy incorporates random resizing of crops, horizontal flipping during training, and central crops during evaluation.

Semantic Segmentation Setup. Semantic segmentation is a pixel-wise classification task that predicts semantic labels for each input image pixel. Our experimental framework follows the setting proposed in BEiT v2 [189] and utilizes the ADE20K dataset [302, 303] as a comprehensive benchmark comprising 25K images spanning 150 semantic categories. For the segmentation architecture, we employ

Hyperparameter	Tiny Size	Small Size	Base Size
Peak learning rate	2.5e-4	5e-3	5e-4
Fine-tuning epochs	300	200	100
Layer-wise learning rate decay	✗	0.65	0.65
Batch size		1024	
Warmup epochs		20	
Optimizer		AdamW	
Adam ϵ		1e-8	
Adam β		(0.9, 0.999)	
Minimal learning rate		1e-6	
Warmup learning rate		1e-6	
Learning rate schedule		Cosine	
Weight decay		0.05	
Label smoothing ε		0.1	
Stoch. depth		0.1	
Dropout		✗	
Gradient clipping		✗	
Erasing prob.		0.25	
Input resolution		224×224	
Rand Augment		9/0.5	
Mixup prob.		0.8	
Cutmix prob.		1.0	

Table 4.2: Hyperparameters for fine-tuning MaPeT on the ImageNet-1k [59] dataset.

UperNet [254] task layer and fine-tune the model for 160K iterations with input images resized to a resolution of 512 × 512. The results are reported in terms of mean Intersection over Union (mIoU). To facilitate reproducibility, Table 4.3 includes a detailed list of training hyperparameters.

Computational Requirements. The pre-training experiments conducted with our MaPeT model involve the utilization of different GPU configurations. Specifically, the ViT-T, ViT-S, and ViT-B models require the deployment of 16, 32, and 64 GPUs, respectively. The pre-training process for each model takes approximately one day to complete. In contrast, during the fine-tuning phase, we employ 4 GPUs for the ViT-T and ViT-S models, and 16 GPUs for the ViT-B model. The fine-tuning duration for each model is 48 hours, 36 hours, and 12 hours, respectively. For all experiments, we utilize an NVIDIA V100 16GB GPU architecture.

Hyperparameter	Tiny / Small / Base Size
Input resolution	512×512
Peak learning rate	5e-5
Fine-tuning steps	160K
Batch size	16
Optimizer	AdamW
Adam ϵ	1e-8
Adam β	(0.9, 0.999)
Layer-wise learning rate decay	0.75
Minimal learning rate	0
Warmup learning rate	5e-11
Learning rate schedule	Linear
Warmup steps	1500
Weight decay	0.05
Stochastic depth	0.15
Dropout	✗

Table 4.3: Hyperparameters for fine-tuning MaPeT on ADE20K [302, 303] for semantic segmentation tasks.

4.3.2 Pre-training Objectives Comparison

To validate the assumptions made on the pre-training objectives presented in Section 4.2, we conduct a comparison of MIM, PIM, and our proposed MaPeT, as shown in Table 4.4. This comparison evaluates the top-1 accuracy and linear probe accuracy of these approaches. The results indicate that MIM pre-training is less effective compared to PIM due to the input noise introduced by masked tokens and the independent reconstruction of patches. PIM outperforms MIM in most comparisons. For instance, when employing ViT-T with k-CLIP and ViT-S with VQ-KD, PIM achieves improvements of 0.4% and 0.1% in classification accuracy, respectively, over MIM. However, PIM demonstrates comparable performance to MIM when applied to the ViT-B backbone. Furthermore, our proposed MaPeT consistently outperforms PIM in nearly all cases. It exhibits accuracy gains of 0.2%, 0.3%, and 0.7% in top-1 accuracy, as well as 2.8%, 2.1%, and 0.5% in linear probe accuracy respectively on ViT-T and ViT-S when using the k-CLIP tokenizer, and on ViT-B when employing the VQ-KD tokenizer. These findings

		ViT-T		*ViT-S*		*ViT-B*	
Method	**Tokenizer**	**Top-1 Acc. (%)**	**Linear Probe Acc. (%)**	**Top-1 Acc. (%)**	**Linear Probe Acc. (%)**	**Top-1 Acc. (%)**	**Linear Probe Acc. (%)**
MIM	VQ-KD	74.6	55.3	82.0	**69.9**	83.8	72.3
PIM	VQ-KD	**74.9**	56.4	82.1	68.7	83.7	73.3
MaPeT	VQ-KD	74.5	**59.6**	**82.2**	69.8	**84.4**	**73.8**
MIM	k-CLIP	74.9	60.4	82.0	71.1	83.3	72.3
PIM	k-CLIP	75.3	59.7	81.8	69.5	83.3	**73.5**
MaPeT	k-CLIP	**75.5**	**62.5**	**82.1**	**71.6**	**83.6**	**73.5**

Table 4.4: Fine-tuning results of different pre-training objectives in terms of top-1 and linear probe accuracy on ImageNet-1k. We report the results using both the VQ-KD and k-CLIP tokenizers.

underscore the significance of addressing position inconsistency between pre-training and fine-tuning, particularly in the context of permutation-based image pre-training.

4.3.3 Comparison with State-of-the-art Models

Table 4.5 presents a comprehensive analysis of the performance of MaPeT, BEiT, and CAE in terms of top-1 accuracy and linear probe accuracy across all ViT-based backbones. Firstly, our pre-trained MaPeT model showcases significant performance improvements across all Tiny, Small, and Base backbones compared to ViT-based models trained with random initialization, as evidenced in Table 4.5. Secondly, our results demonstrate that both BEiT and MaPeT outperform CAE. We hypothesize that the CAE encoder, which only observes 50% of the total sequence during pre-training, may suffer from position discrepancies between the pre-training and fine-tuning phases. Thirdly, it is noteworthy that BEiT improves overall results compared to MIM in Table 4.4. We attribute this improvement to the blockwise masking technique employed by BEiT. This technique follows the principle of image spatial locality, which posits that adjacent patches exhibit similarities in terms of visual information. By employing blockwise masking, the density of uncorrupted visual content is increased, while the noise introduced by masked tokens is concentrated in fewer locations instead of being sparsely distributed. Furthermore, MaPeT consistently outperforms all competitors across the three model variants when employing the k-CLIP tokenizer. Specifically, our model achieves top-1 accuracy margins of 0.7%, 0.2%, and 0.3% against BEiT on

		ViT-T		*ViT-S*		*ViT-B*	
Method	**Tokenizer**	**Top-1 Acc. (%)**	**Linear Probe Acc. (%)**	**Top-1 Acc. (%)**	**Linear Probe Acc. (%)**	**Top-1 Acc. (%)**	**Linear Probe Acc. (%)**
ViT [67]	-	73.7	-	79.8	-	81.8	-
CAE [34]	VQ-KD	73.4	52.4	81.6	63.4	83.5	69.4
BEiT [5]	VQ-KD	**75.0**	**62.0**	**82.2**	**72.6**	**84.4**	**75.0**
MaPeT	VQ-KD	74.5	59.6	**82.2**	69.8	**84.4**	73.8
CAE [34]	k-CLIP	73.7	53.9	81.5	62.9	82.7	67.6
BEiT [5]	k-CLIP	74.8	61.5	81.9	71.1	83.3	73.3
MaPeT	k-CLIP	**75.5**	**62.5**	**82.1**	**71.6**	**83.6**	**73.5**

Table 4.5: Fine-tuning results compared to other self-supervised pre-training approaches, in terms of top-1 and linear probe accuracy on the ImageNet-1k dataset.

ViT-T, ViT-S, and ViT-B, respectively, while exhibiting margins of 1.8%, 0.6%, and 0.9% against CAE across the same three backbones. In contrast, MaPeT demonstrates comparable results to BEiT when employing the VQ-KD tokenizer on the ViT-S and ViT-B backbones.

We also report the comparison of our best variants and other state-of-the-art self-supervised pre-training models in Table 4.6. Note that the variability in the experimental settings and the different supervisory signals used in each approach may affect the fairness of the comparisons. Notably, although current literature has not extensively explored performance benchmarking on the ViT-S backbone, the Small version of our MaPeT model outperforms BEiT [5] and CAE [34] by 0.5% and 0.2% respectively, yielding the best performance for both k-CLIP and VQ-KD. In the case of the ViT-B backbone, MaPeT surpasses most of the considered approaches especially when using the VQ-KD tokenizer, except for BEiT v2 [189] which gets slightly better results. This performance gap can be explained by the lack of data augmentation in our model pre-training, which can significantly increase performance but at a higher computational cost.

The results in Table 4.6 are obtained after pre-training the models for 300 epochs. For completeness, in the lower part, we report the performance of other methods when pre-trained for a considerably large number of epochs (*i.e.*, 800 and 1600). While a direct comparison using different pre-training epochs may not be completely informative, MaPeT still proves to perform better than other methods pre-trained for a larger number of epochs such as BEiT and CAE.

		ViT-S	*ViT-B*
Method	**# Epochs**	**Top-1 Acc. (%)**	**Top-1 Acc. (%)**
BEiT [5]	300	81.7	82.9
CAE [34]	300	82.0	83.6
SplitMask [70]	300	-	83.6
MaskFeat [244]	300	-	83.6
PeCo [66]	300	-	84.1
MVP [245]	300	-	84.4
BEiT v2 [189]	300	-	**85.0**
MaPeT (k-CLIP)	300	82.1	83.6
MaPeT (VQ-KD)	300	**82.2**	84.4
BEiT [5]	800	-	83.2
CAE [34]	800	-	83.8
CAE [34]	1600	-	83.9
BEiT v2 [189]	1600	-	85.5

Table 4.6: Comparison with state-of-the-art self-supervised pre-training models in terms of top-1 accuracy on the ImageNet-1k dataset. We report the number of per-training epochs. Results of competitors are extracted from the original papers.

4.3.4 Semantic Segmentation Results

To evaluate the effectiveness of MaPeT on dense prediction tasks, we present its performance on the semantic segmentation benchmark ADE20K using the UperNet framework with the VQ-KD tokenizer. Table 4.7 presents the mIoU scores achieved with various ViT backbones and pre-training strategies. Notably, MaPeT surpasses both MIM and PIM objectives, delivering significant improvements of $+1.4\%$ and $+0.9\%$ on ViT-S, and $+1.1\%$ and $+1.4\%$ on ViT-B, respectively. Compared to CAE [34] and BEiT [5], MaPeT achieves the highest mIoU across ViT-S and ViT-B backbones, while maintaining competitive performance on ViT-T.

These results underscore the superiority of MaPeT over existing pre-training objectives and self-supervised approaches, particularly in addressing positional inconsistencies during pre-training and fine-tuning, which is critical for dense tasks like semantic segmentation. The robust generalization demonstrated by MaPeT highlights its potential as an effective pre-training strategy for segmentation tasks.

Method	Tokenizer	mIoU (%)		
		ViT-T	*ViT-S*	*ViT-B*
MIM	VQ-KD	39.0	45.6	49.3
PIM	VQ-KD	**40.2**	46.1	49.0
MaPeT	VQ-KD	39.3	**47.0**	**50.4**
CAE [34]	VQ-KD	39.2	45.9	50.2
BEiT [5]	VQ-KD	**39.9**	46.7	50.3
MaPeT	VQ-KD	39.3	**47.0**	**50.4**

Table 4.7: Semantic segmentation results in terms of mIoU on ADE20K, comparing MaPeT with alternative pre-training objectives (top) and other self-supervised pre-training approaches (bottom).

Method	Tokenizer	*Linear Probe Acc. (%)*		
		Stanford-Cars	**Food-101**	**FGVC-Aircraft**
CAE [34]	VQ-KD	53.6	81.4	40.5
BEiT [5]	VQ-KD	64.5	**86.9**	44.9
MaPeT	VQ-KD	**68.5**	**86.9**	**46.8**

Table 4.8: Comparison of model performance on cross-domain transfer learning. We measure the linear probe accuracy of our proposed MaPeT model compared to BEiT and CAE on three distinct data domains: Stanford-Cars [130], Food-101 [11], and FGVC-Aircraft [172].

4.3.5 Cross-domain Transfer Learning

As an additional analysis, we examine the generalization capabilities of our proposed self-supervised pre-training technique, compared to BEiT [5] and CAE [34]. All the considered architectures employ the VQ-KD visual tokenizer [189] and undergo pre-training on the ImageNet-1k dataset [59]. Subsequently, they are fine-tuned on three distinct data domains, namely Stanford-Cars [130], Food-101 [11], and FGVC-Aircraft [172]. These datasets are chosen to represent diverse and real-world scenarios, ranging from object recognition in the automotive domain to food and aircraft classification. By employing a linear probe evaluation, we can quantitatively measure the ability of our pre-trained model to transfer knowledge and adapt to new tasks without fine-tuning.

The results, presented in Table 4.8, clearly demonstrate the superior perform-

Cutting Point c	Reconstruction Ratio	Tokenizer	Top-1 Accuracy (%)
30	85%	VQ-KD	84.2
40	80%	VQ-KD	84.3
50	75%	VQ-KD	84.3
60	70%	VQ-KD	**84.4**
98	50%	VQ-KD	84.1

Table 4.9: Ablation study on the impact of the reconstruction factor (%) on the top-1 fine-tuning accuracy of pre-trained ViT-B under the VQ-KD setting.

ance of MaPeT across all considered datasets. Our model outperforms both BEiT and CAE, highlighting its robustness and efficacy in capturing meaningful visual representations. These findings not only underline the potential of MaPeT as a powerful pre-training technique but also emphasize its cross-domain transfer learning capabilities, which enable practical relevance in various real-world applications.

4.3.6 Reconstruction Ratio Analysis

Here we discuss the relationship between the reconstruction ratio employed in MaPeT and its impact on image classification results. The reconstruction ratio indicates the proportion of target patches in relation to the entire input sequence. An advantage of MaPeT is that it attends to the target patch $\boldsymbol{x}_{\boldsymbol{z}_t} \in \boldsymbol{x}_{z_{>c}}$ during the prediction of $\boldsymbol{x}_{\boldsymbol{z}_{t+1}}$, allowing for the cutting point c to be positioned at every value in the interval $c \in [1 \dots N-1]$. This unique feature of MaPeT enables the potential reconstruction of all patches in an incremental and randomized manner, from the first to the last patch. On the other hand, models like BEiT and CAE lack the ability to reconstruct all image patches because this would lead to a masking ratio that is too high, resulting in the loss of visual context necessary for the reconstruction of the masked elements.

Table 4.9 refers to the analysis of the reconstruction ratio on the ViT-B architecture. It can be noted that MaPeT exhibits a preference for a reconstruction ratio of approximately 70% of the complete sequence. Indeed, the initial target patches are compelled to establish correlations with a small portion of visible patches (namely, 30% of the entire sequence) randomly distributed throughout the image. The model can acquire long-range spatial dependencies that are contingent on the position of the target patch in the permuted order. As the final patches attend to the

	MLP		*ViT-T*	
Tokenizer	**Top-1 Acc. (%)**	**Top-5 Acc. (%)**	**Top-1 Acc. (%)**	**Top-5 Acc. (%)**
DALL-E	4.1	11.7	9.1	21.9
VQ-KD	68.1	89.8	72.3	92.2
k-CLIP	**72.8**	**93.2**	**76.2**	**94.9**

Table 4.10: Accuracy of image classification when employing visual tokens as model input. Note that using CLIP features for zero-shot image classification lead to a top-1 accuracy of 73.6%.

majority of the visual content, they are likely to concentrate more on neighboring image patches, consistent with the principle of spatial locality. A reduction in the reconstruction ratio leads to an increase in the number of visible patches attended by the target patches that occur early in the permuted order. This results in an increase in the likelihood of having neighboring visible patches, which, in turn, diminishes the learned long-range spatial dependencies. In contrast, an excessively high reconstruction ratio of 85% makes the pre-training objective of MaPeT excessively difficult, as early predictions are likely to be arbitrary, having access to only 15% of the overall visual information. Consequently, excessively high or low reconstruction ratios impair the performance, as evidenced by Table 4.9.

4.3.7 Visual Tokenizer Analysis

In this section, we present a comprehensive qualitative and quantitative analysis of our proposed k-CLIP and compare its performance to the existing DALL-E [194] and VQ-KD [189] tokenizers.

Impact of Tokenizer on Model Performance. Analyzing the results in Table 4.4, it can be noted that k-CLIP outperforms VQ-KD across all models evaluated on ViT-T, with improvements in top-1 accuracy of 1.0%, 0.4%, and 0.3% observed on MaPeT, PIM, and MIM, respectively. However, the results differ when evaluated on ViT-S and ViT-B, where VQ-KD exhibits better classification performance. We argue that the simpler semantic features of k-CLIP, with their inherent correlation with the semantic density of the image, can serve as a more effective self-supervisory signal for smaller models. On the other hand, the VQ-KD codebook is trained specifically to reconstruct CLIP features, making it a more demanding pre-training signal and thus more suitable for larger models.

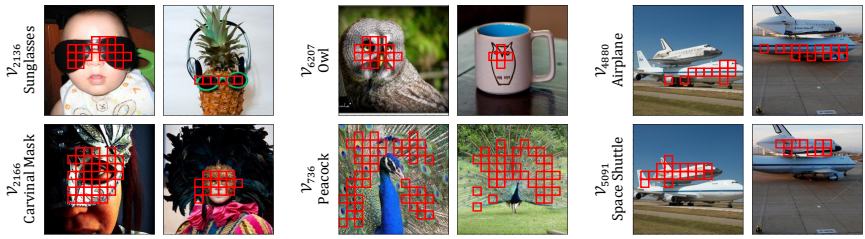


Figure 4.3: Visualization of image patches corresponding to the discrete tokens contained within our k-CLIP codebook. The codebook exhibits a high degree of semantic density and coherency, *i.e.*, semantically similar image patches are consistently linked with the same discrete token in the visual codebook. Corresponding image patches are marked in red rectangle.

Image Classification with Discrete Visual Tokens. In this section, we analyze image classification performance when using discretized visual tokens directly as input to the model. We conduct this analysis using the previously employed tokenizers (*i.e.*, our k-CLIP and VQ-KD [189]) and also include the DALL-E visual tokenizer [194], which consists of a discrete variational autoencoder model. Specifically, we map the sequence of visual tokens to an embedding space of dimension D through learnable embeddings trained in conjunction with two different model backbones: a lightweight MLP classification head and a ViT-Tiny model. The MLP head comprises an embedding layer, a linear layer with output dimension $D = 192$, a ReLU activation, an average pooling operation, and an additional linear layer that projects the pooled features to the number of classes (*i.e.*, 1000).

The results shown in Table 4.10 indicate that the use of DALL-E visual tokens yields limited accuracy in this setting. This can be attributed to the relatively low amount of semantic information derived from the reconstruction task that DALL-E is trained on. In contrast, k-CLIP demonstrates significant performance superiority over alternative tokenizers when employed with both the MLP head and ViT-Tiny backbone. This observed trend can be attributed to the rich semantic information inherent in CLIP visual features, which is effectively preserved through the process of k-means discretization. Consequently, a strong correlation between visual tokens and image classes can be established, leading to favorable performance outcomes even when utilizing small models such as the MLP head.

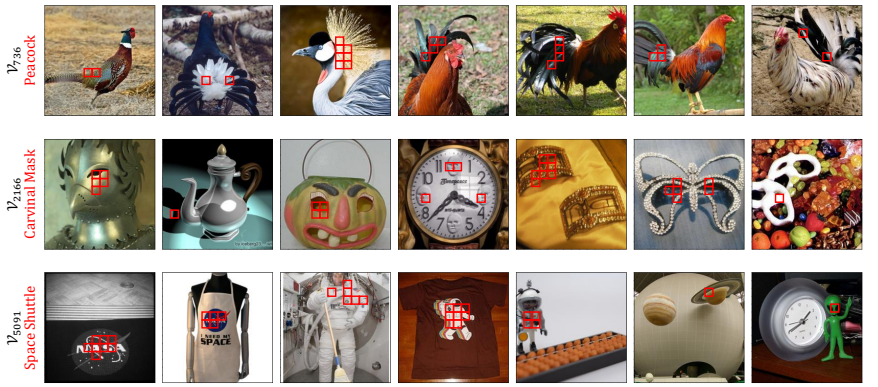


Figure 4.4: Visualization of image patches mismatching the semantic concept associated with the discrete tokens contained within our k-CLIP codebook. Corresponding image patches are marked in red rectangle.

Codebook Visualization. Figure 4.3 presents a collection of examples that showcase the semantic associations between image patches and visual tokens, extracted using our k-CLIP tokenizer. This visualization effectively demonstrates the efficacy of our tokenizer in accurately capturing and representing the semantic content of the images within the considered dataset and shows its ability to recognize and group image patches that share common visual features and semantic meaning. As it can be noticed, the visual tokens shown are congruent with specific semantic concepts, albeit resulting in distinct representations for similar visual elements such as `sunglasses` and `carnival mask`, `owl` and `peacock`, or `airplane` and `space shuttle`. Furthermore, it is worth noting that the visual tokens remain resilient to variations in color, style, rotation, and size. This is exemplified by the middle example, where the image depicts an owl print on a cup, yet it is still accurately identified using the `owl` visual token. These observations emphasize the ability of our k-CLIP tokenizer to effectively identify and classify complex visual concepts. An additional collection of codebook visualization examples is presented in Figure 4.5. Notably, the pairs of `Clownfish` and `Lionfish`, `Leopard snout` and `Cougar snout`, `Yellow butterfly` and `Yellow flower`, as well as `Bicycle` and `Unicycle`, further validate the discriminative nature of visual tokens, even for highly similar categories.

In Figure 4.4, we also show some examples of image patches that mismatch

the semantic concepts associated with the discrete tokens of the codebook. Indeed, we observe some incongruities that may be linked to the resemblance of patches to other semantic concepts. For instance, in the first row of the figure, some image patches are identified with the `Peacock` visual tokens while the bird species are different. The sumptuous tails of these birds can be visually associated with the characteristic features of a peacock. Similarly, in the second row of the image, the silver and gold perforations, which are associated with medieval headgear or adornments, may be perceived as comparable to the surface of a carnival mask. Furthermore, within the third row of the figure, it is worth noting that several images do not precisely match the semantically correct concept of `Space Shuttle`. Nevertheless, these images demonstrate a semantic affinity with the broader, higher-level concept of "Space". The images contain a variety of visual elements, including the NASA logo, astronauts, planets, and aliens. While the specific content of these images may diverge from the targeted concept, the overall semantic theme they evoke can be seen as related to the broader concept of space exploration and travel. These observations highlight our k-CLIP tokenizer enables the identification and classification of complex visual concepts.

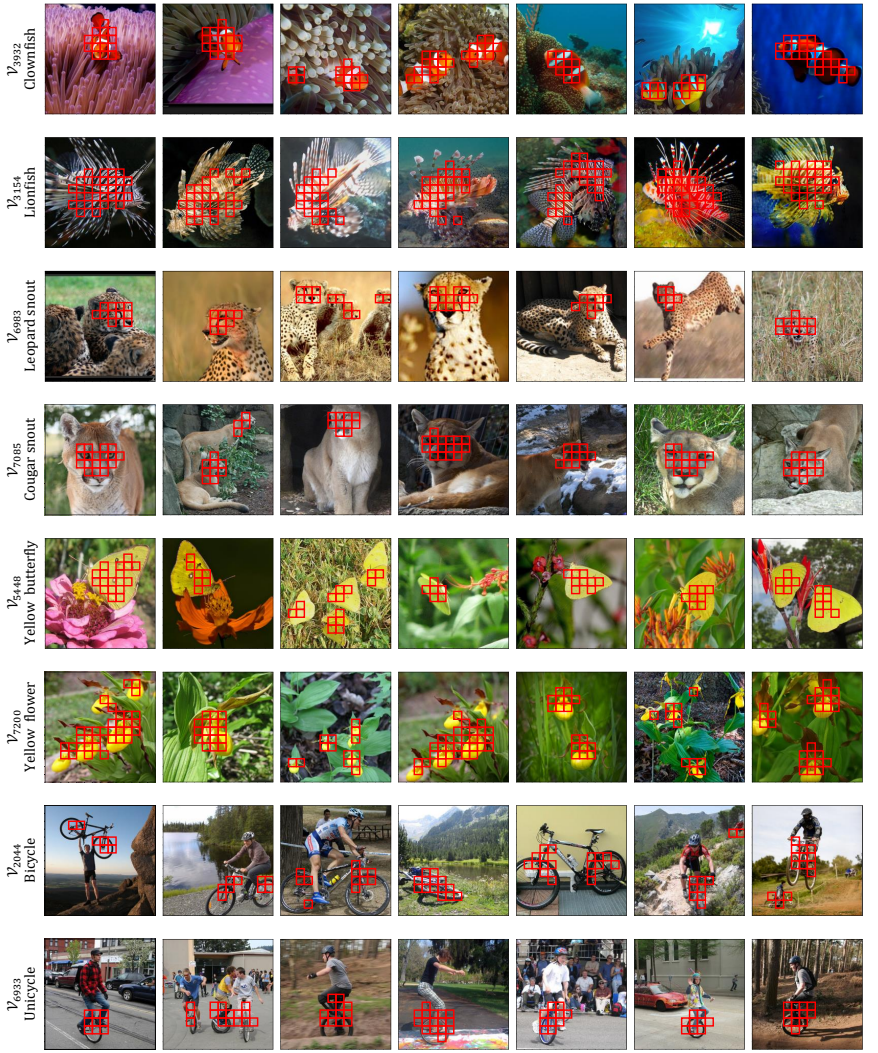


Figure 4.5: Additional examples of image patches corresponding to the discrete tokens contained within our k-CLIP codebook.

Chapter 5

Superpixels for Attentive Segmentation Models

While Vision Transformers (ViTs) have demonstrated remarkable success in tasks such as image classification and feature representation learning, their application to dense prediction tasks, such as semantic segmentation, poses unique challenges. Semantic segmentation is a core problem in Computer Vision, which aims at partitioning an image into coherent regions according to a set of semantic categories [163, 182]. This task requires precise delineation of object boundaries and an understanding of spatial relationships, making it particularly dependent on effective positional encoding and spatial reasoning mechanisms.

This chapter builds on the foundational principles established earlier, focusing on how ViTs can be further optimized for segmentation tasks by integrating perceptual priors from classical perception-based Computer Vision techniques. The rationale behind this idea is that low-level assumptions and knowledge-based models, such as a perceptual grouping based on appearance similarity, can potentially be useful for improving the accuracy of semantic segmentation models.

Semantic segmentation has emerged as a critical application domain for ViTs [299], with recent models such as DPT [195], SegFormer [255], and SETR [299], that have successfully integrated attentive architectures for dense prediction. As

This chapter is related to the publication "Roberto Amoroso, Matteo Tomei, Lorenzo Baraldi, and Rita Cucchiara, Superpixel Positional Encoding to Improve ViT- based Semantic Segmentation Models, BMVC 2023".

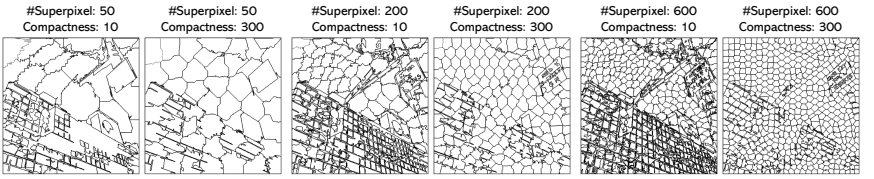


Figure 5.1: Example of superpixel maps with varying resolutions and compactness.

we observed in the previous chapters, Transformer-based architectures share the common characteristic of employing a positional encoding strategy to encode the relative or absolute position of words or tokens in a sequence. This is particularly important in the context of Natural language Processing (NLP), where the order of words in a sentence profoundly impacts its meaning. Similarly, in Vision Transformers, positional encoding is used to encode the relative position of patches in an image, facilitating the understanding of spatial relationships among these objects and enabling more accurate predictions.

Although there are several works investigating positional encoding strategies for classification [67, 110, 186, 195, 250, 255], the literature on positional encoding techniques for dense prediction tasks, such as segmentation, is scarce. Moreover, existing positional encoding strategies share a common limitation, as they solely provide information about the position of input patches, while lacking semantic priors that offer insights into the shape and edges between different objects, *i.e.*, semantic classes, within the input image.

Contributions. To address this limitation, we discuss if and when a grouping strategy based on appearance similarity, such as the one of superpixels [199], can be useful for semantic segmentation when employing self-attentive-based architectures. Superpixels cluster adjacent and perceptually similar pixels in uniform image regions by following edges and color variations (see some examples in Figure 5.1). As such, they have often been adopted in segmentation before the appearance of CNN-based methods. Since the seminal work by Ren and Malik [199], tens of approaches have been proposed to create superpixels according to different optimization functions [1, 157], and also Convolutional Networks have been adopted for creating superpixels [267]. Regardless of the generation approach, superpixels can be considered seed areas for segmentation tasks, as they provide a precise indication of where edges lie and have a perceptual meaning. In Transformer-based architectures, therefore, superpixels may be useful to recover precise boundaries between classes. Building upon these insights, we devise a

novel superpixel-based positional encoding (PE) strategy specifically designed for semantic segmentation. Our strategy injects superpixel shape and position priors into the ViT encoder features, creating more boundary-aware semantic latent space representations. In our evaluation, we also investigate an ablation of several positional encoding strategies to encode and embed superpixel information in the encoder features.

The rest of the chapter is organized as follows: Section 5.1 introduces our superpixel-based positional encoding method, elaborating on two distinct strategies for encoding positional information. Section 5.2 outlines the experimental setup, including datasets, superpixel algorithms, and backbone configurations, and analyzes the results by comparing different positional encoding strategies. Additionally, we evaluate runtime costs and perform ablation studies to examine the impact of superpixel algorithms and hyperparameter variations in the superpixel extraction process.

5.1 Proposed Method

For dense prediction tasks, Vision Transformers are commonly designed following an encoder-decoder architecture [195, 255, 299], where features coming from different encoder layers are used to predict the segmentation map. In this chapter, we propose a superpixel-based positional encoding that injects shape and position priors into the ViT encoder features, creating more boundary-aware semantic latent space representations. The proposed approaches are *model-agnostic*, *parameters-free* and *plug-and-play*; they do not involve any architectural changes to the encoder or decoder and only require the extraction of a superpixel map over the input image through an algorithm that we call $\mathcal{S}$.

More in detail, given the superpixel algorithm $\mathcal{S}$ and an input image I with shape $H \times W \times C$, we apply $\mathcal{S}$ to I to obtain a superpixel map $\mathcal{L}$ with shape $H \times W$, and a set of centroids $\mathcal{C}$ with shape $N_s \times 2$, where N_s represents the number of superpixels extracted by $\mathcal{S}$. $\mathcal{L}$ contains predicted superpixels, *i.e.*, each pixel in $\mathcal{L}$ is represented by an integer in the range $[0, N_s - 1]$ specifying which of the N_s superpixels it belongs to, while $\mathcal{C}$ contains the (x, y) coordinates of the centroid of each superpixel, relative to the input image.

The generation of superpixels is influenced by an important hyperparameter, namely compactness. Varying the compactness magnitude affects shape, size, and boundary smoothness. Increasing compactness results in smoother superpixels, while decreasing compactness leads to irregular shapes, similar to overfitting.

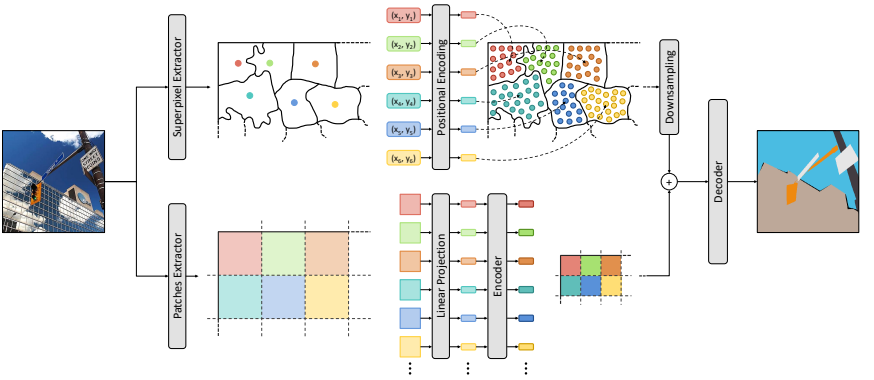


Figure 5.2: Our superpixel-based positional encoding strategy. We extract a positional encoding that can be *absolute*, when exploiting a sinusoidal encoding of the superpixels centroids, or *relative*, when leveraging the progressive index associated with each superpixel. The obtained superpixel position encoding vectors are replicated in each pixel of their corresponding superpixel (top). The resulting map is downsampled and added to the reshaped patch features (bottom).

Higher compactness captures spatial information better and aids information extraction. However, excessive values lead to grid-like superpixel maps, as qualitatively reported in Figure 5.1.

5.1.1 Superpixel-based Positional Encoding

The literature on ViT-based architectures [67, 250] highlights the importance of incorporating a positional encoding into the input embedding to achieve superior model performance. Traditionally, positional encoding conveys information about the *absolute* [67, 195] or *relative* [67, 255] position of the input patches. Recent approaches also incorporate positional encoding into intermediate layers of the architecture [110, 186], especially when the features extracted by the ViT-based encoder are downscaled in some way, such as through the pooling strategy discussed in Chapter 3. This underscores the significance of injecting position information not only at the input but also in the intermediate processing of the model.

Our superpixel-based positional encoding (PE) not only provides information about the position of input patches but also offers insights into the shape and edges between different semantic classes within the input image. The proposed method

is built upon the map of superpixels $\mathcal{L}$ and their centroids $\mathcal{C}$, as outlined in the following:

❶ For a given input image I, we extract a map of superpixels $\mathcal{L}$ along with their centroids $\mathcal{C}$ (colored dots in Figure 5.2–top-left). ❷ For each superpixel $\mathcal{L}_i$, with $i \in [0, N_s - 1]$, we compute the encoding of its position PE_i (colored rectangles in Figure 5.2–top-center) with the same shape d_{model} as the latent vector size of the Vision Transformer. By sharing the same latent shape, our Superpixel-based PE can be added to the features extracted from the ViT encoder. The superpixel position encoding PE_i can be either *absolute*, when employing the coordinates (x, y) of the corresponding centroid $\mathcal{C}_i$, or *relative*, when using the progressive index i associated with the superpixel $\mathcal{L}_i$, as discussed below. ❸ Given the i-th superpixel $\mathcal{L}_i$, we replicate its positional encoding PE_i over every pixel belonging to $\mathcal{L}_i$. In Figure 5.2–top-right, the colored small circles represent the positional encoding PE replicated over the surface of the corresponding superpixel. Following this approach, we obtain a superpixel-based positional encoding map $PE_{\mathcal{L}}$ with shape $H \times W \times d_{model}$. ❹ At the same time, we also extract N_p squared regular patches (dotted grid in Figure 5.2–bottom-left) that we feed as input to the ViT encoder. The j^{th} layer of the encoder outputs a feature vector f^j (colored rectangles in Figure 5.2–bottom-center) with shape $N_p \times d_{model}$, after removing the *cls_token* if adopted. ❺ Finally, we downsample $PE_{\mathcal{L}}$ and sum to the encoder features map f^j, before feeding the result to the downstream segmentation decoder (Figure 5.2–bottom). These operations are executed both at training and inference time.

In other words, the pixels in $\mathcal{L}$ that belong to a specific superpixel are replaced with the positional encoding PE of that superpixel. This results in a full-resolution map that encodes both the shape and position of the semantically homogeneous regions (*i.e.*, superpixels) computed by $\mathcal{S}$. Since the segmentation decoder usually expects features with spatial shape $\sqrt{N_p} \times \sqrt{N_p} \times d_{model}$ as input, obtained by reshaping f^j, we simply downsample the spatial resolution of $PE_{\mathcal{L}}$ in order to match $\sqrt{N_p} \times \sqrt{N_p}$. This downsampling operation does not affect the other components of the decoder, ensuring that they remain unmodified.

5.1.2 Absolute and Relative Position Encoding Strategies

Although superpixel maps provide priors on shape and edges, the position encoding strategy remains to be defined. As a solution, we propose two approaches to inject superpixel position information: an *absolute* positional encoding that exploits the information of the centroids $\mathcal{C}$ and a *relative* positional encoding that leverages

the progressive index associated with each superpixel of $\mathcal{L}$.

Inspired by [230], our *absolute* superpixel positional encoding adopts sine and cosine functions of different frequencies to encode x and y coordinates of the superpixel centroids:

$$\begin{aligned} SinPE^{c}_{(sup,2i)} &= \sin(c_{(sup)}/10000^{2i/d_{pe}}), \\ SinPE^{c}_{(sup,2i+1)} &= \cos(c_{(sup)}/10000^{2i/d_{pe}}), \end{aligned} \tag{5.1}$$

where c represents the x or y coordinate, and d_{pe} is equal to $d_{model}/2$. Here sup represents the index of the superpixel (from 0 to $N_s - 1$), and i covers the d_{pe} dimensions. Afterward, $SinPE^{y}$ and $SinPE^{x}$ are concatenated along the channel dimension:

$$SinPE = (SinPE^{x} \parallel SinPE^{y}) \tag{5.2}$$

with $SinPE$ having shape $N_s \times d_{model}$. After computing $SinPE$ following Equation 5.2, we obtain a $N_s \times d_{model}$ encoding of the (x, y) centroids coordinates $\mathcal{C}$. We now have a sinusoidal positional encoding for each superpixel extracted from the input image. Then, following the same insight as in ❸, we replicate the *absolute* sinusoidal positional encoding of the i-th centroid $SinPE_i$ over the surface of the i-th superpixel. Finally, we obtain $SinPE_{\mathcal{L}}$, with shape $H \times W \times d_{model}$, from $\mathcal{L}$ by replacing the pixels belonging to a specific superpixel with the sinusoidal positional encoding of its centroid.

We also propose a *relative* superpixel positional encoding strategy which exploits the progressive index sup assigned to each superpixel by the algorithm $\mathcal{S}$. This index is an integer in the range $[0, N_s - 1]$ that follows the row-major order, *i.e.*, index 0 corresponds to the first superpixel in the top-left corner and index $(N_s - 1)$ corresponds to the final superpixel in the bottom-right corner. We compute a linear *relative* positional encoding by normalizing the superpixel indices in the interval $[0, 1]$:

$$LinearPE_{(sup)} = \frac{sup}{N_s - 1} \tag{5.3}$$

with $LinearPE$ having shape N_s. We replicate the linear encoding of the superpixel index over the channel size to match the latent space size of the encoder, resulting in a $LinearPE$ with shape $N_s \times d_{model}$. As done for the sinusoidal encoding, we follow the procedure described in ❸ and replicate over the surface of the i-th superpixel the relative linear positional encoding $LinearPE_i$ of its

index, having shape d_{model}. In other words, we obtain $LinearPE_{\mathcal{L}}$, with shape $H \times W \times d_{model}$, from $\mathcal{L}$ by replacing the pixels belonging to a specific superpixel with the linear positional encoding of its index. For both of the proposed approaches, $PE_{\mathcal{L}}$ is spatially downsampled to match the shape of f^j , and added to it, as depicted in Figure 5.2.

In our experiments, we compare the proposed superpixel-based PE with several alternative strategies for encoding position and incorporating superpixel shape and position priors into ViT encoder features, as outlined in the following section.

5.2 Experiments

This section introduces the datasets utilized in our study, followed by an explanation of the superpixel extraction algorithm and the semantic segmentation architectures considered. We then present a comparative analysis of our proposed superpixel positional encoding strategies against alternative position encoding methods. Additionally, we assess runtime performance and conduct ablation studies to investigate the influence of different superpixel algorithms and extraction hyperparameters. Throughout our experiments, semantic segmentation performance is reported in terms of mean Intersection over Union (mIoU).

5.2.1 Experimental Setup

Datasets. We evaluate our proposed superpixel-based techniques over two publicly available datasets: ADE20K [302] and Cityscapes [48]. ADE20K is a challenging scene-parsing dataset with 150 fine-grained semantic concepts. It comprises a training set of 20210 images, a validation set of 2000 images, and a testing set of 3352 images. On the other hand, Cityscapes is a driving dataset for semantic segmentation consisting of 5000 fine-annotated high-resolution images, split into 2975, 500, and 1525 images for training, validation, and testing respectively. It densely annotates 19 object categories in images with urban scenes.

Superpixel Algorithm. As our superpixel extraction algorithm, we adopt an optimized variant of SLIC [1], named FastSLIC[1]. To enhance efficiency, we integrate it into the data loading process. FastSLIC is designed with various optimization techniques, such as color quantization, integer-only arithmetic, row subsampling, and multicore parallelization. In Section 5.2.3, we report an in-depth

[1] `https://github.com/Algy/fast-slic`

analysis of the impact of superpixel extraction on data loading, training, and inference, and in Section 5.2.4 explore alternative superpixel algorithms.

DPT. Dense Prediction Transformer (DPT) consists of an encoder-decoder architecture that employs Vision Transformers for semantic segmentation. The encoder uses a stack of Transformer blocks to extract feature maps at multiple stages. The decoder reconstructs image-like feature representations and progressively fuses them into the final dense prediction through residual convolutional units. Our experiments incorporate the proposed superpixel-PE by summing it to the feature representations extracted by the encoder prior to the fusion operation applied by the decoder. We consider the variant of DPT employing a ViT-Base backbone. DPT uses random horizontal flipping and random re-scaling, with a batch size of 16 and square random crops of size 480 and 512 extracted from ADE20K and Cityscapes images. We use a cross-entropy loss, AdamW optimizer, and cosine learning rate scheduler, with a learning rate set to $(0.002 \cdot bs/512)$.

SegFormer. The SegFormer model is based on a hierarchical Transformer encoder to extract four feature maps (c_1, c_2, c_3, and c_4) at different resolutions, ranging from the high-resolution fine-grained features of c_1 to the low-resolution coarse-grained features of c_4. An MLP-based decoder then fuses these features and predicts the dense segmentation mask. We downsample and sum the proposed superpixel-based PE to the encoder outputs and feed the result to the MLP decoder. SegFormer provides encoders of different sizes, from B0 to B5, and we measure the gain of our proposed approach on the B0 and B4 encoders. SegFormer uses random resizing, horizontal flipping, and cropping to 512. It employs an AdamW optimizer for 160K iterations with a batch size of 16 for B0 and 8 for B4 on ADE20K, and a batch size of 8 on Cityscapes for both B0 and B4. The learning rate is set to $6\mathrm{e}{-5}$.

SETR. We use the multi-level feature aggregation (MLA) variant of SETR, which produces the best results on the segmentation task [299]. SETR-MLA (SETR to shorten) consists of an encoder-decoder ViT-based segmentation architecture. The encoder consists of a series of Transformer blocks that extract visual features at different layers. The encoder reshapes, upsamples, and fuses the visual features through a series of convolutional blocks to generate the final dense prediction. We incorporate our superpixel-PE by summing it to the reshaped visual features before these are fused by the last stack of convolutional layers. We consider two variants of SETR employing a ViT-Tiny and ViT-Small backbone respectively. SETR uses random resizing, horizontal flipping, and cropping to 512. It employs an SGD optimizer for 160K iterations with a batch size of 4 on ADE20K, and for

	#Superpixel	Compact.	Params (M)	ADE20K	Cityscapes
DPT-B [195]	-	-	102.0	44.9	71.0
DPT-B+SinPE$_{\mathcal{L}}$	16,000	20	102.0	45.4	71.7
DPT-B+LinearPE$_{\mathcal{L}}$	28,000	10	102.0	**45.8**	**72.0**
SegFormer-B0 [255]	-	-	3.8	37.5	71.4
SegFormer-B0+SinPE$_{\mathcal{L}}$	16,000	20	3.8	38.2	71.8
SegFormer-B0+LinearPE$_{\mathcal{L}}$	28,000	10	3.8	**38.4**	**72.2**
SegFormer-B4 [255]	-	-	64.1	49.0	78.4
SegFormer-B4+SinPE$_{\mathcal{L}}$	16,000	20	64.1	**49.3**	**78.6**
SegFormer-B4+LinearPE$_{\mathcal{L}}$	28,000	20	64.1	**49.3**	**78.6**
SETR-T [299]	-	-	10.2	35.2	69.3
SETR-T+SinPE$_{\mathcal{L}}$	8,192	10	10.2	**36.3**	**70.1**
SETR-T+LinearPE$_{\mathcal{L}}$	16,384	10	10.2	36.1	70.0
SETR-S [299]	-	-	26.7	42.7	74.6
SETR-S+SinPE$_{\mathcal{L}}$	8,192	10	26.7	43.0	74.9
SETR-S+LinearPE$_{\mathcal{L}}$	8,192	10	26.7	**43.4**	**75.0**

Table 5.1: Results of mIoU on ADE20K and Cityscapes using DPT-Base, SegFormer-B0, SegFormer-B4, SETR-Tiny and SETR-Small with/without superpixel positional encoding.

80K iterations with a batch size of 8 on Cityscapes, for both the Tiny and Small variants. The learning rate is set to $1e{-}3$ and $2e{-}3$ for ADE20K and Cityscapes respectively, with a momentum of 0.9.

5.2.2 Experimental Results

Superpixel-based PE. In this section, we investigate if the injection of shape and edge priors provided by our proposed superpixel-based positional encoding may improve the segmentation performance of existing Vision Transformer architectures. Table 5.1 presents mean IoU results for DPT with ViT-Base backbone, SegFormer with B0 and B4 backbones, and SETR with ViT-Tiny and ViT-Small backbones, with and without our absolute sinusoidal and relative linear superpixel-based positional encoding on ADE20K and Cityscapes datasets. For ADE20K, employing our sinusoidal and linear positional encoding improves mIoU from 44.9 to 45.4 and 45.8, respectively, for DPT-B, from 37.5 to 38.4 and from 49.0 to 49.3, respectively, for SegFormer-B0 and SegFormer-B4, from 35.2 to 36.3 and from 42.7 to 43.4, respectively, for SETR-T and SETR-S. For Cityscapes, mean

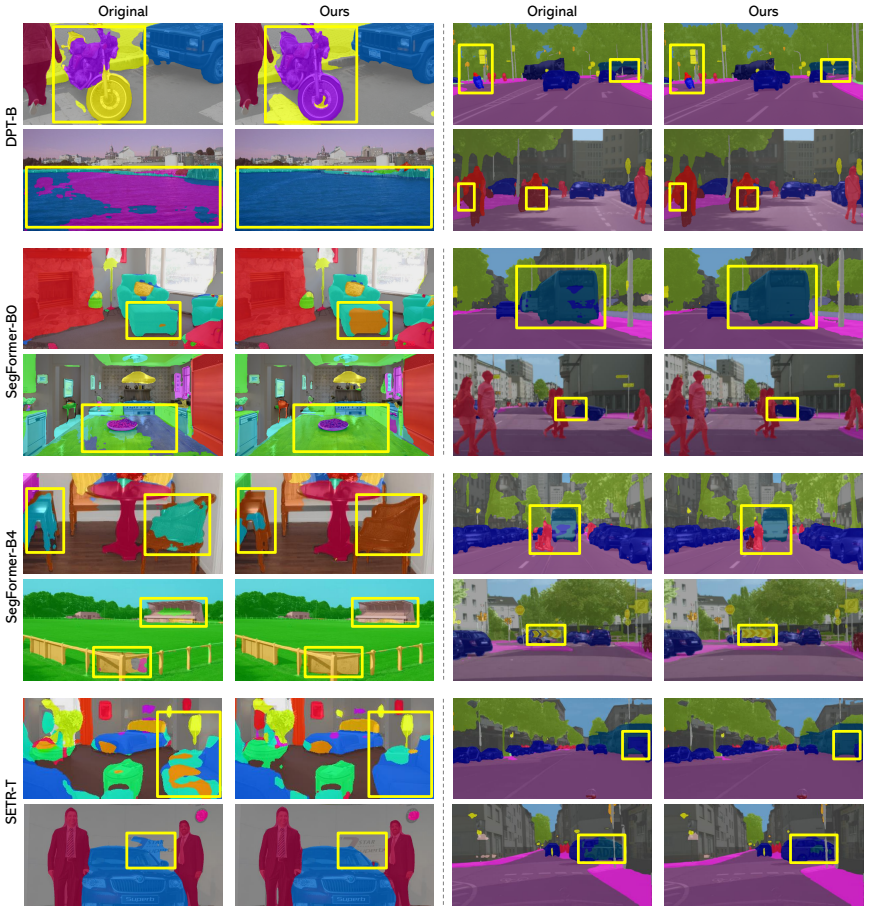


Figure 5.3: Sample results obtained employing DPT-B, SegFormer-B0, SegFormer-B4, and SETR-T, with and without our superpixel positional encoding strategy. Enhanced semantic segmentation outcomes are denoted by yellow boxes.

IoU improves from 71.0 to 71.7 and 72.0 for DPT-B with sinusoidal and linear encoding, respectively, from 71.4 to 72.2 and from 78.4 to 78.6 for SegFormer-B0 and SegFormer-B4, from 69.3 to 70.1 and from 74.6 to 75.0 for SETR-T and SETR-S. We observe that a large number of superpixels is necessary to ensure

	#Superpixel	Compact.	Params (M)	ADE20K	Cityscapes
SegFormer-B0 [255]	-	-	3.8	37.5	71.4
+PixelPE	-	-	3.8	37.8	71.5
+PatchPE-1	-	-	3.8	37.6	71.7
+PatchPE-4	-	-	3.8	37.5	71.7
+PatchPE-16	-	-	3.8	37.7	71.7
+WeightedPE$_{\mathcal{L}}$	16,000	20	3.8	37.7	**72.4**
+LearnablePE$_{\mathcal{L}}$	4,096	10	4.8	38.3	**72.2**
+SinPE$_{\mathcal{L}}$	16,000	20	3.8	38.2	71.8
+LinearPE$_{\mathcal{L}}$	28,000	10	3.8	**38.4**	**72.2**

Table 5.2: Comparison of our sinusoidal and linear superpixel-PE with different positional encoding strategies. Performance measured in terms of mIoU on the ADE20K and Cityscapes datasets, employing the SegFormer-B0 [255] model.

the fine-grained property for $\mathcal{L}$ and Sin/Linear-PE$_{\mathcal{L}}$ consequently. The obtained results show that our positional encoding strategies are effective in improving segmentation performance, as also qualitatively shown in Figure 5.3, which illustrates results obtained from ADE20K (first two columns) and Cityscapes (last two columns) for the DPT-B, SegFormer-B0, SegFormer-B4, and SETR-T models.

Pixel-based PE. We propose a method to quantify how much of the performance boost achieved by our superpixel-PE can be ascribed to positional encoding and how much to the injection of shape and edge priors provided by the superpixels. The construction method is fairly simple, consisting of a readaptation of the sinusoidal positional encoding provided in Section 5.1.2, but this time applied to the (x, y) coordinates of each individual pixel rather than the superpixel centroids. The resulting positional encoding has the same resolution as the original image and embeds a position encoding having the finest obtainable grain, the single pixel. The generated *Pixel-PE* is downscaled and summed with the encoder features. As can be seen from the 2nd row of Table 5.2, embedding the position of individual pixels yields a minor performance gain, which is far less than the increase in mIoU reported by our superpixel-PE. These results demonstrate that superpixel shape and edge priors are critical for improving segmentation performance.

Patch-based PE. A ViT-based encoder generates features from square patches extracted from the input image. Building on this, we explore a positional encoding

strategy that leverages the shape of these input patches rather than relying on superpixels. Specifically, we propose a method akin to the linear positional encoding (*LinearPE*) described in Section 5.1.2, but instead of encoding the progressive index of superpixels, we encode the numeric index that identifies each patch in sequence. The number of patches extracted is inversely proportional to the patch size: smaller patches result in more patches and, consequently, a higher resolution for the positional encoding mask. In the extreme case where the patch size equals 1, this approach effectively becomes a pixel-wise positional encoding. However, unlike the sinusoidal encoding of pixel coordinates used in Pixel-based PE, our method employs a linear encoding of progressive indices. The obtained positional encoding map is then added to the encoder's feature representation. We denote this approach as *PatchPE-*ps, where ps refers to the patch size. As we can see in Table 5.2 (3rd to 5th rows), as the patch-size varies, performance remains rather stable for both datasets considered, even in the extreme case of patch-size 1, and with similar mIoU values as found with the *PixelPE*. This patch-based positional encoding method lacks shape and edge priors provided by superpixels, resulting in reduced performance compared to our superpixel-based PE.

Weighted Superpixel-PE. We introduce the *WeightedPE* method, which combines encoder features and superpixel-based PE through trainable parameters. This approach provides insights into the relationship between superpixel priors and the varying resolutions of encoder features. Unlike prior experiments that involved downscaling and summing the superpixel-based PE to match the progressively lower-resolution outputs (c_1, c_2, c_3, and c_4) of the SegFormer encoder, *WeightedPE* employs a weighted sum of encoder features and superpixel-based PE. The weights are learnable parameters, one for each feature map produced by the encoder, all initialized to 0.10. After training, the values of the learnable weights β_1, β_2, β_3, and β_4 were found to be 0.60, 0.51, 0.19, and 0.03 for ADE20K, and 0.40, 0.30, 0.12, and 0.01 for Cityscapes, respectively. These results indicate that superpixel-PE is particularly beneficial for training high-resolution feature maps (*i.e.*, c_1) that preserve edge details between semantic classes. The proposed *WeightedPE* approach improves performance on both datasets, as evidenced by the results in the 6th row of Table 5.2.

Learnable Superpixel-PE. The techniques discussed thus far can be seamlessly integrated into encoder-decoder models without modifying their architecture or introducing additional parameters, except for the four learnable weights in the previously introduced *WeightedPE* method. To further explore the potential of leveraging superpixel information, we propose a novel approach called *LearnPE*,

Model	#Superpixel	Params (M)	ADE20K	Cityscapes
SegFormer-B0 [255]	-	3.8 (+0.0)	37.5	71.4
+LearnPE$_{\mathcal{L}}$	200	3.8 (+0.0)	37.8	71.7
+LearnPE$_{\mathcal{L}}$	600	3.9 (+0.1)	38.1	71.9
+LearnPE$_{\mathcal{L}}$	4,096	4.8 (+1.0)	**38.3**	**72.2**
+LearnPE$_{\mathcal{L}}$	16,384	8.0 (+4.2)	38.0	71.7
+LearnPE$_{\mathcal{L}}$	20,000	8.9 (+5.1)	37.7	71.1
+LearnPE$_{\mathcal{L}}$	24,000	9.9 (+6.1)	37.6	71.1
+LearnPE$_{\mathcal{L}}$	32,000	12.0 (+8.2)	37.7	71.9

Table 5.3: Relationship between the number of superpixels, the model size, and the mIoU performance for our learnable superpixel-based positional encoding.

which introduces a learnable superpixel encoding mechanism. In this method, the information replicated across each superpixel's surface becomes learnable, enabling the model to adaptively determine the optimal encoding for each superpixel during training. By incorporating learnable weights for each superpixel, the model dynamically adjusts these encodings, potentially enhancing segmentation performance. Specifically, *LearnPE* defines a learnable vector l_i of shape d_{model} for each superpixel, which is then replicated across the superpixel's surface during training. The performance of *LearnPE*, measured in terms of mIoU, is comparable to that of SinPE and LinearPE methods, as evidenced in Table 5.2 (7th row). However, this approach results in an increased number of model parameters. Table 5.3 provides a more detailed analysis of the relationship between the number of superpixels, model size, and mIoU performance. As the number of superpixels increases, the model's parameter count grows significantly—exceeding three times that of the baseline model. Despite this increase in capacity, mIoU performance tends to decline slightly while still surpassing baseline levels. This observation highlights a potential trade-off between model complexity and performance when employing *LearnPE*. It is worth noting that while a larger model may possess more capacity, the heightened complexity may undermine the advantages of leveraging the priors provided by the superpixels, potentially leading to overfitting.

Impact of Superpixels on Semantic Classes. Our experimental results demonstrate that our proposed Superpixel PE approach leads to an improvement in the mIoU scores across both the Cityscapes and ADE20K datasets. In this section, we conduct a more fine-grained analysis of the impact of our methodology on each of the 19 classes in Cityscapes and the 150 classes in ADE20K. To understand

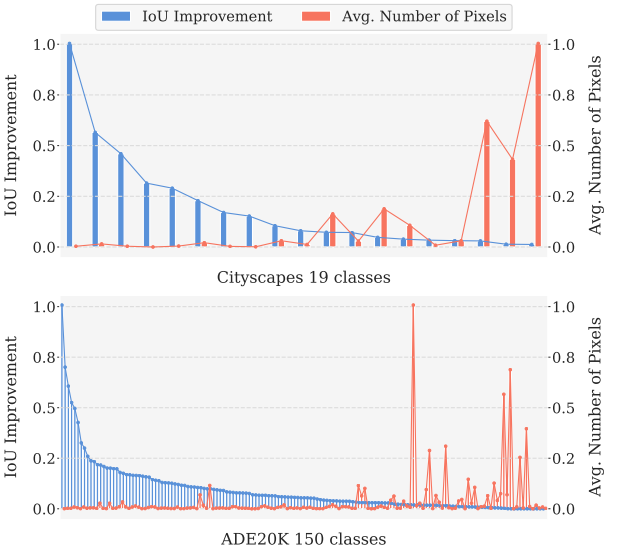


Figure 5.4: Relationship between per-class normalized IoU improvement and per-class normalized average pixels across Cityscapes and ADE20K datasets.

the relationship between IoU improvements and the average number of pixels per class, we present a comparative analysis in Figure 5.4. The findings reveal that the Superpixel PE method consistently enhances performance for classes with lower average pixel counts. For instance, in ADE20K, notable improvements are observed in rare classes such as `traffic light`, `bike`, `ottoman`, `flower`, and `sconce`. Similarly, in Cityscapes, classes like `traffic sign`, `rider`, `motorcycle`, `bus`, and `train` benefit significantly. These findings highlight how our approach is particularly effective in enhancing the segmentation performance for classes with low occurrence in the dataset, while also mitigating the risk of overfitting on classes with higher representation. This represents a significant advantage, as accurate segmentation of less frequent classes is often challenging due to limited data availability and class imbalance issues.

5.2.3 Runtime Analysis

In this Section, we conduct an analysis of the impact of our proposed superpixel-PE on inference time. As described in Section 5.1.1, our methodology requires

	Data train (ms)	Training (ms)	Data eval (ms)	Inference (ms)
ADE20K:				
DPT-B [195]	1.3	151.8	1.0	32.2
DPT-B+PE$_{\mathcal{L}}$	2.2	154.9	2.1	33.9
SegFormer-B4 [255]	1.9	260.4	1.1	62.4
SegFormer-B4+PE$_{\mathcal{L}}$	2.6	262.1	1.9	66.6
Cityscapes:				
DPT-B [195]	1.3	165.3	1.2	35.3
DPT-B+PE$_{\mathcal{L}}$	4.1	168.7	4.3	38.0
SegFormer-B4 [255]	1.8	244.0	1.1	63.0
SegFormer-B4+PE$_{\mathcal{L}}$	4.7	246.6	4.1	68.2

Table 5.4: Training time with an input resolution of 480×480 for ADE20K [302] and 512×512 for Cityscapes [48], both with and without our positional encoding method. We consider 16000 superpixels with a compactness of 20.

extracting superpixels from the input images, calculating a sinusoidal positional encoding based on superpixel centroids, and adding this PE map to the attentive features extracted by a ViT-based backbone. The first operation (superpixel extraction) impacts the data loading time, while the latter (calculating PE maps and summation) affects the forwarding time. To accomplish superpixel extraction, we adopted an optimized variant of SLIC [1], named FastSLIC[2]. For the data loading process, we fix the number of subprocesses for data loading to 8. All timings have been measured using an Nvidia(R) GeForce RTX 2080 Ti GPU and an Intel(R) Core(TM) i9-9820X CPU.

We should remark that for our approach, execution time is a more relevant metric than FLOPs because it accounts for the overhead incurred by the data loading process, which is a consequence of the superpixel algorithm. Consequently, the use of FLOPs as a metric would not accurately reflect this overhead, albeit limited. We measure the data loading and forwarding times of the DPT-Base [195] and SegFormer-B4 [255] semantic segmentation architectures, with a single image per batch. The measured times are averaged over the ADE20K [302] validation set, comprising 2000 images cropped at a resolution of 480×480, and the Cityscapes [48] validation set, which consists of 500 images cropped to 512×512. Table 5.4 provides insights into data loading, training, and inference times on

[2] https://github.com/Algy/fast-slic

Model	#Superpixel	Compact.	Data loading (ms)	Forward (ms)
DPT-B [195]	-	-	1.2	35.3
DPT-B+PE$_{\mathcal{L}}$	4000	1	2.4	37.8
DPT-B+PE$_{\mathcal{L}}$	4000	20	2.1	38.2
DPT-B+PE$_{\mathcal{L}}$	4000	100	2.1	38.5
DPT-B+PE$_{\mathcal{L}}$	8000	1	3.0	38.5
DPT-B+PE$_{\mathcal{L}}$	8000	20	2.8	38.1
DPT-B+PE$_{\mathcal{L}}$	8000	100	3.0	38.6
DPT-B+PE$_{\mathcal{L}}$	16000	1	4.5	38.3
DPT-B+PE$_{\mathcal{L}}$	16000	20	4.3	38.0
DPT-B+PE$_{\mathcal{L}}$	16000	100	4.1	38.2

Table 5.5: Inference time with 512×512 input resolution from Cityscapes [48], using our positional encoding method, for different values of superpixels number and compactness.

both ADE20K and Cityscapes. We measure the data loading time both during the training (*i.e.*, *Data train*) and inference (*i.e.*, *Data eval*). The superpixel extraction affects the data loading time considerably (about $2\times$), while the impact of superpixel-PE encoding calculation and summation is negligible and brings instead an increase of inference time under 8% (around 1 to 5 additional milliseconds per image). However, the data loading time is negligible compared to the inference time ($\sim$1 msec vs $\sim$63 msec), and training time ($\sim$2 msec vs $\sim$260 msec). Thus, our approach does not impose significant overhead on either training or inference.

Moreover, in Table 5.5 we report how data loading and forward times vary when changing the number of superpixels and the compactness value, given a fixed input resolution of 512×512. As expected, FastSLIC is slower as the number of superpixels increases, while its efficiency is not affected by compactness. Once again, the majority of the relative overhead is associated with the data loading process, while the forward time, which accounts for the majority of the overall time, is not significantly affected.

5.2.4 Ablation Studies

Evaluate Additional Superpixel Algorithms. We assess the robustness of our method over alternative superpixel algorithms, namely Watershed [176] and

	FastSLIC	**Watershed** [176]	**SEEDS** [228]
ADE20K:			
SegFormer-B0+SinPE$_{\mathcal{L}}$	38.2	**38.3**	38.1
SegFormer-B0+LinearPE$_{\mathcal{L}}$	**38.4**	**38.4**	**38.4**
Cityscapes:			
SegFormer-B0+SinPE$_{\mathcal{L}}$	71.8	**72.3**	72.2
SegFormer-B0+LinearPE$_{\mathcal{L}}$	**72.2**	72.1	**72.2**

Table 5.6: We investigate alternative superpixel algorithms, namely Watershed [176] and SEEDS [228], on ADE20k and Cityscapes when employing a SegFormer-B0 backbone. Performances measured in terms of mIoU.

Model	**#Superpixels**	**Compact.**	**Params (M)**	**mIoU**
DPT-S [195]	-	-	37.0	38.1
DPT-S+PE$_{\mathcal{L}}$ (after fusion)	600	100	37.0	38.2
DPT-S+PE$_{\mathcal{L}}$ (nearest)	600	100	37.0	38.4
DPT-S+PE$_{\mathcal{L}}$ (before fusion)	600	100	37.0	**38.9**

Table 5.7: We evaluate different strategies for integrating positional encoding with the DPT [195] decoder's fusion operation and investigate replacing bilinear downsampling with a nearest-neighbor alternative.

SEEDS [228]. Both algorithms produce accurate boundaries with potentially irregular shapes compared to SLIC [1, 217]. We use the same backbone architecture and superpixel number to ensure a fair comparison with our initial results. Our superpixel-PE consistently improves segmentation performance also when integrated with Watershed and SEEDS on both the ADE20k and Cityscapes datasets, demonstrating its robustness and adaptability, as shown in Table 5.6. Note that Watershed and SEEDS entail a significant computational overhead compared to FastSLIC ($\times 3$ and $\times 2$ slower, respectively). Therefore, FastSLIC remains a favorable choice due to its improved performance and efficiency.

Impact of Superpixel-PE Injection Point. In the DPT-based design, our positional encoding is resized through bilinear downsampling and added to the feature representations extracted by the encoder *before* the fusion operation applied by the decoder. Here we investigate the effects of varying the injection point of the superpixel-based positional encoding in encoder-decoder architectures, along with

Model	Compact.	Params (M)	ADE20K	Cityscapes
SegFormer-B0 [255]	-	3.8	37.5	71.4
SegFormer-B0+PE$_{\mathcal{L}}$	1	3.8	37.9	71.1
SegFormer-B0+PE$_{\mathcal{L}}$	10	3.8	38.1	71.5
SegFormer-B0+PE$_{\mathcal{L}}$	20	3.8	**38.2**	**71.8**
SegFormer-B0+PE$_{\mathcal{L}}$	30	3.8	38.0	71.2

Table 5.8: mIoU performance on ADE20K [302] and Cityscapes [48] using SegFormer-B0 [255] for different compactness values with a fixed superpixel count of 16,000.

the chosen downsampling strategy. The results are summarized in Table 5.7. For these experiments, we consider a DPT model based on ViT-S. When switching to the Small backbone, and considering an input resolution of 224×224, we observe that a smaller number of superpixels is necessary to generate a fine-grained segments map $\mathcal{L}$, and consequently a fine-grained $PE_{\mathcal{L}}$. In this scenario, we opt for 600 superpixels and a compactness of 100. In the second row of Table 5.7, we demonstrate that shifting the injection of the superpixel-based positional encoding after the fusion operation but before the last convolutional layer of the decoder, does not yield beneficial results. This observation can be explained by the fact that a single convolutional layer is insufficient to capture the spatial arrangement of the centroids encoding given by the superpixel shape. Furthermore, we also explore a nearest neighbor downsampling strategy as an alternative to bilinear interpolation for generating $PE_{\mathcal{L}}$ (to match the shape of each single f^i). However, we observe inferior performance with this approach, as indicated in the third row of Table 5.7.

Impact of Superpixel Compactness Variation. In Table 5.8, we apply our positional encoding to the SegFormer decoder [255] and examine how variation in the superpixels compactness value impacts model performance. In this experiment, we utilize the lightweight B0 version of SegFormer with an input resolution of 512×512 and a fixed number of 16,000 superpixels. The results demonstrate that the mIoU improves as the compactness value increases from 1 to 20. However, we observe a decline in performance when the compactness is further increased to a value of 30. This can be explained by the fact that excessively increasing the compactness of the superpixels around their centroid makes them similar to polygonal patches with regular edges, invalidating the priors of the boundaries between distinct semantic classes. Qualitative evidence of this phenomenon is depicted in Figure 5.1.

Chapter 6

Open-vocabulary Semantic Segmentation

In Chapter 5, we investigated the integration of Vision Transformers (ViTs) into dense prediction tasks, highlighting how perceptual priors, such as superpixels, can enhance semantic segmentation by improving boundary delineation. This exploration underscored the versatility of Vision Transformers in capturing both global context and fine-grained details, laying the groundwork for their broader application in complex vision tasks. Beyond their contributions to traditional Computer Vision problems, Transformers have emerged as a cornerstone for advancing multimodal learning. Thanks also to the progress explored in the previous chapters, Transformers represent a crucial component in bridging the gap between vision and language, enabling models to process and integrate information from multiple modalities effectively. This capability is particularly significant for tasks that require understanding complex relationships between visual content and semantic concepts.

Open-vocabulary semantic segmentation exemplifies the challenges and opportunities at the intersection of vision and language. Unlike traditional semantic segmentation tasks, which rely on pre-defined categories and extensive labeled datasets, open-vocabulary segmentation aims to generalize beyond seen categories

This chapter is related to the publication "Luca Barsellotti*, Roberto Amoroso*, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara, Training-Free Open-Vocabulary Segmentation with Offline Diffusion-Augmented Prototype Generation, CVPR 2024 (* Equal Contribution)".

Figure 6.1: Open-vocabulary segmentation with: (a) TCL [21], 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.

by leveraging vision-language models to align textual descriptions with visual regions [62, 82, 138, 158, 262, 293]. One of the major challenges in this setting is how to transfer the ability to match texts and images of large-scale vision-language models (*e.g.*, CLIP [191] and ALIGN [112]) to a text-pixel alignment. Given a large-scale set of web-crawled image-caption pairs, previous approaches [21, 167, 197, 258, 260, 304] force the ability to localize textual concepts to emerge through contrastive learning techniques combined with grounding mechanisms [21, 258, 260]. However, captions often capture the global scene and might present ambiguities with respect to fine-grained elements, making this approach sub-optimal and computationally intensive.

On a different note, advances in diffusion models [100, 201] have shown remarkable results in text-to-image generation, and recent works have shown that their features encompass knowledge regarding the positioning of the generated objects [150, 219, 251]. This information can be exploited to generate large sets of attribution maps, which are more active in the area corresponding to a semantic class, thus providing a valuable source of information for semantic segmentation.

Contributions. In this chapter, we propose to explore this mechanism as an alternative to multimodal contrastive training, in a fully training-free methodology where no parameter is learned. In contrast to previous works, our proposed approach follows an efficient two-step protocol: in an offline stage, we leverage a diffusion-augmented generation in which a collection of textual-visual reference vectors is generated. Then, at inference time, these references are retrieved to compute local and global similarities to segment the input image. In detail, we employ a large set of textual captions to generate synthetic images and corresponding attribution maps, through a localization mechanism based on cross-attention. Subsequently, we leverage a self-supervised visual backbone, DINOv2 [185], to build an offline set of visual prototypes associated with textual vectors, each representing the context of an instance in its synthetic scene. At inference time, we extract both global features with a multimodal encoder (*i.e.*, CLIP) and local dense features with DINOv2, characterized by high semantic relatedness, and employ a superpixel algorithm to detect class-agnostic regions in the input image. By querying the input textual category against the textual-visual reference embeddings, we then assign each superpixel to the category that exhibits the highest combined similarity, between the global and local modalities. As our approach is training-*free* and relies on *D*iffusion-*A*ugmented generation, we name it FreeDA.

We validate the proposed framework by conducting extensive experiments on multiple datasets, including Pascal VOC [72], Pascal Context [179], COCO Stuff [16] and Object [153], Cityscapes [48], and ADE20K [302, 303]. Remarkably, without requiring any form of training, FreeDA consistently outperforms previous approaches by a large margin, achieving state-of-the-art performance on all datasets, as qualitatively shown in Figure 6.1. Overall, our work demonstrates that non-parametric approaches can provide a compelling and efficient alternative for open-vocabulary semantic segmentation, and opens up new opportunities for subsequent works.

The remainder of the chapter is organized as follows. Section 6.1 presents our proposed open-vocabulary segmentation methodology, which comprises an offline diffusion-augmented prototype generation phase and a retrieval-based online inference pipeline that utilizes superpixels as training-free mask proposers to combine local and global similarities. Section 6.2 details the experimental setup, including the datasets employed, implementation specifics, and evaluation protocols. This section also reports our experimental results, featuring both quantitative and qualitative comparisons with state-of-the-art methods, and in-depth ablation studies.

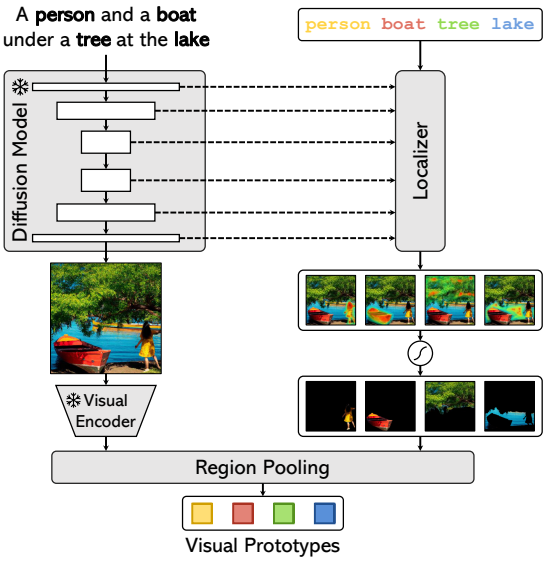


Figure 6.2: Overview of the diffusion-augmented prototype generation phase of FreeDA. Visual prototypes are generated by pooling self-supervised visual features on weak localization masks extracted from Stable Diffusion.

6.1 Training-Free Open-Vocabulary Segmentation with Diffusion-Augmented Prototypes

The goal of open-vocabulary segmentation is to segment an image according to an arbitrary set of categories represented through free-form texts. Our training-free approach decouples the task into two phases: a *diffusion-augmented prototype generation* phase, which is carried out in an off-line manner (visually represented in Figure 6.2), 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 6.3.

6.1.1 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.

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 generate a large set of real-world scenes using Stable Diffusion [201] 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 [219].

Diffusion models, indeed, map word embeddings of the conditioning text to the activations of their denoising subnetwork (*e.g.*, U-Net [201, 203]) 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 denoising 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

$$A(I, w) = \frac{1}{TLH} \sum_{t,l,h} \text{upsample}(\mathcal{A}(I, w)_{t,l,h}), \tag{6.1}$$

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 [201] and generate the corresponding set of synthetic images. Through a noun parser [164], 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 [185], 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:

$$p(I, w_i) = \frac{\sum_{h=0}^{\frac{H}{P}} \sum_{w=0}^{\frac{W}{P}} v(I)[h, w] \hat{M}(I, w_i)[h, w]}{\sum_{h=0}^{\frac{H}{P}} \sum_{w=0}^{\frac{W}{P}} \hat{M}(I, w_i)[h, w]}, \tag{6.2}$$

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, $t_i(w) \in \mathbb{R}^{D_t}, i = 1 \ldots, T$, 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

$$k(c, w) = \alpha \hat{t}(w) + (1 - \alpha)\hat{c}, \tag{6.3}$$

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.

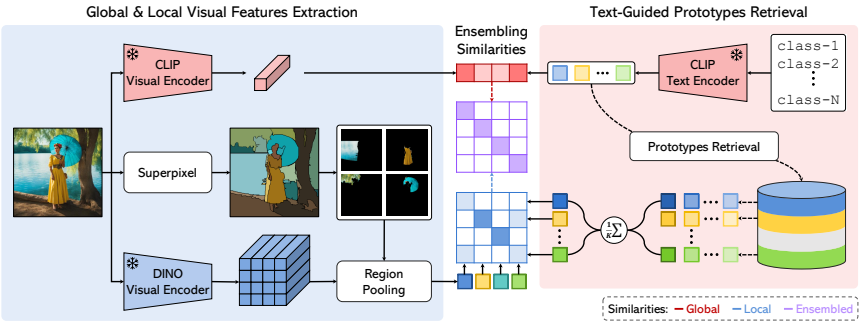


Figure 6.3: Overview of the inference process in FreeDA. Local (region-level) and global similarities are computed by employing, respectively, visual self-supervised and multimodal contrastive embedding spaces, and by comparing them with input texts and prototypes, built during the off-line stage.

6.1.2 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,

$$\overline{p}(c_i) = \frac{\sum_{k=1}^{K} p_{ik}}{K}, \tag{6.4}$$

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 $\overline{p}(I, c_i), \quad i = 1, \ldots, S$, 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 [286]. As discussed in Chapter 5, superpixels offer a compelling solution by introducing edge, shape, and boundary priors into attention-based models, thereby improving semantic segmentation performance. Building on these insights, we address the limitations of DINOv2 by incorporating a superpixel algorithm (*i.e.*, the Felzenszwalb's algorithm [75]) to partition the image by grouping pixels into class-agnostic non-overlapping regions according to their visual appearances and positions, thereby enhancing model's ability to delineate object boundaries more effectively.

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 Equation 6.2 to produce superpixel embeddings $r_i \in \mathbb{R}^{D_v}, \quad i = 1, \ldots, |R|$. 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 [191]), 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 $\hat{t}(c_i), \quad i = 1, \ldots, c_S$. 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

$$s(r_j, c_i) = \beta l(r_j, c_i) + (1 - \beta) g(I, c_i), \tag{6.5}$$

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.

6.2 Experiments

In this section, we evaluate the performance and contributions of the proposed FreeDA method across various benchmarks and scenarios. The evaluation begins with a detailed description of the experimental setup, including the datasets, metrics, and implementation details. FreeDA's performance is then benchmarked against state-of-the-art approaches, across diverse datasets and configurations. Ablation studies further dissect the impact of key components and design choices—such as visual backbones, global similarity computation, and superpixel-based mask refinement—on overall performance. Additional experiments provide insights into retrieval efficiency, parameter sensitivity, and qualitative outcomes, illustrating both the strengths and limitations of FreeDA in real-world applications.

6.2.1 Experimental Setup

Datasets. We evaluate FreeDA on the validation splits of traditional semantic segmentation benchmarks, namely Pascal VOC 2012 [72], Pascal Context [179], COCO Stuff [16], Cityscapes [48], and ADE20K [302, 303]. In particular, the validation sets of these datasets respectively contain 20, 59, 171, 150, and 19 semantic categories and 1449, 5104, 5000, 2000, and 500 images. 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 [16] 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.

Implementation Details. Textual sentences used as input in our diffusion-augmented prototype generation pipeline are taken from the COCO Captions dataset [35, 153]. 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 [201] 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 [185] 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 [191] 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 [75]. We build and leverage an efficient exact retrieval index through the `faiss` library [115] 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. In Section 6.2.3 we analize the impact of the these hyperparameters.

Evaluation Protocol. To perform all experiments, we follow the unified evaluation protocol for unsupervised open-vocabulary semantic segmentation established by Cha *et al*. [21]. Specifically, we consider 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.

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 [191]:

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

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 6.1: Negative prompts employed in Stable Diffusion during prototypes generation.

As discussed in [191], 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 6.1. 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.

Model	PAMR	Dataset	Parameters (M)		Similarity		mIoU				
			Total	Trainable	Textual	Visual	VOC	Context	Stuff	Cityscapes	ADE
ReCo [212]	✗	ImageNet1k★	313.0	0.0	✗	✓	57.7	22.3	14.8	21.1	11.2
GroupViT [258]	✗	CC12M+RedCaps♦	55.8	55.8	✓	✗	79.7	23.4	15.3	11.1	9.2
MaskCLIP [304]	✗	-	291.0	0.0	✓	✗	74.9	26.4	16.4	12.6	9.8
TCL [21]	✗	CC3M+CC12M♦	178.3	21.7	✓	✗	77.5	30.3	19.6	23.1	14.9
OVDiff [117]	✗	-	1,226.4	0.0	✗	✓	81.7	33.7	-	-	14.9
MaskCLIP [304]	✓	-	291.0	0.0	✓	✗	72.1	25.3	15.1	11.2	9.0
ReCo [212]	✓	ImageNet1k★	313.0	0.0	✗	✓	62.4	24.7	16.3	22.8	12.4
GroupViT [258]	✓	CC12M+YFCC♦	55.8	55.8	✓	✗	81.5	23.8	15.4	11.6	9.4
TCL [21]	✓	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 6.2: Comparison with state-of-the-art unsupervised open-vocabulary semantic segmentation models on Pascal VOC [72], Pascal Context [179], COCO Stuff [16], Cityscapes [48], and ADE20K [302, 303], without considering the unknown category. The markers ♦ and ★ refer, respectively, to datasets used for training and support only.

6.2.2 Comparison with the State-of-the-art

We first compare FreeDA with recent state-of-the-art approaches for unsupervised open-vocabulary semantic segmentation. Specifically, we include ReCo [212] and OVDiff [117] that, similarly to our approach, exploit the arbitrary input categories to obtain a set of visual references. While ReCo curates an archive based on ImageNet1k [59], OVDiff generates a set of synthetic references at inference time by conditioning on a fixed prompt template, without necessitating external support data. Also, we compare with MaskCLIP [304], which introduces some modifications to the CLIP architecture to exploit its multimodal embedding space, and GroupViT [258] and TCL [21] that rely on extensive contrastive training on large-scale datasets to learn a textual-visual alignment. When considering segmentation benchmarks with the background class, we also include ViewCo [197], SegCLIP [167], and OVSegmentor [260] that, analogously to GroupViT and TCL, are based on natural language supervision via contrastive learning paradigms.

Table 6.2 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 considered competitors when using the Pixel-Adaptive Mask Refinement (PAMR) proposed in [3] 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 [21], 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 6.3, we instead report the results on the three segmentation datasets, namely Pascal VOC, Pascal Context, and COCO Object, used to validate the effectiveness of segmentation methods when also considering the additional "unknown" category. Following [258], 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 restrain the comparison to methods

Model	PAMR	Training Dataset	mIoU		
			VOC	Context	Object
GroupViT [258]	-	CC12M+RedCaps	50.4	18.7	27.5
MaskCLIP [304]	-	-	38.8	23.6	20.6
ReCo [212]	-	-	25.1	19.9	15.7
ViewCo [197]	-	CC12M+YFCC	52.4	23.0	23.5
SegCLIP [167]	-	CC3M+COCO Captions	52.6	24.7	26.5
TCL [21]	-	CC3M+CC12M	51.2	24.3	30.4
OVSegmentor [260]	-	CC4M	53.8	20.4	25.1
GroupViT [258]	✓	CC12M+YFCC	51.1	19.0	27.9
MaskCLIP [304]	✓	-	37.2	22.6	18.9
TCL [21]	✓	CC3M+CC12M	55.0	30.4	31.6
FreeDA (ViT-L)	-	-	**55.4** (+0.4)	**38.3** (+7.9)	**37.4** (+5.8)

Table 6.3: Comparison with state-of-the-art unsupervised open-vocabulary semantic segmentation models on the validation sets of Pascal VOC [72], Pascal Context [179], and COCO Object [16], when considering the unknown category.

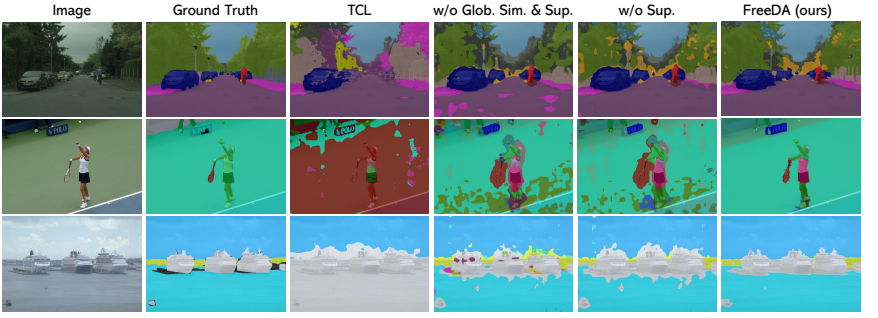


Figure 6.4: Qualitative results of FreeDA in comparison with TCL [21], with and without global similarities and superpixels.

that do not employ specific techniques to take into account the background of the scene but instead perform a threshold 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 [3]. In particular, FreeDA reaches 55.4, 38.3, and 37.4 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 [21] using PAMR as mask refinement technique).

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 6.4: 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 [72], Cityscapes [48], and ADE20K [302, 303].

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 6.4.

6.2.3 Ablation Studies and Analyses

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 [225] pre-trained for image classification on ImageNet1k and based on ViT-L/16, CLIP [191] in both its ViT-B/16 and ViT-L/14 versions, DINO [20] based on the ViT-B/16 architecture, and our final choice DINOv2 [185] 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 path size of 16, thus always having features with a spatial size equal to 37×37. To

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

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 6.4, 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 global features and superpixel-based mask proposals, we report in the lower part of Table 6.4 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 few objects occupying large areas of the scene, thus favoring global features instead of local ones.

Impact of Backbone Size. In Table 6.5 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 6.2, 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.

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

Superpixel Algorithms and Prototype Aggregation Strategies. In Table 6.6, we instead validate the choice of employing Felzenszwalb's algorithm [75] to extract superpixels by comparing it with three widely adopted superpixel proposal algorithms, namely Watershed [105], SLIC [1], and SEEDS [228]. 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 [3] as a mask refinement method. 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 assess the aggregation strategy in FreeDA, where retrieved prototypes are aggregated by computing their average embedding (*i.e.*, "mean embedding" in Table 6.6), we compare it against two alternative approaches: aggregating local similarities of retrieved prototypes using either their mean or maximum values (*i.e.*, "mean similarity" and "max similarity"). The results show that averaging the embeddings of all retrieved prototypes yields the best performance 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 examined the impact of using an approximate search strategy. Specifically, the left plot of Figure 6.5 shows the trade-off between speed and performance when using a graph-based HNSW (Hierarchical Navigable Small

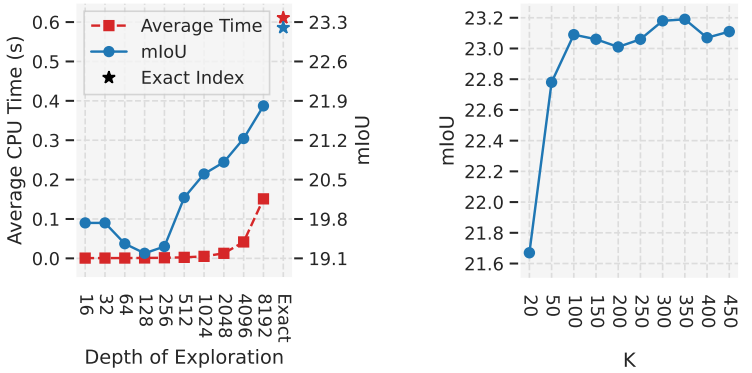


Figure 6.5: Retrieval results when using an approximate index (left) and varying the number of retrieved key-prototype pairs (right).

World) index [173]. We report the CPU times to search the most similar $K = 350$ key embeddings while varying the depth of exploration in the index, alongside 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 6.5, we instead show the performance variation observed when changing the number K of searched keys. These experiments were conducted on the ADE20K dataset. As it can be seen, using an approximate index only slightly decreases performance while consistently improving computational efficiency. Furthermore, increasing the number of retrieved key embeddings beyond a certain point does not yield significant performance gains, whereas reducing this number leads to a noticeable drop in performance.

Effect of Superpixel Parameters. Felzenszwalb *et al*. [75] 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.

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

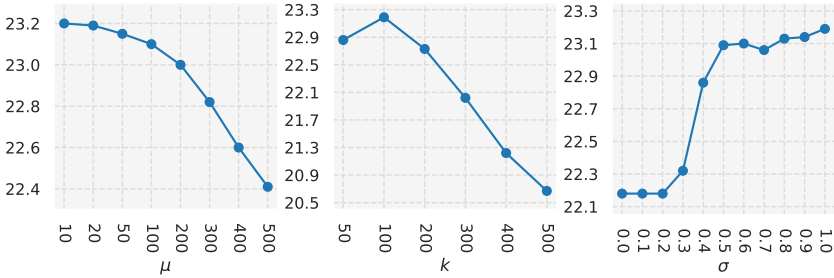


Figure 6.6: Effect of the variation of superpixel hyperparameters on ADE20K, measured in terms of mIoU.

In Table 6.7, we report the parameter values employed on the examined datasets. Figure 6.6 further shows the performance variations obtained when altering these parameters on the ADE20K dataset [302, 303]. Notably, minor variations in these parameters have negligible effects on final performance. However, imposing large superpixels through minimum size or scale of observation can significantly degrade the results.

Impact of Caption Context. In Section 6.1.1, 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 6.8, 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.

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

Table 6.8: 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 6.9: mIoU results with DINOv2 for local/global matching.

Impact of Unimodal Global Matching. In Table 6.9, 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 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 w.r.t. the unimodal DINOv2 global features.

In-the-wild Results. In Figure 6.7 we report a collection of in-the-wild examples obtained by prompting our model with free-form textual inputs. Specifically, we extract noun chunks from sample captions of the COCO Captions validation set. After removing stop-words, the noun chunks are utilized as input categories for segmenting the corresponding 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. Figure 6.8 showcases 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

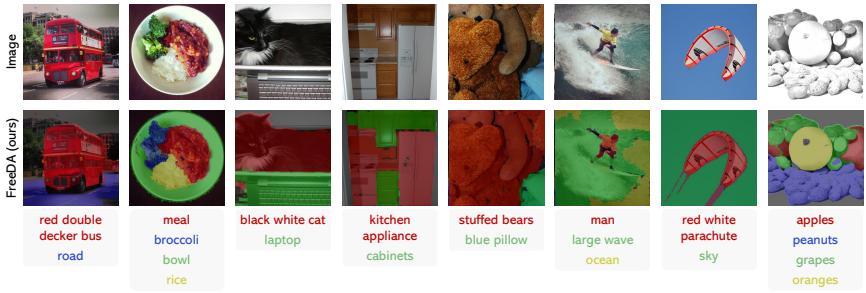


Figure 6.7: In-the-wild segmentation results obtained by prompting our model with diverse free-form textual inputs.

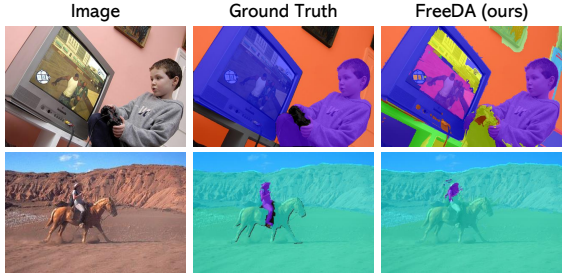


Figure 6.8: Sample failure cases.

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.

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. Specifically, our method enables the visualization of generated images associated with the retrieved prototypes for a given input category, alongside their associated attribution maps and binary masks. Figure 6.9 exemplifies the explainability features of our approach by presenting examples of retrieved prototypes for a specific category, highlighted within the captions in which the corresponding noun is mentioned. Additionally, we provide the generated images, attribution maps, and binarized masks linked to these prototypes.

Figure 6.9: 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).

Chapter 7

Fine-grained Segmentation Error Analysis

Thus far, we have explored the versatility of Transformers in tackling a wide range of challenges, from achieving precise boundary delineation in dense prediction tasks to aligning textual and visual modalities for generalizing beyond seen categories. While these advancements underscore the transformative potential of Vision Transformers (ViTs) in redefining segmentation paradigms, they also highlight the need for comprehensive evaluation of model performance across diverse application domains and learning settings.

Semantic segmentation, as a foundational task in Computer Vision, serves as a critical benchmark for assessing the strengths and limitations of modern architectures. Its applications span numerous domains, each with unique challenges and requirements. For instance, in medical image segmentation, precise delineation of class boundaries is paramount due to its direct impact on clinical decision-making [306]. Conversely, in land use and land cover segmentation [58], ground-truth annotations may lack clear-cut boundaries—or such boundaries may not even exist—shifting the focus toward accurate classification and coarse localization. Furthermore, there are different learning settings in semantic segmentation, *e.g.*, weakly, semi-, or unsupervised, which also pose different challenges. For

This chapter is related to the publication "Maximilian Bernhard, Roberto Amoroso, Yannic Kindermann, Lorenzo Baraldi, Rita Cucchiara, Volker Tresp, and Schubert Matthias. What's Outside the Intersection? Fine-grained Error Analysis for Semantic Segmentation Beyond IoU, WACV 2024". Work realized during the ELLIS internship at the LMU University in Munich, Germany.

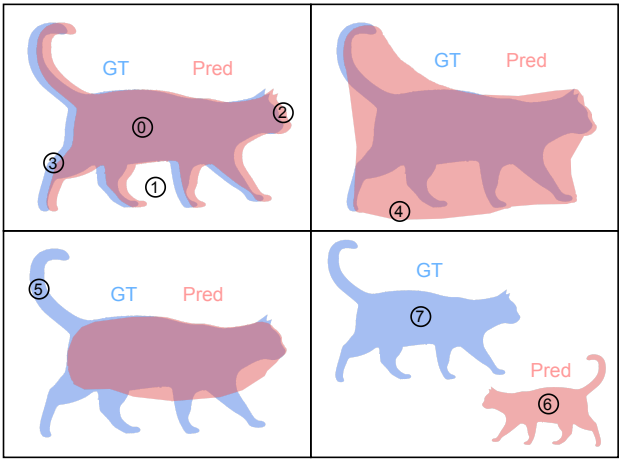


Figure 7.1: Our proposed error categorization. Each pixel is assigned to one of these categories: 0: true positive, 1: true negative, 2: false positive boundary, 3: false negative boundary, 4: false positive extent, 5: false negative extent, 6: false positive segment, 7: false negative segment.

example, the correct segmentation of non-discriminative object parts is known to be a crucial problem in weakly supervised semantic segmentation and has attracted much attention from researchers [25, 125, 133, 139, 238].

Numerous semantic segmentation architectures and methods have been proposed to account for the diversity of the task. Typically, these semantic segmentation methods and applications are evaluated via the *mean intersection over union ($mIoU$)*, which is currently the gold standard for model comparison and benchmarking. As the name already suggests, $mIoU$ measures the average intersection over union across all classes in a dataset. While it has the advantage of being an interpretable metric, it does not suit every application and dataset equally well. Moreover, $mIoU$ does not convey any insights into what types of errors a model makes. Other metrics that are occasionally used, *e.g.*, *F1-measure*, *Pixel Accuracy* or *Boundary IoU* [39], behave similarly and share this shortcoming. Thus, we argue that the current evaluation metrics, and especially $mIoU$ as a single metric, are not sufficient to evaluate semantic segmentation models in a differentiated way across various applications and datasets.

Contributions. In this chapter, we propose an intuitive error categorization that allows us to assess models w.r.t. various aspects, thereby meeting the diversity of semantic segmentation applications. We distinguish segmentation errors by assigning one of the following three categories (illustrated in Figure 7.1) to every incorrect pixel: (1) **Boundary errors** indicate that the model was able to correctly detect a transition between two semantic classes in the region of the respective pixel, but failed at the exact delineation of the boundary. (2) **Extent errors** indicate that a model recognized an instance (represented by a contiguous segment) and its class, but, in contrast to boundary errors, severely over- or underestimated its extent (*e.g.*, missing non-discriminative parts). (3) **Segment errors** are in no apparent relation to true positive predictions, *i.e.*, entire segments are mispredicted. Thus, a high number of segment errors indicates a model's weakness in the classification of its predicted segments. We propose that the mean error over union for each of these error types represents an intuitive extension of $mIoU$ and allows to determine how much loss in $mIoU$ each error type causes. Based on these error rates, conclusions about the model strengths and weaknesses can be drawn.

To validate our error categorization framework, we conduct extensive experiments comparing a variety of methods, datasets, and learning settings. Notably, we find that the advantage of cutting-edge segmentation architectures, such as MaskFormer [41], Mask2Former [40], and OneFormer [109], stems from a superior capability in precisely delineating boundaries and properly predicting segment extents. At the same time, these architectures have a remarkable weakness in the classification of segments. Building on these insights, we combine these architectures with complementary models that address their classification shortcomings, achieving consistent improvements in $mIoU$. This underscores the utility of our error categorization framework in guiding model development and evaluation for diverse semantic segmentation tasks.

The rest of the chapter is organized as follows: Section 7.1 introduces our novel categorization of segmentation errors. This section defines key preliminary concepts and provides a high-level distinction between boundary, extent, and segment errors. We also present a statistical framework for quantifying these error categories, introducing metrics such as Error over Union (EoU) and its re-normalized variant for more nuanced insights. Section 7.2 outlines the experimental validation of the proposed framework, including sensitivity analyses, comparisons of state-of-the-art models, and case studies on combining complementary models. Finally, this section examines error distributions across datasets, specific classes, and learning paradigms while offering qualitative analyses and formal proof of the disjointness of the error categories.

7.1 Segmentation Error Categorization

Semantic segmentation is fundamentally a per-pixel classification task, as reflected in its evaluation via IoU, treating each incorrect pixel separately and identically. However, from a human perspective, segmentation errors can appear rather diverse, as our perception is focused on entire visual instances formed by segments and objects instead of single pixels. Thus, it is sensible to base an error categorization on the concept of contiguous segments, considering the relations between pixels. On a high level, we distinguish between erroneous pixels belonging to (at least partially) correctly predicted segments (*i.e.*, boundary and extent errors) and pixels belonging to completely erroneous segments (*i.e.*, segment errors). A visual overview of these error types is illustrated in Figure 7.1.

7.1.1 Notation and Preliminary Definitions

Similar to the $mIoU$, our proposed error categorization framework independently evaluates each class in a dataset. Consequently, for clarity, hereafter we consider only a single class. Given an image with pixel locations $\Omega = [H] \times [W]$, we denote the binary ground-truth segmentation with $G \in \{0,1\}^{H \times W}$ and the corresponding binary prediction with $P \in \{0,1\}^{H \times W}$, *i.e.*, zero and one indicate background and foreground for the considered class, respectively. The true positive pixels are defined as $TP = \{x \in \Omega \mid G_x = 1 \wedge P_x = 1\}$ and FP, FN, and TN follow analogously. We define the d-neighborhood of a pixel as $\mathcal{N}_d(x) = \{x' \in \Omega \mid \delta(x, x') \leq d\}$, where $\delta(\cdot, \cdot)$ denotes the Euclidean pixel distance rounded to the nearest integer. Thus, $\mathcal{N}_1(x)$ describes x plus its eight neighboring pixels. Furthermore, we introduce an operator $\mathcal{S}(\cdot)$, that extracts all contiguous segments with label one from a binary segmentation mask such as G or P. For ease of notation, we also allow a binary mask represented by a set of pixels describing the locations of ones as input to $\mathcal{S}(\cdot)$. Each contiguous segment $s \in \mathcal{S}(\cdot)$ is represented by a set of pixel locations, *i.e.*, $s \subseteq \Omega$. Moreover, $\mathcal{S}(\cdot)_x$ denote the unique contiguous segment that contains pixel location x, given that the provided binary mask has label one at location x.

7.1.2 Boundary Errors

A boundary error in our categorization arises when the transition between the foreground and background of a class is identified but not accurately delineated. Thus, we first formulate a preliminary definition via the occurrence of true positive

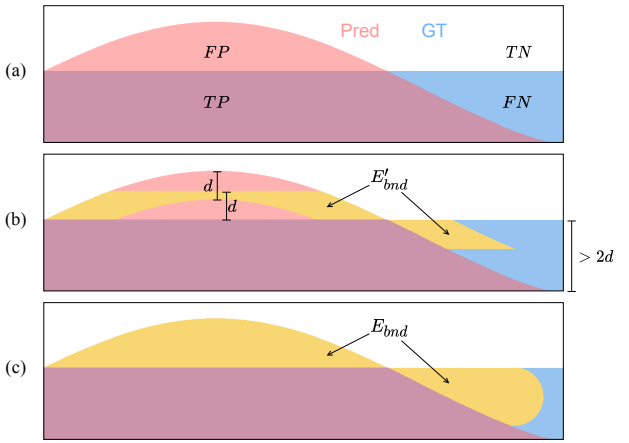


Figure 7.2: Extending the boundary area E'_{bnd} in subfigure (b) to E_{bnd} in subfigure (c) via Equation 7.2 avoids unwanted transitions in the form of $TP \rightarrow E_{ext} \rightarrow E'_{bnd} \rightarrow E_{ext} \rightarrow TN$.

and true negative pixels in the neighborhood, *i.e.*,

$$\begin{aligned} FP'_{bnd} &= \{x \in FP \mid \mathcal{N}_d(x) \cap TP \neq \emptyset \wedge \mathcal{N}_d(x) \cap TN \neq \emptyset\}, \\ FN'_{bnd} &= \{x \in FN \mid \mathcal{N}_d(x) \cap TP \neq \emptyset \wedge \mathcal{N}_d(x) \cap TN \neq \emptyset\}, \\ E'_{bnd} &= FP'_{bnd} \cup FN'_{bnd}. \end{aligned} \tag{7.1}$$

According to this definition, boundary errors are not just prediction errors along the boundaries of the ground truth but require proximity to the boundaries of both the ground truth and the prediction. The dependency on the distance parameter d is discussed in Section 7.1.6 and suppressed in the notation of the error sets for readability.

In addition, we propose two modifications to this definition to account for unwanted effects. First, to avoid transitions between boundary and non-boundary errors (see Figure 7.2), we extend the boundary error area via

$$\begin{aligned} &FP''_{bnd} = FP'_{bnd} \cup \{x \in FP \mid \mathcal{N}_d(x) \cap FP'_{bnd} \neq \emptyset\}, \\ &FN''_{bnd},\ E''_{bnd} \text{ analogous.} \end{aligned} \tag{7.2}$$

Thus, boundary errors can be at most $2d$ pixels away from true positive and true negative pixels.

Second, we remove all contiguous segments that have no true positives or no true negatives as direct neighbors, *i.e.*,

$$FP_{bnd} = \bigcup \{ s \in \mathcal{S}(FP''_{bnd}) \mid \exists_{x_1 \in s} : \mathcal{N}_1(x_1) \cap TP \neq \emptyset \\ \wedge \exists_{x_2 \in s} : \mathcal{N}_1(x_2) \cap TN \neq \emptyset \},$$
$$FN_{bnd},\ E_{bnd} \text{ analogous.} \tag{7.3}$$

Hence, if a contiguous segment of potential boundary errors $s \in \mathcal{S}(FP''_{bnd})$ is not adjacent to at least one true positive and one true negative, its pixels are not considered boundary errors. This is because, in such cases, there is no valid transition between the foreground and background of the class.

7.1.3 Extent Errors

Extent errors describe errors that occur when a segment has been recognized, but largely over- or underestimated in its extent (*e.g.*, when non-discriminative parts are not recognized). In the false positive case, they are pixels that belong to a contiguous predicted segment intersecting with the ground-truth foreground, and in the false negative case, there are pixels that belong to a contiguous ground-truth segment intersecting with the predicted foreground. Formally, we define extent errors as:

$$\begin{aligned} FP_{ext} &= \{ x \in FP \setminus FP_{bnd} \mid \mathcal{S}(P)_x \cap TP \neq \emptyset \}, \\ FN_{ext} &= \{ x \in FN \setminus FN_{bnd} \mid \mathcal{S}(G)_x \cap TP \neq \emptyset \}, \\ E_{ext} &= FP_{ext} \cup FN_{ext}. \end{aligned} \tag{7.4}$$

In other words, extent errors can be thought of as error pixels that would become boundary errors if the distance parameter d was increased to infinity. As extent errors can have an arbitrary distance to the boundary of their corresponding ground-truth segment, we consider them more severe than boundary errors.

7.1.4 Segment Errors

Having established the concepts of boundary and extent errors, we now turn our attention to a distinct category of errors that lack any direct association with true positive predictions. Specifically, we are now dealing with predicted segments that do not have any intersection with the ground-truth foreground (false positive) and ground-truth foreground segments that do not have any intersection with the

predicted foreground (false negative). We refer to these discrepancies as segment errors and define them as follows:

$$\begin{aligned} FP_{seg} &= \{x \in FP \mid \mathcal{S}(P)_x \cap TP = \emptyset\}, \\ FN_{seg} &= \{x \in FN \mid \mathcal{S}(G)_x \cap TP = \emptyset\}, \\ E_{seg} &= FP_{seg} \cup FN_{seg}. \end{aligned} \tag{7.5}$$

Therefore, segment errors occur when models predict wrong classes for entire segments, and a large number of segment errors indicates poor performance in classification.

7.1.5 Error Statistics

Error over Union The proposed error categorization ensures that each false positive and false negative pixel is uniquely assigned to a specific error category. This allows for a precise quantification of the number of pixels associated with each error type across all images. Using this categorization, the error over union can be defined in a manner analogous to the IoU, *i.e.*,

$$E_\star oU = \frac{|E_\star|}{|U|}, \star \in \{bnd,\ ext,\ seg\}, \tag{7.6}$$

where $U = TP \cup FP \cup FN$. Furthermore, we define $mE_{\{bnd,ext,seg\}}oU$ analogously to $mIoU$ as the mean error over union over all classes. As the three error categories are disjoint, the sum of the IoU and all $E_\star oU$ terms equals one,

$$IoU + E_{bnd}oU + E_{ext}oU + E_{seg}oU = 100\%. \tag{7.7}$$

Clearly, the same holds for $mIoU$ and $mE_{\{bnd,ext,seg\}}oU$. Therefore, the error over union quantifies how much loss in $mIoU$ each error type causes. This kind of interpretability makes the proposed error categorization easy to grasp and perfectly fit with the evaluation via $mIoU$.

Re-normalized Error over Union However, when comparing models with rather different strengths, it is sensible to consider another quantity next to $mIoU$ and $mE_{\{bnd,ext,seg\}}oU$ to gain reliable insights. Consider a model A, which has better classification capabilities and, therefore, a lower $E_{seg}oU$ than another model B. Due to the fewer segment errors, model A will face more occasions to produce boundary and extent errors. Hence, the errors over union for boundary

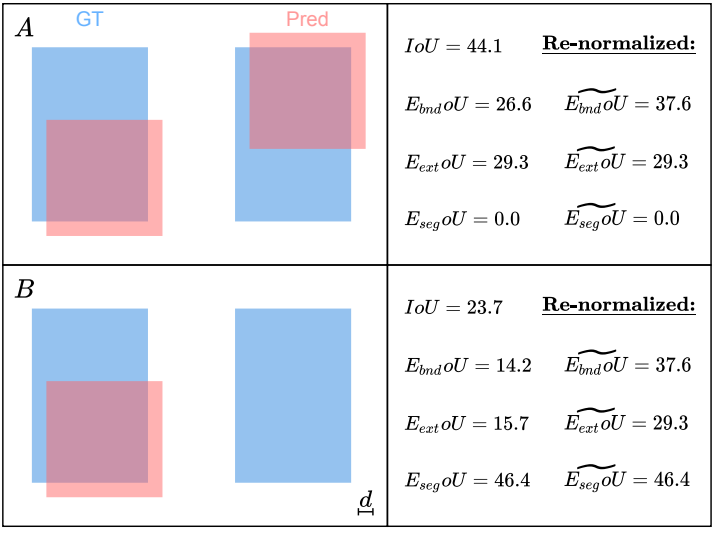


Figure 7.3: The effect of re-normalizing $E_\star oU$. A (top) and B (bottom) are two models equally good at segmenting boundaries and extents. However, A has substantially higher values for $E_{bnd}oU$ and $E_{ext}oU$ than B because B completely misses the second ground-truth segment. The re-normalized error over union accounts for this effect such that A and B solely differ in $\widetilde{E_{seg}oU}$, while $\widetilde{E_{bnd}oU}$ and $\widetilde{E_{ext}oU}$ are identical.

and extent would show larger values for model A, even if A and B had equal performances in these regards. Carrying this logic forward, a lower $E_{ext}oU$ will cause larger values for $E_{bnd}oU$. To account for this, we propose the re-normalized errors over union

$$\begin{aligned}
\widetilde{E_{seg}oU} &= E_{seg}oU, \\
\widetilde{E_{ext}oU} &= \frac{|E_{ext}|}{|U| - |E_{seg}|} = \frac{|E_{ext}|}{|TP| + |E_{bnd}| + |E_{ext}|}, \\
\widetilde{E_{bnd}oU} &= \frac{|E_{bnd}|}{|U| - |E_{seg}| - |E_{ext}|} = \frac{|E_{bnd}|}{|TP| + |E_{bnd}|}.
\end{aligned} \tag{7.8}$$

That is, for each error type, we remove the more fundamental errors (in terms of localization) from the denominator. We can interpret the re-normalized error

over union as the loss in IoU caused by an error type if the model had perfect performance w.r.t. to the other error types. The re-normalized error over union still ranges from zero to one; however, it loses the property that the IoU plus the error rates sum up to one. An illustration of the effect of the re-normalization is provided in Figure 7.3.

7.1.6 Choosing the Distance Parameter d

For choosing the parameter d, similar considerations as for the pixel distance parameter in Boundary IoU [39] apply. In our error categorization, d determines how far away from the true boundary a boundary error may occur (at most $2d$). Decreasing d will lead to fewer boundary and more extent errors, whereas setting d loosely will increase the number of boundary errors and small or thin segments may only consist of their boundaries, thereby reducing the number of extent errors. Overall, d should be chosen such that a deviation of $2d$ can still be considered close to the true boundary for the application and the dataset at hand.

Like [39], we set d dependent on the image size. For ADE20K [302], Pascal-VOC [72], COCO-Stuff 164k [16], and STARE [102], we select d as 1% of the image diagonal, whereas for CityScapes [48] and iSAID [243], we use 0.25% and 0.5% of the image diagonal, respectively, due to their high-resolution images and high-quality annotations.

7.2 Experiments

In this section, we present a comprehensive evaluation of our proposed segmentation error categorization framework across various scenarios. First, we conduct a sensitivity analysis to investigate how different error types respond to systematic transformations of ground-truth masks, providing insights into the distinct behaviors of boundary, extent, and segment errors. Next, we compare state-of-the-art semantic segmentation models using our error metrics to uncover fundamental differences in their prediction characteristics. We then explore the potential of combining models with complementary strengths to achieve improved performance. Additional analyses examine error distributions across datasets, learning settings, and individual classes, highlighting the versatility of our approach in addressing diverse challenges. Finally, we provide qualitative visualizations and theoretical proofs to further substantiate the validity of our methodology.

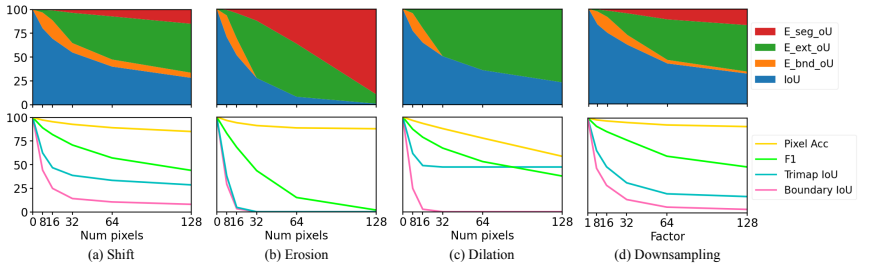


Figure 7.4: Sensitivity analysis for our error types (top) and existing metrics (bottom) under different transformations.

7.2.1 Sensitivity Analysis

To facilitate a better understanding of our proposed error categorization, we analyze the sensitivity of the different error types toward systematic transformations of ground-truth masks. Specifically, we take binary ground-truth masks from ADE20K, corrupt them, and compute the error rates compared to the original ground truth. For the corruptions, we use shifting, erosion, dilation, and downsampling of different severities to simulate various erroneous prediction behaviors. In particular, with downsampling, we mimic the predictions of a model that cannot produce precise delineations at class boundaries. To have well-defined segmentation masks with exactly one class label for each pixel after applying the transformations, we consider only binary masks and compute the metrics for a single global foreground class.

The results, illustrated in Figure 7.4, reveal distinct patterns across transformations. For small shifts (Figure 7.4 (a)), boundary errors increase initially. As the shift magnitude grows, extent errors emerge before segments fail to overlap with their originals, leading to segment errors. For erosion and dilation (see Figure 7.4 (b,c)), we observe a similar behavior for boundary errors under light corruptions. Also, extent errors grow rapidly if the number of erosion/dilation pixels is further increased. However, in this case, boundary errors are completely replaced by extent errors as erosion and dilation inevitably move boundaries away from their original locations. Furthermore, erosion at high severity introduces segment errors when small segments disappear entirely, a phenomenon absent in dilation. Similar to shift, erosion, and dilation, the downsampling corruption in Figure 7.4 (d) also leads to an increase in boundary errors for small severities. When the downsampling factor is increased to rather extreme values (*e.g.*, 64

or 128), we observe segment errors because small segments are lost at such low resolutions.

In contrast, if we look at existing evaluation metrics in the bottom row of Figure 7.4, we observe monotonically decreasing curves as these metrics are primarily designed to measure the overall segmentation quality, thus providing limited insights into specific prediction errors. In contrast, the error rates for our proposed categories reach their maxima at different severities and behave differently under distinct corruptions, making it easier to develop an understanding of the prediction errors. In summary, our analysis demonstrates that the proposed error categories align with expected outcomes under systematic corruption and offer valuable complementary insights to existing evaluation metrics.

7.2.2 Comparing State-of-the-art Models

In Table 7.1, we provide a broad comparison of state-of-the-art semantic segmentation models on ADE20K. The models under consideration are categorized into three architectural paradigms: CNN-based, Transformer-based, and mask-classification-based approaches. For the evaluation, the primary metric of interest is the re-normalized error over union $\widetilde{mE_{\star}oU}$, but we also report the standard $mE_{\star}oU$ for completeness.

The CNN architectures (1) PSPNet [295], (2) DeepLabV3 [28], and (3) DeepLabV3+ [29] generally perform rather similarly w.r.t. to $mIoU$ and all error types. If we compare DeepLabV3 and DeepLabV3+ more closely, we can see that DeepLabV3+ outperforms DeepLabV3 mainly in terms of boundary errors (18.8 vs. 20.0% $\widetilde{mE_{bnd}oU}$). Interestingly, this observation is in line with the claim that DeepLabV3+ is able to predict more precise boundaries [29].

The Transformer-based methods (4) SETR [299], (5) SegFormer [255], and (6) Segmenter [216], reach higher $mIoU$ scores than the considered CNN models (partly due to stronger backbones). These performance gains mainly stem from a reduction in extent and segment errors, which is intuitive considering Transformers' superior capabilities in contextualization and global reasoning.

However, (7) MaskFormer [41] and its successors (8,9) Mask2Former [40] and (11,12) OneFormer [109] are able to outperform these Transformer methods while showing a remarkably different distribution of errors. The paradigm of classifying entire masks instead of single pixels leads to substantially lower error rates for boundaries and extents. On the other hand, these models have relatively high numbers of segment errors, *e.g.*, 34.5% for (8) Mask2Former + R101-D32.

				$\mathbf{mE_\star oU}$			$\mathbf{\widetilde{mE_\star oU}}$		
	Architecture	**Backbone**	**mIoU**	*bnd*	*ext*	*seg*	*bnd*	*ext*	*seg*
(1)	PSPNet [295]	R101-D8	44.4	9.0	15.9	30.7	19.2	25.2	30.7
(2)	DeepLabV3 [28]	R101-D8	45.0	9.5	15.9	29.6	20.0	24.9	29.6
(3)	DeepLabV3+ [29]	R101-D8	45.5	9.0	15.8	29.8	18.8	24.1	29.8
(4)	SETR [299]	ViT-L	48.3	10.1	15.1	26.6	19.5	22.0	26.6
(5)	SegFormer [255]	MiT-b5	49.6	9.6	13.0	27.8	18.1	19.9	27.8
(6)	Segmenter [216]	ViT-L	52.2	10.1	12.7	25.0	18.4	18.8	25.0
(7)	MaskFormer [41]	Swin-L	54.1	7.1	8.4	30.3	12.7	13.0	30.3
(8)	Mask2Former [40]	R101-D32	48.6	6.7	10.2	34.5	13.4	16.7	34.5
(9)	Mask2Former [40]	Swin-L	56.0	7.2	8.2	28.5	12.7	12.7	28.5
(10)	UPerNet [5, 254]	BEiT-L	56.3	8.4	11.7	23.6	14.5	17.1	23.6
(11)	OneFormer [109]	Swin-L	57.0	6.5	8.1	28.4	11.2	12.1	28.4
(12)	OneFormer [109]	DiNAT-L	58.0	6.6	7.9	27.6	11.0	11.4	27.6
(13)	UPerNet [254]	R50	42.1	8.9	17.5	31.5	20.1	28.1	31.5
(14)	UPerNet [254]	R101	43.8	8.7	16.0	31.5	19.3	25.8	31.5
(15)	UPerNet [254]	ViT-B	48.8	9.3	13.8	28.1	18.6	21.6	28.1
(16)	UPerNet [254]	Swin-B	50.8	9.1	13.1	27.1	17.0	19.8	27.1
(17)	UPerNet [254]	BEiT-B	53.1	8.8	13.8	24.3	16.0	19.6	24.3

Table 7.1: Results of state-of-the-art models on ADE20K val. CNN models are mostly outperformed by early Transformer-based models w.r.t. extent and segment errors. Mask-classification-based models (7-9,11,12) reach new levels of $mIoU$ only because of large improvements w.r.t. boundary and extent errors.

The only method in our comparison that does not follow this paradigm but that can rival these architectures with a similarly sized backbone is (10) UPerNet + BEiT-L [5, 254]. The main strength of this model seems to be classification, as shown by the lowest segment error rate in our analysis (23.6%). However, it produces significantly more boundary and extent errors than the mask-classification-based models. Hence, there are fundamental differences in the predictions and errors of state-of-the-art models, which we further analyze in Section 7.2.3.

In addition, we compare different backbones within the context UPerNet (13-17) in the lower section of Table 7.1. Once again, we observe that Transformer models outperform CNNs mostly w.r.t. extent and segment errors. Furthermore, stronger backbones reduce errors across all categories rather uniformly, without a distinct advantage favoring any particular type of error.

Segmentor		Classifier	mIoU (+Δ)
(8) Mask2Former	∘	(4) SETR	50.1 *(+1.5)*
(9) Mask2Former	∘	(6) Segmenter	56.4 *(+0.4)*
(9) Mask2Former	∘	(10) UPerNet+BEiT	56.9 *(+0.6)*
(11) OneFormer	∘	(10) UPerNet+BEiT	57.9 *(+0.9)*
(12) OneFormer	∘	(10) UPerNet+BEiT	58.6 *(+0.6)*
(4) SETR	∘	*(8) Mask2Former*†	48.0 *(–0.3)*

Table 7.2: Combining models with complementary strengths consistently improves performance on ADE20K val, even beyond the state-of-the-art set by OneFormer [109]. $A \circ B$ denotes applying model A for segmentation after multi-label classification with B. †: Combining model weaknesses for comparison.

7.2.3 Combining Models with Complementary Strengths

Our evaluation of architectures in Table 7.1 has shown that Mask2Former and OneFormer produce many segment errors, and their overall strong performances mainly come from minimal boundary and extent errors. At the same time, architectures like Segmenter and UPernet + BEiT produce substantially fewer segment errors, and their performance bottlenecks are boundary and extent errors. Therefore, the question arises whether models with such complementary strengths can benefit from each other. To test this hypothesis, we evaluate simple combinations of two models, where one model is employed for multi-label image classification and another model produces segmentation masks for the predicted classes. More precisely, for each class that is deemed to be absent in the image by the first model (classifier), we set the predicted logits of the second model (segmentor) to minus infinity. We deliberately kept the combination strategy simple, avoiding more sophisticated techniques since the primary goal of this experiment is to demonstrate the practicability of insights gained through our error analysis.

The results for these combinations of models are provided in Table 7.2. First, we combine (8) Mask2Former + ResNet-101, having the highest segment error rate in Table 7.1 (34.5%), with (4) SETR, having a similar overall performance, but much fewer segment errors (26.6%). This combination increases Mask2Former's $mIoU$ from 48.6% to 50.1%, while decreasing its $mE_{seg}oU$ from 34.5% to 33.4%. Conversely, if we combine these two models the other way around, *i.e.*, using (8) Mask2Former for classification and (4) SETR for segmentation, their weaknesses are emphasized, leading to a drop in $mIoU$ (48.0%).

		mE$_\star$oU		
Dataset	**mIoU**	*bnd*	*ext*	*seg*
ADE20K [302]	42.5	9.3	17.4	30.8
COCO-Stuff 164k [16]	40.5	6.0	15.6	37.8
PascalVOC 2012 [72]	77.3	3.6	11.2	8.0
CityScapes [48]	79.6	8.4	6.5	5.6
iSAID [243]	65.4	11.8	7.2	15.6
STARE† [102]	84.0	15.4	0.4	0.2

Table 7.3: Comparing error rates across datasets from different domains with PSPNet (R50-D8). The datasets exhibit significantly different error distributions. †: UNet-S5-D16 backbone.

The observed improvement in performance is not unique to Mask2Former with the relatively weak ResNet backbone, but can also be achieved with the stronger Swin-L backbone. Even when combining (9) Mask2Former + Swin-L with (6) Segmenter + ViT-L, which has a 3.8% lower $mIoU$, the combined $mIoU$ improves by 0.4%, reaching 56.4%. Finally, we combine both (11) OneFormer + Swin-L and (12) OneFormer + DiNAT-L with (10) UPerNet + BEiT-L, reaching $mIoU$s of 57.9% and 58.6%, respectively. This surpasses the previous state-of-the-art on ADE20K val, which was set by OneFormer (single-scale inference, no additional training data, see [109] for details).

These findings underscore that segment errors are a major limiting factor for mask-classification-based models such as Mask2Former and OneFormer. This insight demonstrates the usefulness of our error analysis methodology and opens up an intriguing research direction for improving these models and further advancing the state-of-the-art in semantic segmentation.

7.2.4 Additional Comparative Analyses

Comparing Datasets. Since there are not only a large number of model architectures for semantic segmentation but also a variety of commonly used benchmark datasets coming from different application domains, it is worth looking into error rates on different datasets as well. In Table 7.3, we compare error rates of PSPNet on ADE20K [302], CityScapes [48], PascalVOC 2012 [72], COCO-Stuff 164k [16], iSAID [243], and STARE [102]. ADE20K, PascalVOC, and COCO-

Stuff contain natural scene images. PascalVOC only contains 21 semantic classes (including background), making classification comparatively easy. Thus, the fraction of segment errors on PascalVOC is 8.0%, much smaller than on ADE20K (30.8%) and COCO-Stuff 164k (37.8%), having 150 and 171 classes, respectively. Also, object contours in PascalVOC are surrounded by a band of ignore pixels, leading to fewer boundary errors (3.6%) as well. On all three of these natural scene datasets, segment errors and extent errors dominate.

CityScapes is an autonomous driving dataset containing urban road scenes. It has high-quality annotations for 19 classes, enabling a high overall $mIoU$. With a segment error rate of 5.6%, classification on this dataset seems less challenging. The remote sensing dataset iSAID contains many small objects such as vehicles in overhead imagery, leading to a relatively high boundary error rate of 11.8%. A higher boundary error rate in our selection can only be observed on STARE, a medical image dataset for the segmentation of retinal vessels. As it only contains the classes "vessel" and "background", the segment error rate is only 0.2% on this dataset. Also, extent errors are very low, making boundary errors the main limiting factor. This is because the vessels in the dataset are usually a single large contiguous structure with very thin and fine elements.

Overall, we can see that, similar to model architectures, different datasets and different domains have rather distinctive features and come with different challenges, which is reflected in the error distributions. Therefore, we argue that our error analysis can help in selecting or developing a suitable segmentation architecture for a specific task.

Comparing Learning Settings. In addition to exploring diverse model architectures and datasets, semantic segmentation research also involves various learning settings. A comparative analysis of these settings is presented in Table 7.4, using PascalVOC as the benchmark due to its popularity in weakly and semi-supervised segmentation tasks. Looking at the image-level weakly supervised method BECO [202], we see that BECO is slightly superior to the fully supervised DeepLabV3+ in terms of segment errors (7.1% vs. 8.1%). However, the weak supervision does not provide any localization information, leading to higher numbers of boundary and extent errors. Extent errors are particularly high for BECO as segmenting non-discriminative parts is a key challenge in weakly supervised semantic segmentation [25, 125, 133, 139, 238]. For the semi-supervised method U2PL [242], we observe that the error rates decrease rather uniformly for all types as the number of supervised samples increases. Thus, we conclude that using more supervised samples for training does not resolve only specific error types, but it is beneficial for all of the three proposed error categories.

			$\mathbf{mE_\star oU}$			$\widetilde{\mathbf{mE_\star oU}}$		
Setting	**Architecture**	**mIoU**	*bnd*	*ext*	*seg*	*bnd*	*ext*	*seg*
Full sup.	DeeplabV3+ [29]	78.6	3.4	9.8	8.1	4.7	11.4	8.1
Weak sup. ($\mathcal{I}$)	BECO [202]	70.8	4.1	18.0	7.1	6.1	20.1	7.1
Semi-sup. (1/16)	U2PL [242]	68.0	4.6	14.6	12.8	8.1	18.0	12.8
Semi-sup. (1/8)	U2PL [242]	71.4	3.9	12.1	12.6	5.8	15.9	12.6
Semi-sup. (1/4)	U2PL [242]	74.8	4.4	11.3	9.5	6.1	13.4	9.5
Semi-sup. (1/2)	U2PL [242]	78.4	3.7	9.9	8.0	5.0	11.6	8.0
Open vocabulary	ODISE [259]	83.9	1.5	6.5	8.1	2.0	7.3	8.1
Open vocabulary	TCL [22]	51.1	7.2	29.9	11.7	13.8	36.1	11.7

Table 7.4: Results on PascalVOC 2012 for different learning settings. Number in parentheses for semi-sup. indicates the proportion of labeled samples (see [242]).

In addition to these weakly and semi-supervised approaches, we also assess two representatives of open-vocabulary semantic segmentation. Although ODISE [259] achieves a strong $mIoU$ of 83.9%, its segment error rate of 8.1% is not lower than the one of the fully supervised DeepLabV3+ (8.1%) and the one of the weakly supervised BECO (7.1%). Also, TCL [22] produces many segment errors (11.7%). However, since TCL is supervised with only image-text pairs, it receives no localization information during training, leading to high boundary and extent errors. Altogether, the observed segment error rates indicate that, to this date, closed-vocabulary methods are stronger in terms of classification than open-vocabulary methods.

Comparing Error Distributions across Classes. Since our proposed error categorization considers each semantic class separately, one can analyze the error distribution of each class individually. We visualize a subset of the error distributions per class for PSPNet + R50-D8 on COCO-Stuff 164k in Figure 7.5. Looking at the plot, we can see a substantial difference between classes. In particular, the class "fog" exhibits an error distribution that differs substantially from those of other classes, having an extremely high segment error rate, while boundary and extent errors are exceptionally low. We can derive that the model often fails to recognize the class "fog", but when it does, it can segment it remarkably well. To investigate this observation further, we visually inspect the COCO-Stuff dataset and find a label ambiguity between "fog" and "clouds", for which we provide examples in Figure 7.6. Thus, the insights derived by the application of our error analysis indicate a problem in the annotations process, but not on the model side.

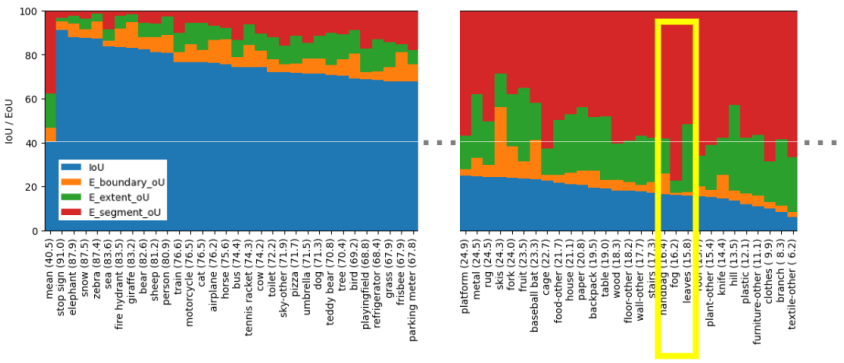
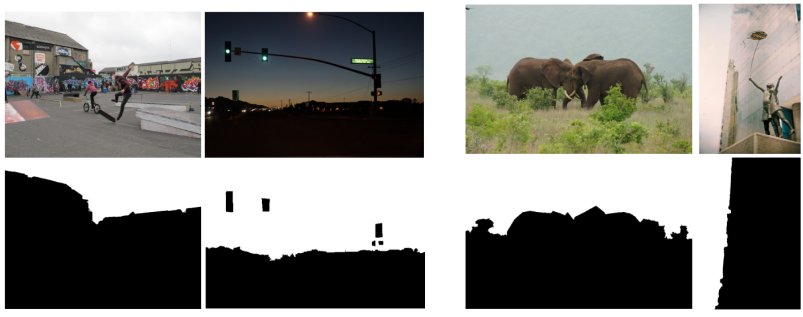


Figure 7.5: IoU and EoUs per class on COCO-Stuff 164k. The class "fog" (highlighted) stands out with high segment errors and low boundary and extent errors.



(a) Fog **(b) Clouds**

Figure 7.6: Two examples from the COCO-Stuff 164k dataset for the classes "fog" and "clouds", along with their corresponding binary ground-truth masks, illustrate the lack of a clear distinction between these categories in the dataset annotations.

Analyzing the Effect of Training Schedules. We examine the effect of training schedules on the example of PSPNet on COCO-Stuff 164k in Table 7.5. Overall, a longer schedule improves the $mIoU$. Looking at the error rates, we can see that this improvement is mainly caused by fewer segment and extent errors, whereas the number of boundary errors remains relatively stable. This suggests that the low-level cues needed to segment boundaries are learned rather early, while high-

		$\mathbf{mE_\star oU}$			$\widetilde{\mathbf{mE_\star oU}}$		
Schedule	**IoU**	*bnd*	*ext*	*seg*	*bnd*	*ext*	*seg*
80k	38.8	5.9	16.3	39.0	16.8	32.8	39.0
160k	39.6	5.9	16.2	38.2	17.0	32.2	38.2
320k	40.5	6.0	15.6	37.8	16.3	30.9	37.8

Table 7.5: Comparing different training schedules for PSPNet + R50-D8 on COCO-Stuff 164k.

			$\mathbf{mE_\star oU}$			$\widetilde{\mathbf{mE_\star oU}}$		
Model	**TTA**	**IoU**	*bnd*	*ext*	*seg*	*bnd*	*ext*	*seg*
PSPNet		44.4	9.0	15.9	30.7	19.2	25.2	30.7
PSPNet	✓	45.1	8.0	15.6	31.3	17.2	24.6	31.3
DeepLabV3+		45.5	9.0	15.8	29.8	18.8	24.1	29.8
DeepLabV3+	✓	46.0	7.9	14.9	31.2	17.3	23.9	31.2

Table 7.6: Analyzing the effects of test-time augmentation (TTA) on ADE20K. All models use R101-D8 as a backbone.

level semantic features needed for classification and capturing extents are better learned with more training iterations.

Analyzing the Effect of Test-time Augmentation. We also investigate the effect of test-time augmentation (TTA) on the distribution of error types on the ADE20K dataset. Specifically, TTA involves applying transformations such as flipping and multi-scale resizing to input images. The results are provided in Table 7.6. The results, summarized in Table 7.6, reveal that TTA primarily reduces boundary and extent errors. This improvement can be attributed to the inherent uncertainty in predictions near class boundaries. By aggregating predictions across multiple augmented views of the input, TTA effectively mitigates this uncertainty, leading to more accurate segmentation outcomes.

Surprisingly, we also observe an increase in segment errors when using TTA. In fact, a closer inspection of the error statistics reveals that only the false negative segment errors $FN_{seg}oU$ increase (from 17.4% to 19.4% and from 15.5% to 18.1% for PSPNet and DeepLabV3+, respectively), while the false positive segment errors $FP_{seg}oU$ decrease (from 13.3% to 11.8% and from 14.2% to 13.1% for PSPNet and DeepLabV3+, respectively). This behavior can be explained by the averaging

of the TTA predictions, making the final predictions more consistent and less fragmented. However, more fragmented predictions increase the chance of true positive predictions for at least small parts of the ground-truth segments, which is why the predictions without TTA yield substantially lower values for $FN_{seg}oU$.

Boundary Errors vs. Boundary IoU. Given the potential confusion between the terms "Boundary IoU– [39] and "boundary errors" in our work, we clarify the distinction between these concepts. Boundary IoU measures the IoU on pixels that are close to the boundary but inside the ground-truth foreground ($G_d \cap G$ in the Boundary IoU paper, where G_d is the set of pixels in the boundary region of the ground-truth mask) as well as on pixels that are close to the boundary but inside the predicted foreground ($P_d \cap P$).

In contrast, our definition of boundary errors focuses on regions near both true positive and true negative boundaries. This approach introduces two fundamental differences: (1) instead of examining boundaries of the predicted and ground-truth masks, we analyze boundaries of true positives and true negatives, and (2) only the intersection of these boundary regions is considered relevant. This design enforces an intuitive constraint that boundary errors occur exclusively near correct predictions—where transitions between foreground and background are identified.

In other words, Boundary IoU measures overall segmentation quality by assessing proximity to boundaries, while our boundary error metric serves as a targeted indicator of segmentation quality specifically along accurately recognized boundaries. Additionally, our boundary error formulation includes adjustments to mitigate undesirable effects (refer to Equations 7.2 and 7.3), extending the maximum relevant distance from boundaries to $2d$, compared to d in Boundary IoU. These refinements enhance its robustness as a fine-grained evaluation metric for boundary-sensitive segmentation tasks.

7.2.5 Qualitative Analyses

Visualization of Error Categories. In Figure 7.7, we present a visualization of the ground truth, predictions, and error types for selected classes from a sample in ADE20K. Notably, the visualized error categories align well with their intended definitions: boundary errors are localized near transitions between foreground and background regions, extent errors arise when predicted segment boundaries are significantly misplaced, and segment errors occur when entire segments are incorrectly predicted. This correspondence underscores the intuitive validity of the error categorizations.

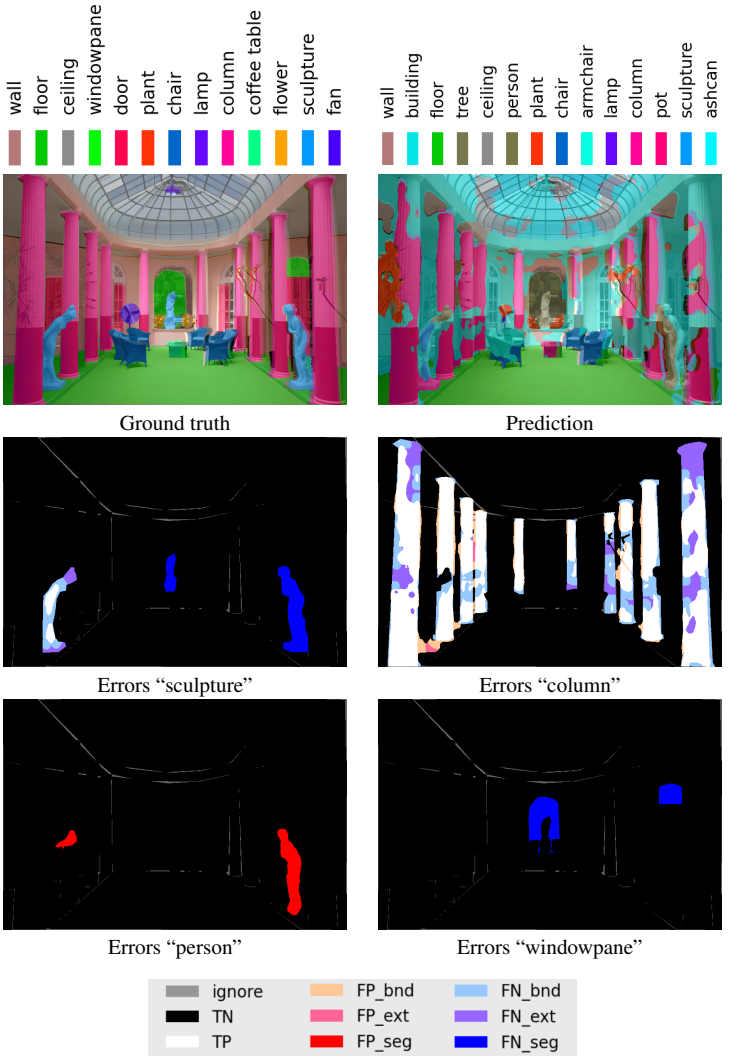
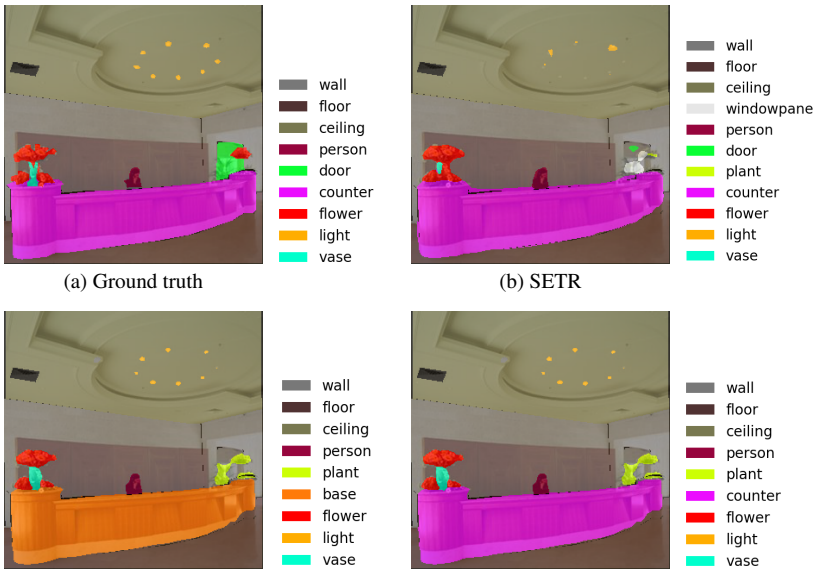


Figure 7.7: Visualization of error types for a sample from ADE20K with DeepLabV3+.



(a) Ground truth (b) SETR

(c) Mask2Former (d) Mask2Former ∘ SETR

Figure 7.8: Qualitative Example from ADE20K for SETR + ViT-L, Mask2Former + R101-D8, and their combination. The combined prediction (d) preserves the fine detail of Mask2Former (lamps, vase, flowers) as well as the superior segment classification of SETR (counter).

Qualitative Comparison of Single and Combined Models. Figure 7.8 illustrates qualitative results for SETR + ViT-L, Mask2Former + R101-D8, and their combination. The results highlight that Mask2Former excels in delineating precise segment boundaries, while SETR demonstrates superior performance in object and region classification. These observations are corroborated by the predictions visualized in the figure. As discussed in Section 7.2.3, by integrating these two models, the goal is to harness their respective strengths while compensating for their limitations. Subfigure (d) showcases the efficacy of this combination: it retains SETR's strong classification capabilities (e.g., correctly identifying the counter in the image) alongside Mask2Former's ability to capture fine-grained details (e.g., accurately segmenting the lamps, vase, and flowers). This synergy highlights the potential of our proposed error categorization framework to effectively combine complementary segmentation architectures.

7.2.6 Proof of the Disjointness of the Error Categories

Assigning each erroneous pixel to a single error category is straightforward to implement, but demonstrating that the three error categories defined in Section 7.1 are mutually exclusive is less intuitive. To substantiate this claim, we provide formal proof.

- $E_{bnd} \cap E_{ext} = \emptyset$: Trivial by definition.
- $E_{ext} \cap E_{seg} = \emptyset$: Also trivial by definition.
- $E_{bnd} \cap E_{seg} = \emptyset$: Suppose there is a pixel

From Equation 7.3, we can deduce that there is an $x' \in \mathcal{S}(FP''_{bnd})_x$ such that $\mathcal{N}_1(x') \cap TP \neq \emptyset$. Let x'' be a TP pixel in $\mathcal{N}_1(x')$. Then, we know that $x'' \in \mathcal{S}(P)_x$ as x'' is a direct neighbor of $\mathcal{S}(FP''_{bnd})_x$. With $x'' \in \mathcal{S}(P)_x$, we obtain $\mathcal{S}(P)_x \cap TP \neq \emptyset$, contradicting $x \in E_{seg}$ and, therefore, concluding the proof. Additionally, it follows directly from their definitions that every erroneous pixel must belong to at least one of these error categories.

Chapter 8

Multimodal Deepfake Detection

In Chapter 6, we explored the potential of Transformers in multimodal learning, highlighting their ability to bridge vision and language for tasks like open-vocabulary segmentation. Transformers have emerged as powerful tools for multimodal alignment, enabling models to capture complex semantic relationships across diverse domains. This ability has proven particularly valuable in addressing challenges at the intersection of vision and language, such as deepfake detection.

The rise of text-to-image generative models, powered by attentive architectures like diffusion models [61, 193, 201, 206], has created new challenges in content authenticity verification. While image generation tools can be employed for lawful goals, such as assisting content creators, generating simulated datasets, or enabling multimodal interactive applications, they have raised concerns regarding their potential for illegal and malicious purposes [13, 43, 94]. These include the forgery of natural images, the generation of images in support of fake news, and the generation of NSFW contents [190, 208]. In this context, assessing the authenticity of images becomes a fundamental goal for security and for guaranteeing the trustworthiness of AI algorithms.

Most of the past approaches for deepfake detection have employed perceptual

This chapter is related to the publication "Roberto Amoroso*, Davide Morelli*, Marcella Cornia, Lorenzo Baraldi, Alberto Del Bimbo, and Rita Cucchiara, Parents and Children: Distinguishing Multimodal DeepFakes from Natural Images, TOMM 2024 (* Equal Contribution)".

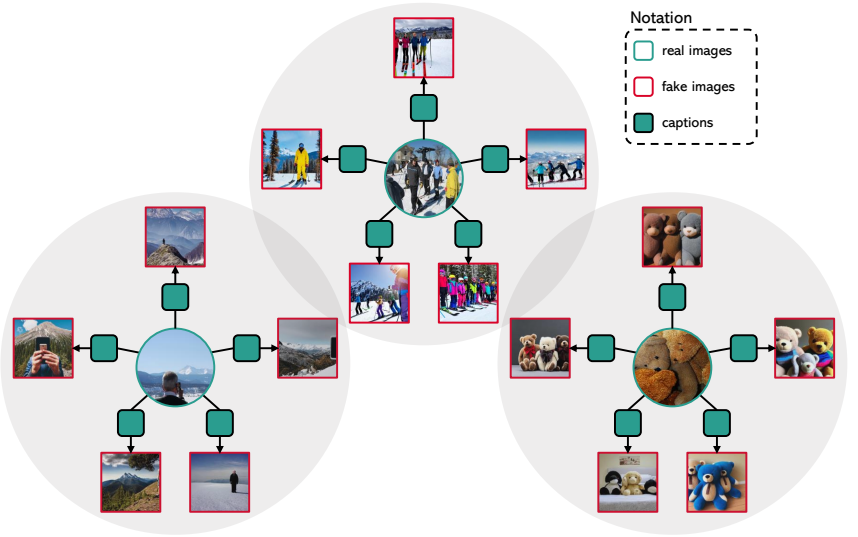


Figure 8.1: Overview of our multimodal deepfakes detection setting, in which five subsets of the semantics contained in a given image are employed to generate as many fake images.

cues [68, 78, 291], including frequency analysis, the detection of artifacts, or pixel discontinuities. Furthermore, a significant portion of the early studies has focused exclusively on fake faces [147, 148, 204]. Today's generators [63, 80, 193, 194, 201, 206] are general-purpose, text-driven, and exhibit higher generation quality. If we look at images generated by Stable Diffusion [201] (a few examples are reported in Section 8.1.5), we might notice that some of them appear hyper-realistic and, thus, easily recognizable, while others contain semantic anomalies. However, most of them are realistically plausible.

Contributions. In this chapter, we aim at developing a systematic study on deepfake detection, in an era when generated content is becoming increasingly realistic and text-driven. We do this in a multimodal setting that enables us to examine deepfake detection from both a perceptual and a semantic perspective. Specifically, given an image, we consider different textual descriptions and fake images generated by using each of the descriptions as a prompt (Figure 8.1). In this manner, we build clusters sharing similar semantics, containing one real image

and multiple fake images. Under this setting, we first train a classifier to recognize deepfakes and investigate the effectiveness of different visual features extracted from both contrastive-based backbones like CLIP [191] and classification-based ones such as ResNet [99] and ViT-based networks [67] trained on ImageNet. Surprisingly, we find out that high-level contrastive-based features learned on image and text pairs are very effective in discriminating between real and generated images. We hypothesize that low-level perceptual features also percolate into such descriptors, even though they are trained at a semantic level.

While these findings might be effective in defending us from current generators, we can expect that tomorrow's generators will increase their quality and become less detectable via low-level features. Thus, we devise a contrastive-based disentanglement strategy that enables to remove the contribution of low-level features. This approach establishes a more complex setting in which generated images cannot be distinguished at a perceptual level. Under this setting, we propose and discuss a general procedure for discriminating between fake and real images based on semantic information. To evaluate the effectiveness of the proposed method, we introduce a new dataset, namely COCOFake, which comprises approximately 1.2M images generated from the original COCO image-caption pairs using both Stable Diffusion v1.4 and v2.0 as text-to-image generative models.

The rest of the chapter is organized as follows: Section 8.1 introduces the proposed framework for multimodal deepfake detection, detailing its key components, including the disentanglement of semantic and style features using contrastive learning, and the creation of a novel dataset, COCOFake, tailored for this task. Section 8.2 provides a comprehensive evaluation of the methodology, encompassing experiments on discriminative power, semantic preservation, and semantic-style disentanglement. It also includes robustness analyses against image transformations and comparisons with existing deepfake detection methods.

8.1 Proposed Method

This section introduces our proposed framework for studying and detecting multimodal generated fake images, which encompasses the identification and separation of their perceptual and semantic components. First, we establish the foundational notation and preliminaries, defining key concepts such as real images, fake images, and their semantic relationships. The method then explores a discriminative analysis to evaluate the ability of pre-trained models to distinguish real from fake images, and a semantic preservation analysis to assess whether captions'

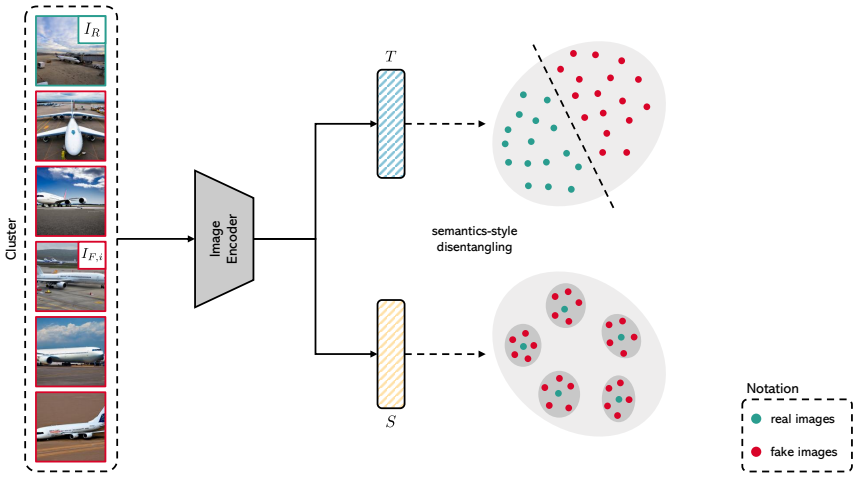


Figure 8.2: Schema of our approach for disentangling semantics and style for deepfake detection.

semantic information is retained in generated images. To address challenges in detection, a novel approach is proposed to disentangle semantics and style, using contrastive learning to separate these components for robust classification, as illustrated in Figure 8.2. Finally, we introduce COCOFake, a multimodal dataset specifically designed for this task, containing real images paired with multiple fake counterparts generated using Stable Diffusion.

8.1.1 Notation and Preliminaries

In the rest of the chapter, we will employ the following notation: I_R will indicate a natural (real) image, C a textual description (*i.e.*, a caption), and I_F will indicate a fake image produced by a generator. Under this setting, a *parent* real image I_R can be the seed for N different *children* fake images $I_{F,i}$ given a set of textual descriptions $\{C_i\}$ of I_R, with $i = 1, ..., N$, by using each of the descriptions as prompt for the generator.

Semantic and Style Components of an Image. The information content of an image can be credited to many factors. For simplicity, we assume that an image I, regardless of its authenticity, embodies two information contributions, namely a

semantic component $\mathcal{H}_{sem}(I)$ and a perceptual or *style component* $\mathcal{H}_{sty}(I)$. The former represents the content that could be expressed in a textual sentence, while the latter describes the image appearance, encompassing elements such as colors, textures, brightness, and low-level visual cues. Given a real image I_R, we can therefore express its total information $\mathcal{H}$ as a function of its semantic and style components, as follows:

$$\mathcal{H}(I_R) = f(\mathcal{H}_{sem}(I_R), \mathcal{H}_{sty}(I_R)). \tag{8.1}$$

However, when an image is described through a natural language sentence, only a portion of its semantics is actually conveyed inside the caption. In other words, natural language descriptions act as a filter for the semantic content of the image. Hence, we introduce $\Delta\mathcal{H}_{sem}(I, C)$ to represent the portion of semantic information described by a caption C. By analogy, we could say that the textual descriptions of an image act as DNA fragments that can be utilized to generate an offspring of images.

Generating Offspring with Natural Language Utterances. From an input image I_R we can extract N semantic information subsets $\Delta\mathcal{H}^i_{sem}(I_R, \cdot)$ and feed them to a generator obtaining N different fake images $I_{F,i}$, with $i = 1, ..., N$. We define *semantic cluster* the ensemble of the starting real image I_R and the offspring of N fake images $I_{F,i}$ generated from it. For instance, given a dataset such as COCO [153], containing K real images, each represented by $N = 5$ captions, we could create K clusters of $N + 1$ images with one parent and N children.

8.1.2 Learning to Discriminate Real and Fake images

Once a dataset in the aforementioned form has been built, we first measure to what extent real and generated images can be discriminated independently from their membership to a semantic cluster. Instead of doing this by learning ad-hoc visual features, we investigate the usage of state-of-the-art pre-trained visual models. In other words, given a dataset containing both real and generated images, we develop a model that identifies real images by using visual features extracted with a pre-trained backbone. Regarding the generation of the images, in the following, we will employ Stable Diffusion [201], which is freely available and represents a state-of-the-art approach. Nevertheless, the approach could be easily extended to other generators.

To evaluate the discriminative power of current pre-trained visual features, we model the discriminator as a two-class linear classifier, so that input visual

features are only linearly projected before taking the final decision on their realism. Formally, given a real image $I_R \in \mathbb{R}^{3\times H\times W}$ and an image encoder $E_I : \mathbb{R}^{3\times H\times W} \rightarrow \mathbb{R}^D$, we extract a vectorial image feature F_I as

$$F_I = E_I(I_R). \tag{8.2}$$

The features F_I are then fed into a linear layer $L : \mathbb{R}^D \rightarrow \mathbb{R}$, whose output is thresholded to classify between *real* (*i.e.*, 0) and *fake* (*i.e.*, 1) images. As it will be discussed in the experimental section, our findings indicate that this is (still) a relatively simple task even when employing a state-of-the-art generator. This is, most likely, due to the fact that fake images are slightly different in terms of low-level cues with respect to real images.

8.1.3 Semantic Preservation Analysis

As a second analysis, we investigate the preservation of semantic information across both real and generated fake images. To do so, we consider a multimodal embedding space, in which both images and texts can be projected. Specifically, we verify if, starting from a generated image, we can retrieve the particular caption used as prompt during its generation. In other words, we test if the subset of the real semantic information $\Delta\mathcal{H}_{sem}(I_F, C)$ associated with a caption C is still recognizable in the visual features extracted from the generated image.

Formally, given a caption C describing a real image I_R, and a textual encoder E_T, we tokenize and extract the textual features F_T as:

$$F_T = E_T(C). \tag{8.3}$$

For each visual feature of a given fake image I_F, we verify the ability to retrieve the corresponding textual feature used to create I_F through the generator model.

As it will be shown in the experimental section, we find out that (a) the alteration of low-level cues induced by the generator does not affect the semantic contribution coming from the original image, and (b) the semantic contribution of the generator does not obfuscate the original semantic content.

8.1.4 Disentangling Semantics and Style

As the detection of fake images is likely promoted by the difference in low-level cues between generated and real images, we finally investigate a more challenging setting in which the style component induced by the generator is

disentangled and removed. To do so, we learn a model which identifies the style component of the generator which is common to all generated images. We then measure whether, after eliminating such a component, the remaining semantic information is sufficient to discriminate between real and fake images. Noticeably, this corresponds to a more challenging setting where all the common low-level traits left by the generator are removed and not employed to perform deepfake classification. In other words, this also corresponds to recognizing fakes generated by an "ideal" generator that does not leave common low-level traits.

To perform this analysis, we propose a new contrastive-based learning model that can project images in a semantic space and in a style space (Figure 8.2). For a good style-semantic disentanglement we expect that, in the style embedding latent space, the feature vectors of real images should be separated from features of fake images in a cluster-agnostic way, while in the semantic embedding latent space the cluster compactness should be preserved. Specifically, we train two separate linear projections T and S, where T focuses on style while S on semantics. For the T layer we aim at increasing the distance between fake and real elements, regardless of their membership in a specific cluster. For the S layer, instead, we want to create compact clusters of elements sharing the same semantic content, while increasing the distance among two fake elements or two real elements.

We express these requirements through two loss components $\mathcal{L}_c$ and $\mathcal{L}_{fr}$. The former attracts elements of the same cluster, while the latter attracts elements having the same label (*i.e.*, real and fake). From here, we can define the losses needed to train T and S, respectively, as follows:

$$\begin{aligned} \mathcal{L}_T &= \mathcal{L}_{fr} - \mathcal{L}_c, \\ \mathcal{L}_S &= \mathcal{L}_c - \mathcal{L}_{fr}. \end{aligned} \tag{8.4}$$

To implement both $\mathcal{L}_c$ and $\mathcal{L}_{fr}$, we leverage a Supervised Contrastive Loss [124], defined as follows:

$$\mathcal{L}_{SupCon} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\mathcal{F}_i \mathcal{F}_p^\intercal / \tau)}{\sum\limits_{a \in A(i)} \exp(\mathcal{F}_i \mathcal{F}_a^\intercal / \tau)}, \tag{8.5}$$

where $i \in I \equiv \{1, ..., N+1\}$ represents the index of an arbitrary sample, $\mathcal{F}$ are ℓ_2-normalized input features of a given image, τ is a temperature parameter, $A(i) \equiv I / \{i\}$. $P(i)$ is the set of indices of all items sharing the same label of i, and $|P(i)|$ is its cardinality.

Depending on the nature of the labels used in the training of the supervised contrastive loss, we can implement repulsive and attractive forces in the form of

the loss components $\mathcal{L}_c$ and $\mathcal{L}_{fr}$. In $\mathcal{L}_c$, in particular, we assign the same label to elements belonging to the same cluster, while in $\mathcal{L}_{fr}$ we assign the same label to all real samples, and the same label to all fake images. The objective of $\mathcal{L}_c$ is to attract elements of the same cluster, while $\mathcal{L}_{fr}$ pushes real and fake images.

8.1.5 COCOFake: a Multimodal Deepfake Recognition Dataset

In literature, to the best of our knowledge, there are no multimodal datasets containing texts, real and fake images that are compatible with our multimodal setting. Thus, we generate and release the COCOFake dataset, an extension of COCO [153]. Each real image in COCOFake is paired with five fake images that are conditionally generated based on each of the captions associated with the same image. We employ the Stable Diffusion model [201] as our generator. Specifically, we create two different versions of our dataset, one based on Stable Diffusion v1.4 and the other based on Stable Diffusion v2.0. Both text-to-image generators have been pre-trained on the English image-text pairs of the LAION-5B dataset [209] and finetuned on the LAION-Aesthetics subset. While Stable Diffusion v1.4 is based on the CLIP ViT-L/14 text encoder [191], the 2.0 version exploits the OpenCLIP ViT-H/14 one [191]. During image generation, we employ the safety checker module to reduce the probability of explicit images and disable the invisible watermarking of the outputs to prevent easy identification of the images as machine-generated.

Overall, referring to the splits defined in [118] and typically employed in image captioning literature [6, 17, 207], the COCO dataset comprises 113,287 training images, 5,000 validation, and 5,000 test images. Preserving the same splits, COCOFake is composed of 679,722 training images, 30,000 validation, and 30,000 test images for each version of Stable Diffusion, thus comprising more than 1.2M generated images (*i.e.*, around 600k for each version of Stable Diffusion). Sample real-generated image clusters from the COCOFake dataset are shown in Figure 8.3. For each example, we present the real image alongside the five fake images generated from each of the five captions from the original COCO dataset. As it can be seen, the generated images are generally coherent with the corresponding caption. However, in some cases, the generated images are overly realistic with brighter colors and a more professional photographic style than the real counterpart. This can be attributed to the dataset employed in the finetuning phase (*i.e.* the LAION-Aesthetics subset) of the Stable Diffusion model [201], used to generate fake images. In Figure 8.4 we report less realistic examples from the COCOFake dataset, again showing the original image and the five fake images

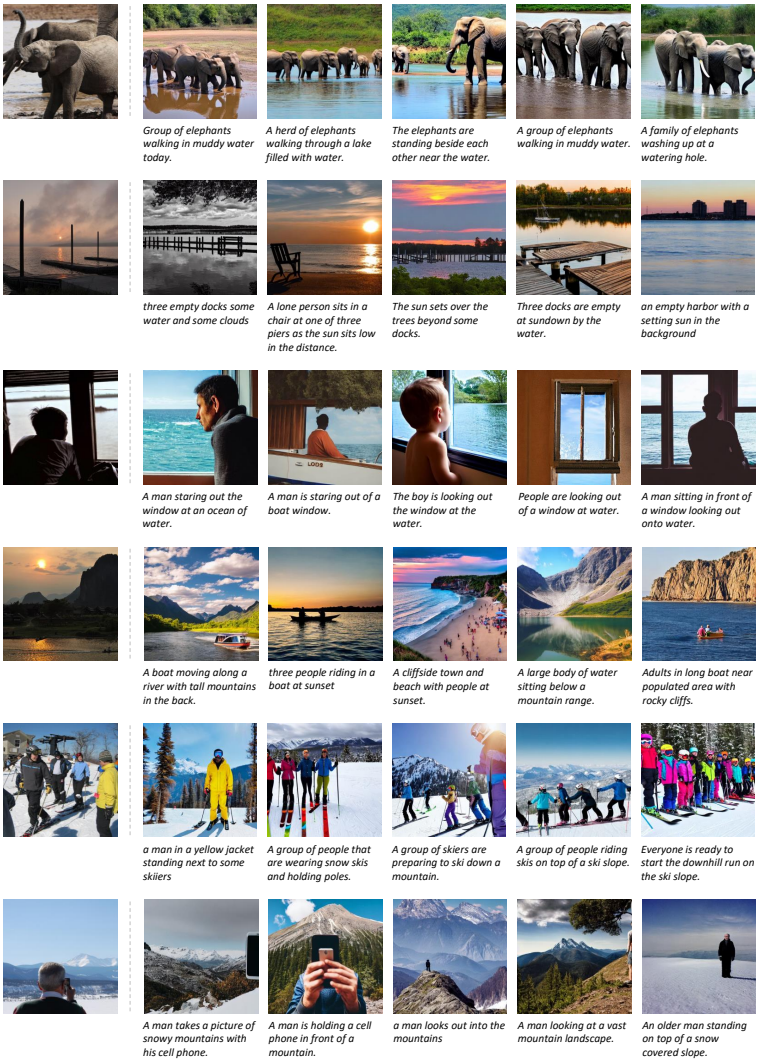


Figure 8.3: Sample images from COCOFake. The leftmost column shows the original (real) image, while the remaining ones show fake images generated by Stable Diffusion v1.4 from each of the five COCO captions.

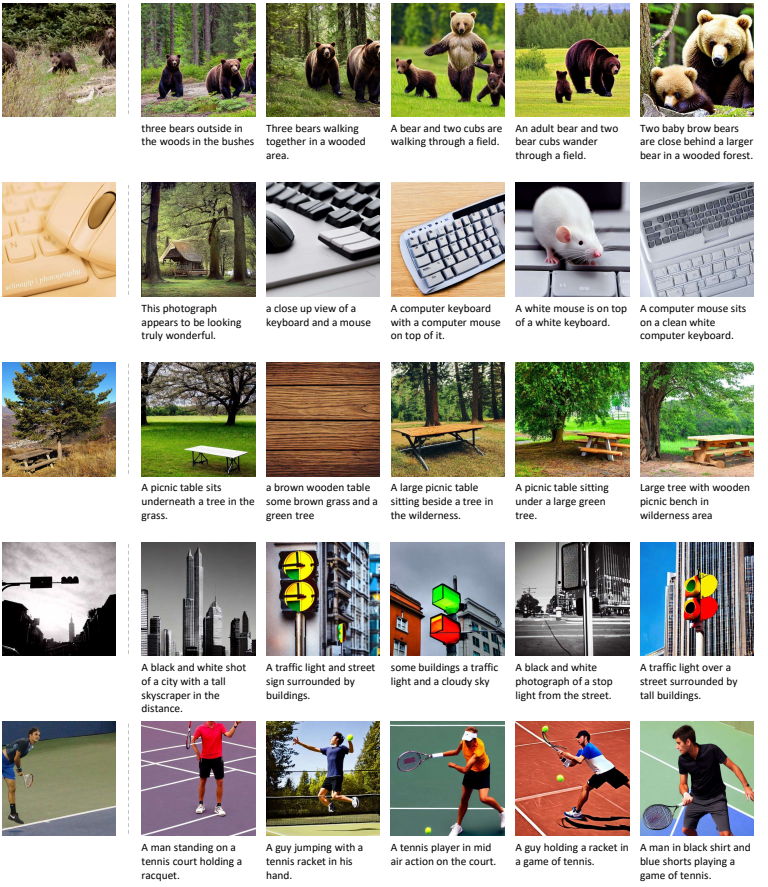


Figure 8.4: Less realistic images from COCOFake. The leftmost column shows the original (real) image, while the remaining ones show fake images generated by Stable Diffusion v1.4 from each of the five COCO captions.

with the corresponding captions. Failure cases include hallucinating the semantic content of the caption (first two rows), incorrect understanding of the caption (third row), abstract rendering of objects (traffic lights in the third row), and unrealistic rendering of human poses (last row).

In our experiments, we evaluate deepfake detection performance under a standard setting in which we train the model on images generated by one Stable Diffusion version and test on images generated by the same model. Furthermore, to assess the robustness of our analysis, we also consider the generalization capabilities to images generated by a text-to-image diffusion model different from the one used during training. Under this setting, we compare the performance of our method on images generated by different versions of Stable Diffusion, providing insights into the impact of the generative model on the deepfake detection performance.

8.2 Experiments

This section provides a comprehensive evaluation of the proposed methodology, assessing its performance across diverse tasks using various visual backbones, metrics, and experimental setups. It begins by detailing the implementation specifics, including the choice of image encoders, training configurations, and evaluation metrics designed to disentangle semantic and stylistic information. The experiments explore the performance of visual features in unsupervised classification and linear probing tasks, while a retrieval-based analysis further examines the preservation of semantic information during image generation. The proposed semantic-style disentanglement approach is also evaluated. Additionally, robustness analyses investigate the impact of common image transformations on the model's performance. Finally, a comparative study with existing deepfake detection methods underscores the superior performance of the proposed approach.

8.2.1 Implementation Details

Image Encoders. We test two families of backbones: the first are trained for classification on ImageNet [205], while the second are trained on a cross-modal setting on large-scale datasets using contrastive-based loss functions. Due to the nature of the task these networks were trained for, only the latter family provides also text encoders E_T.

Specifically, we employ a ResNet [99] model with 48 convolutional layers and a Vision Transformer (ViT) [67] architecture in its B/32 configuration. The ViT encoder takes as input squared patches extracted from the input image and consists of a sequence of multi-head self-attention layers [230]. Both these architectures are trained on the ImageNet dataset [205] that contains around 1.3M images.

As cross-modal architectures, we use two models coming from CLIP [191]. In particular, we employ CLIP RN50 and CLIP ViT-B/32 models, both pre-trained on the OpenAI WebImageText (WIT) dataset, composed of 400 million image-text pairs collected from the web. Moreover, we employ the open source implementation of CLIP (*i.e.*, OpenCLIP [248]), trained with a post-ensemble method for improving robustness to out-of-distribution samples. In our experiments, we consider two versions of the OpenCLIP ViT-B/32 model: one trained on the LAION-400M dataset [210] that contains 400 million CLIP-filtered image-text pairs crawled from the web and the other trained on the larger LAION-2B composed of 2 billion image-text pairs [209].

Linear Probing Details. In our experiments, we also conduct linear probes. In this case, we follow the approach of [191] and employ the features extracted from the backbones to train a logistic regression model with ℓ_2 penalty and LBFGS solver [15, 307]. To balance the training samples, we employ one randomly extracted fake image for each cluster.

Disentanglement Architecture and Training Details. When disentangling semantics and styles, we train the two linear layers S and T, which perform a linear projection to the same dimensionality of the backbone visual features. To train these layers, we employ AdamW [165] as optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We use a batch size of 1,024 and a learning rate of 0.001, training all models for 25 epochs.

8.2.2 Metrics

To assess the performance of our proposed methodology and evaluate spatial relationships between elements in the embedding spaces, we employ seven different metrics. These aim to quantify the capability to discriminate between real and fake images and to quantify disentanglement.

Min and Max Intra-Cluster Distance Accuracy. These two metrics are employed to evaluate the relative spatial positions of the elements inside a cluster. In particular, for each cluster, we measure the distances between the real image and each of the fake images belonging to the cluster. We then check how many times the real image is the item having the minimum or maximum distance with respect to all the others in the cluster. In other words, for each cluster, the min distance accuracy scores if the real image feature is on average the nearest to all the fake image features, while the max distance accuracy scores if it is the most distant one.

Overall and Full Cluster Accuracy. These two metrics measure the real/fake classification accuracy both over the entire dataset and inside each cluster. The former metric is cluster-independent and is computed using all the elements of a dataset split (*i.e.*, validation, test). The latter, instead, is a cluster-based metric that scores if all elements of a cluster are correctly classified as real or fake, and the metric is then averaged across all clusters.

Overall AUC. As reported in previous deepfake detection literature [51, 174], this metric is used along with accuracy to evaluate how well a deepfake detection model can distinguish between real and fake images. In our setting, it is computed using all the elements of the validation or test set of our dataset.

Exact Pair and Intra-Cluster Retrieval. These metrics are used to evaluate the goodness of the retrieval task (see Sec. 8.1.3), in which given a generated image we seek to retrieve its parent caption. The former metric is a recall@k computed considering as ground-truth, for each fake image, the caption used for generating it. The latter is a recall@k that measures for a given fake image if the retrieved caption matches one of the five captions of the cluster the image belongs to.

8.2.3 Performance of Visual Features

Unsupervised Classification. We start by assessing the capabilities of existing image features to discriminate between real and generated images, in an unsupervised setting. We employ the min and max distance accuracy metrics defined above and check the presence of spatial relationships between real and generated images inside each cluster.

Results are reported in Table 8.1 on the test and validation sets of both Stable Diffusion v1.4 and v2.0. We employ six different visual backbones, namely two ResNet-50 pre-trained on ImageNet and OpenAI WIT and four ViT-B/32 pre-trained on ImageNet, OpenAI WIT, LAION-400M, and LAION-2B. As it can be seen, according to the features extracted from the aforementioned backbones, the real image of each cluster tends to be the one with maximum distance with respect to all the other elements. This suggests that these features are discriminative for the task of deepfake classification and that they percolate low-level features that allow for distinction between real and generated items inside of each semantic cluster. Noticeably, the maximum distance accuracy increases when considering backbones trained on multimodal datasets compared to backbones trained on classification, suggesting that image-text matching promotes the percolation of perceptual features.

		Validation Set (SD v1.4)		**Test Set (SD v1.4)**		**Validation Set (SD v2.0)**		**Test Set (SD v2.0)**	
Backbone	**Dataset**	Min Dist. Accuracy	Max Dist. Accuracy	Min Dist. Accuracy	Max Dist. Accuracy	Min Dist. Accuracy	Max Dist. Accuracy	Min Dist. Accuracy	Max Dist. Accuracy
RN50	ImageNet	8.50	23.58	8.82	24.82	5.98	29.62	6.62	30.16
ViT-B/32	ImageNet	6.84	23.12	6.88	23.88	5.12	29.18	4.92	30.00
CLIP RN50	OpenAI WIT	3.72	38.48	3.60	41.24	2.40	46.72	2.20	48.28
CLIP ViT-B/32	OpenAI WIT	3.30	38.88	3.24	40.10	2.92	42.08	2.98	44.18
OpenCLIP ViT-B/32	LAION-400M	5.28	31.94	5.00	32.02	4.58	34.06	4.62	36.02
OpenCLIP ViT-B/32	LAION-2B	1.40	42.80	1.72	44.00	1.88	42.64	1.78	43.80

Table 8.1: Minimum and maximum distance accuracy on validation and test sets of COCOFake, using different visual backbones. Results are reported using images generated by both Stable Diffusion v1.4 and v2.0.

When comparing the performance of synthetic images generated by the two versions of Stable Diffusion under consideration, it can be noticed that Stable Diffusion v2.0 exhibits an improvement over v1.4 as evidenced by an increase in the maximum distance metric and a decrease in the minimum distance metric. These changes indicate that the features extracted from images generated by v2.0 are more distinct and separable, making them easier to detect.

Linear Probing. Following the approach popularized by [191], we train a linear projection through logistic regression on top of the features extracted from the aforementioned backbones. We perform this experiment by training on both Stable Diffusion v1.4 and v2.0 images and testing either on the validation and test sets containing images generated by the same Stable Diffusion version used during training or on the validation and test sets containing images generated by the Stable Diffusion model not used to train the linear projection.

The results, detailed in Table 8.2, are presented in terms of overall accuracy and full cluster accuracy. As can be seen, all the selected visual features exhibit a significant capability in linearly discriminating real and fake images, on the validation and test sets of the COCOFake dataset, regardless of whether the images were generated by Stable Diffusion v1.4 or v2.0. In continuity with the previous experiment, we observe that contrastive-based visual backbones showcase significantly higher accuracy levels, up to 98.01% and 97.16% of full cluster accuracy respectively on the validation set with Stable Diffusion v1.4 and v2.0 images, and up to 99.68% and 99.52% overall accuracy on the same split. This further confirms the observation that contrastive-based backbones extract and project into their embedding space, low-level and perceptual features that allow discriminating current deepfakes.

To assess the robustness of the method, we further test the trained classifiers on the data generated by the Stable Diffusion model not used during training (*i.e.*, Stable Diffusion v2.0 for the linear projection trained on the 1.4 version, and Stable Diffusion v1.4 for the linear projection trained on the 2.0 version). As it can be observed in the right part of Table 8.2, the trained classifier performs comparably also in this setting with an overall accuracy close to or greater than 99% in all cases. In particular, training on Stable Diffusion v2.0 images generalizes slightly better on images generated by Stable Diffusion v1.4 than the opposite direction with 99.47% and 96.80% of overall and full cluster validation accuracy compared to 98.88% and 93.68% obtained when testing the linear projection trained on Stable Diffusion v1.4 images on the validation set with images generated by the 2.0 version. Overall, these experiments show that the pre-trained visual backbones exhibit high discrimination power when identifying deepfakes.

		Validation Set (SD v1.4 → SD v1.4)		**Test Set (SD v1.4 → SD v1.4)**		**Validation Set (SD v1.4 → SD v2.0)**		**Test Set (SD v1.4 → SD v2.0)**	
Backbone	**Dataset**	Overall Accuracy	Full Cluster Accuracy	Overall Accuracy	Full Cluster Accuracy	Overall Accuracy	Full Cluster Accuracy	Overall Accuracy	Full Cluster Accuracy
RN50	ImageNet	90.31	57.56	90.62	57.94	81.71	34.94	82.31	35.84
ViT-B/32	ImageNet	87.64	47.62	87.16	47.32	76.71	24.68	77.31	26.92
CLIP RN50	OpenAI WIT	99.07	94.60	99.17	95.30	93.54	69.08	93.74	69.64
CLIP ViT-B/32	OpenAI WIT	99.11	94.84	98.97	94.24	94.41	72.30	94.72	73.62
OpenCLIP ViT-B/32	LAION-400M	97.88	88.18	97.83	87.80	83.30	38.48	84.32	40.74
OpenCLIP ViT-B/32	LAION-2B	99.68	98.01	99.64	97.84	98.88	93.68	98.96	94.08
		Validation Set (SD v2.0 → SD v2.0)		**Test Set (SD v2.0 → SD v2.0)**		**Validation Set (SD v2.0 → SD v1.4)**		**Test Set (SD v2.0 → SD v1.4)**	
Backbone	**Dataset**	Overall Accuracy	Full Cluster Accuracy	Overall Accuracy	Full Cluster Accuracy	Overall Accuracy	Full Cluster Accuracy	Overall Accuracy	Full Cluster Accuracy
RN50	ImageNet	91.07	59.84	91.45	61.44	91.08	60.44	91.33	60.76
ViT-B/32	ImageNet	85.55	42.92	86.12	44.90	84.89	41.50	84.49	39.60
CLIP RN50	OpenAI WIT	98.67	92.56	98.68	92.60	98.57	91.94	98.66	92.48
CLIP ViT-B/32	OpenAI WIT	98.56	92.04	98.48	91.48	98.58	92.02	98.48	91.76
OpenCLIP ViT-B/32	LAION-400M	95.03	74.70	95.57	77.42	97.40	85.62	97.29	84.88
OpenCLIP ViT-B/32	LAION-2B	99.52	97.16	99.59	97.54	99.47	96.80	99.41	96.56

Table 8.2: Overall and full cluster accuracy results on the validation and test sets, using linear probing and features of different backbones trained on the COCOFake training set. Results are reported using images generated by both Stable Diffusion v1.4 and v2.0.

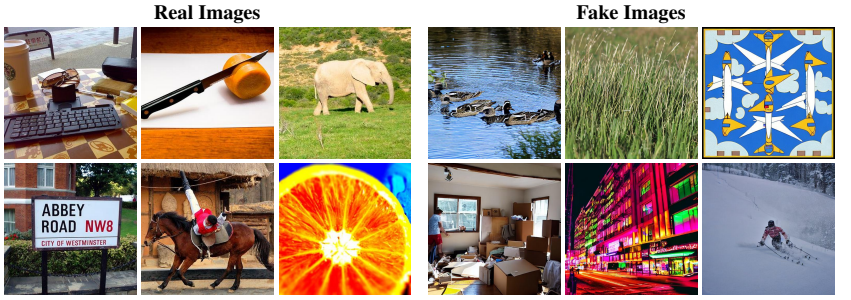


Figure 8.5: Sample misclassification errors on both real (left) and fake (right) images, using OpenCLIP ViT-B/32 trained on LAION-2B as the visual encoder.

In light of the high accuracy levels of the aforementioned experiment, in Figure 8.5 we report sample misclassified images. It can be noted, in particular, that fake images incorrectly classified as authentic (right side of the figure) depict close-ups and artistic drawings, whose authenticity is visually harder to guarantee.

Semantic Preservation. We then conduct the retrieval-based analysis anticipated in Sec. 8.1.3, in which we look for the original caption used to generate a image inside of a multimodal embedding space. The objective of this experiment is to assess whether the semantic information contained in the caption is preserved after the generation and to what extent the generation process alters semantic features.

Results are reported in Table 8.3, using the exact pair and intra-cluster retrieval metrics and considering validation and test sets containing Stable Diffusion v1.4 and v2.0 images. Surprisingly, retrieving the exact caption used to generate an image is not always easy, and the process is successful only in 40% of the cases when selecting a proper backbone. Even when considering all captions of the same clusters as positives, moreover, we observe a recall@1 of around 50%, again highlighting the difficulty of the task. The results are slightly higher when performing the experiment on the COCOFake version with Stable Diffusion v2.0 images, achieving 48.67% and 58.39% in terms of exact pair and intra-cluster retrieval on the validation set of the dataset. This suggests that Stable Diffusion v2.0 can generate images more semantically aligned with the corresponding captions than v1.4, probably due to its more powerful text encoder. Nonetheless, these results point out that current generators produce images with partially altered semantic features, and are also in line with the previous observation that contrastive-based extractors percolate low-level features.

		Validation Set (SD v1.4)						Test Set (SD v1.4)					
		Exact Pair			Intra-Cluster			Exact Pair			Intra-Cluster		
Backbone	**Dataset**	R@1	R@3	R@5	R@1	R@3	R@5	R@1	R@3	R@5	R@1	R@3	R@5
CLIP RN50	OpenAI WIT	31.33	49.05	56.93	41.91	58.46	66.01	30.98	48.38	56.42	42.09	58.35	65.93
CLIP ViT-B/32	OpenAI WIT	32.12	50.43	58.36	43.34	60.15	67.42	31.96	49.67	57.51	43.24	59.3	66.78
OpenCLIP ViT-B/32	LAION-400M	36.48	55.36	63.28	47.17	63.62	70.73	35.53	54.49	62.56	46.72	62.92	70.22
OpenCLIP ViT-B/32	LAION-2B	40.34	59.44	67.18	50.78	66.64	73.58	39.57	58.78	66.18	50.46	66.34	73.03
		Validation Set (SD v2.0)						**Test Set (SD v2.0)**					
		Exact Pair			**Intra-Cluster**			**Exact Pair**			**Intra-Cluster**		
Backbone	**Dataset**	R@1	R@3	R@5	R@1	R@3	R@5	R@1	R@3	R@5	R@1	R@3	R@5
CLIP RN50	OpenAI WIT	33.05	51.17	59.21	44.73	61.32	69.05	32.53	59.96	58.89	44.67	61.43	68.65
CLIP ViT-B/32	OpenAI WIT	34.70	53.48	61.31	46.73	.63.26	70.49	34.20	52.73	60.94	46.30	62.62	69.99
OpenCLIP ViT-B/32	LAION-400M	42.62	62.31	69.67	53.66	69.71	76.24	42.07	61.74	69.26	53.04	69.06	75.88
OpenCLIP ViT-B/32	LAION-2B	48.67	67.68	74.77	58.39	73.76	80.07	47.83	67.25	74.22	58.24	73.60	79.53

Table 8.3: Exact pair and intra-cluster retrieval results. Results are reported using images generated by both Stable Diffusion v1.4 and v2.0.

		Test Set (SD v1.4 → SD v1.4)					
Backbone	**Dataset**	Overall AUC S	Overall Accuracy S	Full Cluster Accuracy S	Overall AUC T	Min Dist. Accuracy T	Max Dist. Accuracy T
RN50	ImageNet	74.93	62.96	8.64	98.45	0.42	89.08
ViT-B/32	ImageNet	68.19	64.04	8.46	96.60	1.30	76.26
CLIP RN50	OpenAI WIT	80.73	74.76	21.40	99.87	0.00	98.46
CLIP ViT-B/32	OpenAI WIT	71.29	67.48	12.90	99.74	0.20	98.14
OpenCLIP ViT-B/32	LAION-400M	70.27	66.84	10.98	99.45	0.10	94.48
OpenCLIP ViT-B/32	LAION-2B	78.00	72.62	17.32	99.93	0.06	99.39
		Test Set (SD v1.4 → SD v2.0)					
Backbone	**Dataset**	Overall AUC S	Overall Accuracy S	Full Cluster Accuracy S	Overall AUC T	Min Dist. Accuracy T	Max Dist. Accuracy T
RN50	ImageNet	74.05	58.53	6.62	98.15	0.52	89.48
ViT-B/32	ImageNet	68.46	63.00	8.86	94.92	1.78	72.84
CLIP RN50	OpenAI WIT	77.58	64.77	12.74	99.71	0.12	96.42
CLIP ViT-B/32	OpenAI WIT	70.98	62.66	10.06	99.30	0.26	94.60
OpenCLIP ViT-B/32	LAION-400M	70.87	68.32	12.02	98.25	0.52	83.98
OpenCLIP ViT-B/32	LAION-2B	76.49	71.92	16.98	99.86	0.04	98.70

Table 8.4: AUC and accuracy results on the semantic space S and on the style space T. These results are obtained by training on the COCOFake training set with Stable Diffusion v1.4 images under the disentanglement setting and evaluating on test set of the COCOFake dataset, using data extracted from both Stable Diffusion v1.4 and v2.0.

8.2.4 Semantic-Style Disentangling Results

We then turn our attention to evaluating the semantic-style disentanglement approach, in which we aim at training two separate embedding spaces, one storing semantic information and the second focusing on style information. We evaluate the semantic projection in terms of overall AUC and full cluster and overall classification accuracy, and the style projection in terms of overall AUC and minimum and maximum distance accuracy. Specifically, this is done by performing linear probing on top of the two disentangled projections S and T, following the approach described in Sec. 8.2.3, and computing AUC and overall and full cluster accuracy scores. Instead, minimum and maximum distance accuracy are directly computed on the T projection, to evaluate the relative spatial positions of the elements inside each cluster after disentangling semantics and style.

Results are reported in Table 8.4 and Table 8.5 on the COCOFake test set for all the aforementioned backbones, training the semantic-style disentanglement

		Test Set (SD v2.0 → SD v2.0)					
Backbone	**Dataset**	Overall AUC S	Overall Accuracy S	Full Cluster Accuracy S	Overall AUC T	Min Dist. Accuracy T	Max Dist. Accuracy T
RN50	ImageNet	79.30	68.01	13.04	98.43	0.54	89.58
ViT-B/32	ImageNet	69.20	66.31	11.40	95.80	1.94	72.94
CLIP RN50	OpenAI WIT	85.54	80.71	31.92	99.79	0.04	97.92
CLIP ViT-B/32	OpenAI WIT	74.51	68.98	14.20	99.76	0.08	97.60
OpenCLIP ViT-B/32	LAION-400M	72.64	68.51	12.80	99.02	0.38	90.52
OpenCLIP ViT-B/32	LAION-2B	82.69	76.60	23.94	99.87	0.04	99.20
		Test Set (SD v2.0 → SD v1.4)					
Backbone	**Dataset**	Overall AUC S	Overall Accuracy S	Full Cluster Accuracy S	Overall AUC T	Min Dist. Accuracy T	Max Dist. Accuracy T
RN50	ImageNet	76.87	67.91	12.62	97.54	0.60	83.70
ViT-B/32	ImageNet	67.36	65.77	10.16	94.45	2.34	69.00
CLIP RN50	OpenAI WIT	83.00	78.67	27.10	99.76	0.06	97.66
CLIP ViT-B/32	OpenAI WIT	72.48	68.79	45.36	99.73	0.05	97.88
OpenCLIP ViT-B/32	LAION-400M	69.85	65.39	9.60	99.32	0.10	94.14
OpenCLIP ViT-B/32	LAION-2B	82.58	78.76	26.44	99.86	0.08	99.34

Table 8.5: AUC and accuracy results on the semantic space S and on the style space T. These results are obtained by training on the COCOFake training set with Stable Diffusion v2.0 images under the disentanglement setting and evaluating on test set of the COCOFake dataset, using data extracted from both Stable Diffusion v1.4 and v2.0.

on the training set respectively with Stable Diffusion v1.4 and v2.0 images. In both cases, we observe that, in the T space which focuses on style, real and fake images can be properly distinguished, as the real image is always far apart from the generated ones. On the contrary, this does not happen in the S space, which focuses on semantics, and in which all elements belonging to the same cluster are pulled together, independently of their authenticity. Still, the identification of deepfakes is feasible even in this more challenging space, although with lower AUC and accuracy scores (*i.e.*, with an AUC up to 86% and an accuracy of up to 80%.) As this corresponds to testing a more challenging generator that leaves fewer lower-level traces, we believe this result might offer interesting insights for future works. Similar but slightly lower results can also be observed when testing on images generated by the Stable Diffusion version not used during training, with an overall AUC up to 83% and an overall accuracy up to 79%. When instead considering the overall AUC computed over the T projection, we can notice that

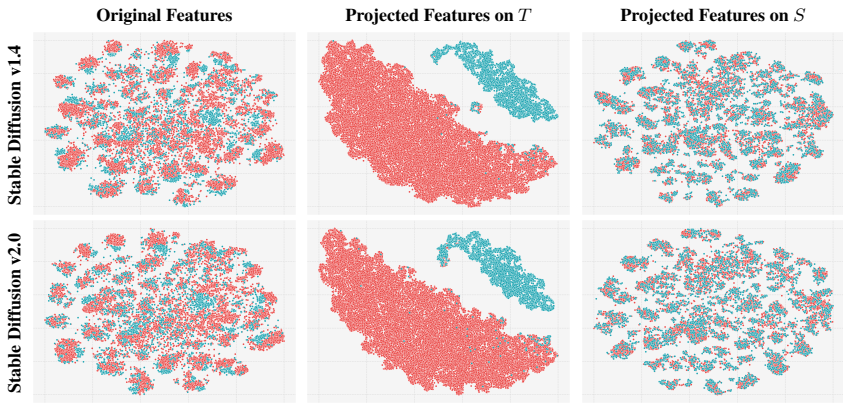


Figure 8.6: t-SNE visualizations over the validation set using the original visual features from the OpenCLIP ViT-B/32 LAION-2B backbone (left), the features projected on the T space (style) after disentanglement (middle), and the features projected on the S space (semantics) after disentanglement (right), using Stable Diffusion v1.4 (top) and v2.0 (bottom). Red dots indicate fake images, blue dots indicate real images.

the best results are above 99% across almost all settings, thus confirming the proper distinction between real and fake images in the T space.

The structure of the two spaces can be further visualized in Figure 8.6, in which we report 2D t-SNE visualizations [229] of the feature space of the OpenCLIP ViT-B/32 LAION-2B backbone, before and after disentanglement and for both Stable Diffusion v1.4 and Stable Diffusion v2.0. In the original embedding space, as provided by the backbone, real and generated samples appear to be mostly overlapped, even if we do not observe a complete overlap – which is in line with the results presented in Table 8.1 and Table 8.2. After the disentanglement, instead, the geometry of the T and S spaces appears completely different: the T space clearly separates real and fake data (with the exception of a few outliers), while in the S space we can observe a complete overlap between real and generated samples and a tendency to group into semantic clusters.

A closer visualization of the original feature space and the embedding spaces produced by the two projections is reported in Figure 8.7. In this case, we report, on each row, the relative positioning of eight sample clusters from the COCOFake

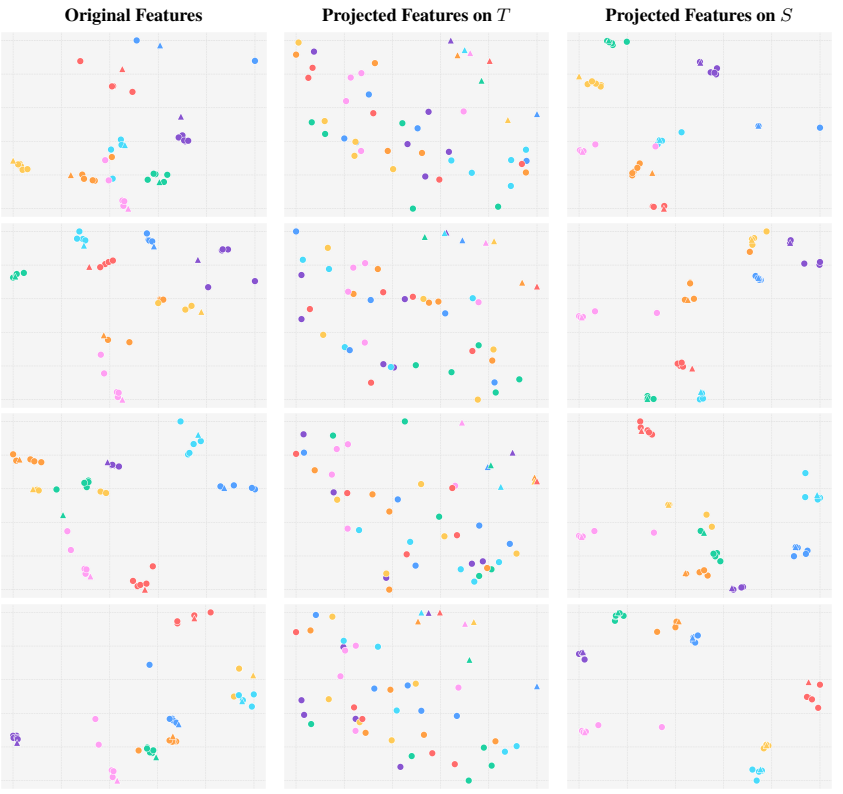


Figure 8.7: t-SNE visualizations on sampled clusters from the Stable Diffusion v1.4 test set using features extracted from the OpenCLIP ViT-B/32 architecture pre-trained on LAION-2B. We report the original features from the visual backbone (left), the features projected on the T space (style) after disentanglement (middle), and the features projected on the S space (semantics) after disentanglement (right). Dots indicate fake images, triangles indicate real images. Images from the same cluster are shown with the same color.

test set with Stable Diffusion v1.4 images. As it can be seen, the two proposed projections are again effective both in separating real and fake images and in promoting the clustering of images sharing similar semantics regardless of their authenticity.

		Gaussian Blur (SD v2.0 → SD v2.0)					
Backbone	**Dataset**	Overall AUC S	Overall Accuracy S	Full Cluster Accuracy S	Overall AUC T	Min Dist. Accuracy T	Max Dist. Accuracy T
CLIP RN50	OpenAI WIT	77.44	74.28	19.42	99.26	0.12	90.96
CLIP ViT-B/32	OpenAI WIT	70.41	59.65	7.72	99.48	0.16	94.70
OpenCLIP ViT-B/32	LAION-400M	71.20	68.75	12.52	98.27	0.56	86.28
OpenCLIP ViT-B/32	LAION-2B	79.31	75.16	21.38	99.80	0.12	98.50
		JPEG Compression (SD v2.0 → SD v2.0)					
Backbone	**Dataset**	Overall AUC S	Overall Accuracy S	Full Cluster Accuracy S	Overall AUC T	Min Dist. Accuracy T	Max Dist. Accuracy T
CLIP RN50	OpenAI WIT	61.64	55.05	5.10	88.60	3.62	57.62
CLIP ViT-B/32	OpenAI WIT	64.77	57.14	6.00	89.38	4.04	54.30
OpenCLIP ViT-B/32	LAION-400M	69.22	62.01	8.26	96.97	0.82	82.56
OpenCLIP ViT-B/32	LAION-2B	69.06	69.75	13.62	93.32	2.28	65.66
		Resize (SD v2.0 → SD v2.0)					
Backbone	**Dataset**	Overall AUC S	Overall Accuracy S	Full Cluster Accuracy S	Overall AUC T	Min Dist. Accuracy T	Max Dist. Accuracy T
CLIP RN50	OpenAI WIT	62.75	70.05	10.50	87.70	3.20	56.82
CLIP ViT-B/32	OpenAI WIT	67.78	31.70	0.36	90.85	2.70	62.28
OpenCLIP ViT-B/32	LAION-400M	71.61	41.34	1.74	94.32	2.16	72.50
OpenCLIP ViT-B/32	LAION-2B	75.12	30.71	0.70	86.72	6.00	40.54

Table 8.6: AUC and accuracy results on the semantic space S and on the style space T. These results are obtained by training on the COCOFake training set with Stable Diffusion v2.0 images under the disentanglement setting and evaluating on test set of the COCOFake dataset, using different image transformations.

8.2.5 Robustness Analysis to Image Transformations

As shown in recent literature [47, 76, 166], in addition to evaluating deepfake detection methods in a standard setting, it is also important to assess their robustness to image transformations, which may cause a severe performance degradation in some cases. To this aim, we replicate the experiment described in Sec. 8.2.4 by testing on real and fake images that have undergone one of three considered image transformation techniques (*i.e.*, Gaussian blur, JPEG compression, and resize). Specifically, we consider the disentangled spaces trained on non-transformed Stable Diffusion v2.0 images and evaluate on the corresponding test set where one image transformation is applied to all real and fake images, using a kernel size of 3 for Gaussian blurring, an image compression rate of 60 for JPEG compression, and an image edge size equal to 64 pixels for resizing.

Results are shown in Table 8.6 in terms of the previously described AUC and

Model	Validation Set (SD v1.4)			Test Set (SD v1.4)		
	Overall AUC	Overall Accuracy	Full Cluster Accuracy	Overall AUC	Overall Accuracy	Full Cluster Accuracy
Wang *et al*. (RN50 Blur+JPEG 0.5) [236]	40.61	83.26	0.34	41.29	83.26	0.48
Mandelli *et al*. (DetectGAN) [174]	54.55	83.12	4.78	54.84	83.09	5.06
Ours (CLIP RN50)	99.85	98.79	93.32	99.87	98.89	93.90
Ours (CLIP ViT-B/32)	99.79	98.63	92.26	99.74	98.47	91.58
Ours (OpenCLIP ViT-B/32-LAION-400M)	99.44	97.08	84.34	99.45	97.21	85.00
Ours (OpenCLIP ViT-B/32-LAION-2B)	**99.93**	**99.44**	**96.68**	**99.93**	**99.39**	**96.44**
Model	**Validation Set (SD v2.0)**			**Test Set (SD v2.0)**		
	Overall AUC	Overall Accuracy	Full Cluster Accuracy	Overall AUC	Overall Accuracy	Full Cluster Accuracy
Wang *et al*. (RN50 Blur+JPEG 0.5) [236]	53.05	83.32	0.40	53.53	83.35	0.48
Mandelli *et al*. (DetectGAN) [174]	64.26	83.55	4.98	64.79	83.55	5.28
Ours (CLIP RN50)	99.79	98.38	91.06	99.82	98.29	90.80
Ours (CLIP ViT-B/32)	99.76	98.29	90.50	99.69	98.14	90.10
Ours (OpenCLIP ViT-B/32-LAION-400M)	99.02	97.31	85.30	99.14	97.28	84.98
Ours (OpenCLIP ViT-B/32-LAION-2B)	**99.87**	**99.26**	**95.68**	**99.87**	**99.30**	**96.02**

Table 8.7: Comparison with existing deepfake detection methods in terms of overall AUC, overall accuracy, and full cluster accuracy. Our results are obtained by performing linear probing on the style space T using the validation and test sets of COCOFake, with images extracted from both Stable Diffusion v1.4 and v2.0.

accuracy evaluation metrics. Notably, while all image transformations cause a slight deterioration in performance, applying JPEG compression or scaling images to a lower resolution leads to the most drastic degradation of the final results, especially considering the results on the T space with an overall AUC of 89% to 97% for JPEG compression and 87% to 94% for resizing. Conversely, Gaussian blur does not significantly impact performance with an overall AUC above 98%.

8.2.6 Comparison with Other Methods

Finally, we compare our results with existing deepfake detection methods specifically tailored to recognize fake images from GAN-based generators. Specifically, we include in the comparison the models proposed by Wang *et al*. [236] which are based on a ResNet-50 model trained with different image transformations (*i.e*., Gaussian blur and JPEG compression) and DetectGAN [174] based on an ensemble of different CNNs. For both competitors, we use the pre-trained weights downloaded from the official repositories provided by the authors.

Table 8.7 reports the results in terms of overall AUC, overall accuracy, and full cluster accuracy on the validation and test sets of COCOFake, using images generated by both versions of Stable Diffusion. Our results are obtained after disentangling semantics and style and by performing linear probing on the style space T which is in charge of distinguishing real and fake images. As it can be seen, both competitors fail to effectively discriminate fake images from real ones with an overall AUC around 40% for the model proposed in [236] and 55% for the DetectGAN approach [174], when tested on Stable Diffusion v1.4 images. On the contrary, all versions of our model achieve AUC scores greater than 99% confirming the effectiveness of the T space in correctly detecting deepfakes.

Chapter 9

Multimodal Video Question Answering

The discussions in the previous chapters emphasized the versatility of Transformers in capturing complex relationships between different data modalities and their effectiveness in aligning high-level semantic features with fine-grained perceptual details. While these discussions primarily centered on static image understanding and its integration with textual information, in this chapter we extend our focus to video analysis and its unique challenges. Modeling temporal dependencies, dynamic content, and event reasoning in videos requires innovative multimodal approaches that go beyond static image paradigms.

In recent years, video understanding has garnered significant attention, fueled by advancements in multimodal learning and vision-language models [18, 155, 263, 283]. Videos are ubiquitous on the Internet and their inherently dynamic and complex nature poses a significant challenge for video understanding [170, 221, 281, 300]. Video Question Answering (Video QA) emerges as a particularly fundamental and challenging task within this domain, requiring models to not only perceive videos for recognition-level tasks like classification or segmentation but also to comprehend and rationalize the video content in response to specific questions. The challenge is twofold: models must process the temporal and

This chapter is related to the publication "Roberto Amoroso*, Gengyuan Zhang*, Rajat Koner, Lorenzo Baraldi, Rita Cucchiara, and Volker Tresp, Perceive, Query & Reason: Enhancing Video QA with Question-Guided Temporal Queries, WACV 2025 (* Equal Contribution)". Work realized during the ELLIS internship at the LMU University in Munich, Germany.

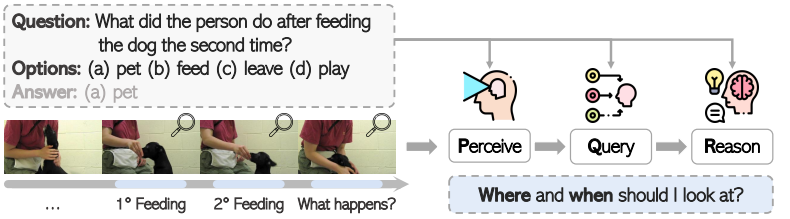


Figure 9.1: Adapting LLMs' visual reasoning capabilities to video QA requires extracting the most relevant video features based on the input question. Our approach addresses this challenge by extracting question-guided temporal features from dynamic frame sequences, enabling accurate and context-aware reasoning.

dynamic nature of videos with their sparse and heterogeneous event distributions, while simultaneously developing a deep understanding that connects these events through commonsense and temporal reasoning. This necessitates a blend of model abilities including visual perception, question contextualization, and answer reasoning.

The advent of Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) has marked significant progress in visual reasoning, including image QA [33, 140, 156] and video QA [275, 283] tasks. MLLMs exhibit strong general knowledge and reasoning capabilities, excelling in interpreting visual content for video QA applications. A common paradigm in MLLMs involves integrating a visual encoder for embedding video frames and a projection network that compresses visual features and aligns them with LLMs. Conventional compression strategies, such as sparse sampling and pooling-based methods [108, 134, 168, 305], face limitations when dealing with lengthy and dynamic videos. Recent studies [2, 283, 294, 308] have achieved more promising results by adopting attention-based models like Q-Former [57, 140]. The Q-Former extracts prominent visual features from the visual encoder into a small set of learnable queries, which are then aligned with the textual embeddings from a pre-trained LLM via a linear layer. This paradigm has proven effective across various vision-language tasks [276, 283, 294, 308], demonstrating that visual features can be compressed without compromising the overall representation while also reducing computational costs.

However, directly adapting image-based Q-Former approaches to video data often results in suboptimal performance due to the spatio-temporal complexities

and varying information density inherent in videos [275]. We hypothesize that pre-trained image-level models lack the necessary spatio-temporal inductive bias for video understanding. This limitation hinders the seamless integration of LLMs with video content, especially in scenarios involving complex dynamics and diverse information distributions.

Contributions. In this work, we introduce T-Former, a question-guided temporal querying Transformer designed to efficiently sample and learn video-specific features tailored to a given question. Unlike existing methods, T-Former leverages knowledge from image-based pre-training without requiring additional video pre-training. With T-Former, we propose a new LLM-based model, PQR, based on three core principles: ***P****erceive* to extract framewise question-guided spatial features from input videos, ***Q****uery* to extract a fixed-size set of question-guided temporal visual features from the whole sequence, and ***R****eason* to integrate these extracted features with LLMs for generating answers, as illustrated in Figure 9.1. T-Former employs an efficient temporal sampling strategy that incorporates an inductive bias for video queries by initializing learnable input queries with discrete key video event features. This design bridges the gap between pre-trained image-level knowledge and video-specific representations while focusing on relevant question-guided information. By doing so, T-Former ensures a seamless transition from generic to question-tailored video features.

Our evaluations demonstrate that T-Former outperforms alternative temporal modeling techniques, achieving peak performance at a sampling ratio equivalent to four frames, thereby effectively harnessing the potential of LLMs. Notably, PQR sets a new state-of-the-art benchmark in video QA, surpassing existing methods.

The rest of the chapter is organized as follows: Section 9.1 introduces our proposed framework, PQR, a novel LLM-based video QA model. This section provides an in-depth explanation of the three core modules—Perceive, Query, and Reason—alongside the innovative question-guided temporal querying Transformer (T-Former) for extracting question-relevant spatio-temporal features. Section 9.2 details the experimental evaluation of PQR across diverse video QA benchmarks, highlighting its superior performance compared to state-of-the-art models. This section also explores architectural design choices, ablation studies, and attention visualizations, which highlight the model's reasoning capabilities and its ability to identify important frames without explicit localization. Finally, we discuss broader insights, including the impact of linguistic bias in video QA datasets and additional evaluations on extended benchmarks, further validating the robustness and effectiveness of our approach.

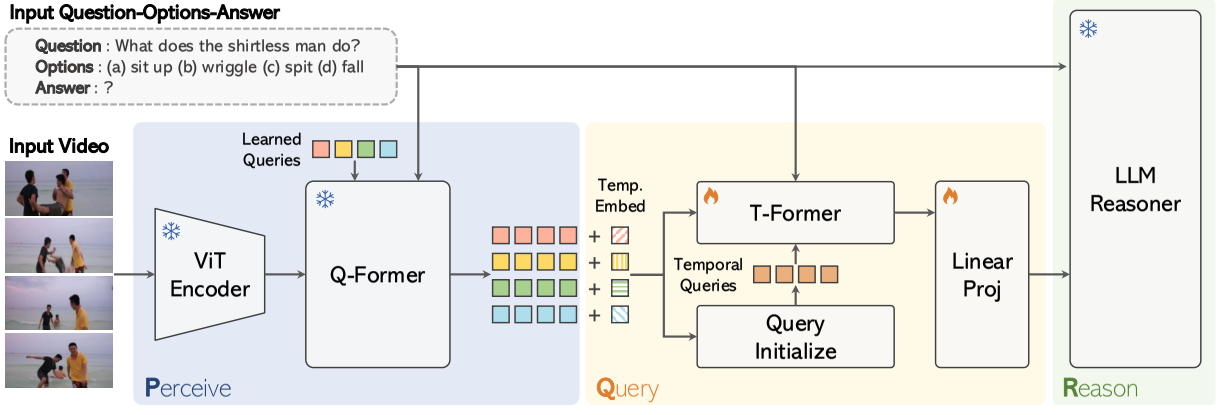


Figure 9.2: Our model consists of three primary stages: ***P**erceive*, ***Q**uery*, and ***R**eason*. The *Perceive* stage employs a ViT encoder and a Q-Former to extract spatial visual features from each frame independently, conditioned on the input question. The *Query* stage introduces T-Former, our question-guided temporal querying Transformer that captures the most relevant temporal information conditioned on both the question and the visual context. Finally, in the *Reason* stage, the condensed visual-temporal features are integrated with the question and answer options before being fed into a frozen LLM as a reasoning agent to rationalize and answer the question.

9.1 Enhancing Video QA with Question-Guided Temporal Queries

In this section, we introduce PQR, our LLM-based video QA model, alongside T-Former, a novel question-guided temporal querying Transformer. Our model, illustrated in Figure 9.2, consists of three main modules: (1) ***P**erceive*: framewise visual encoding of input videos conditioned on given questions; (2) ***Q**uery*: T-Former for selecting question-relevant temporal information across frames; and (3) ***R**eason*: an adapted LLM as a reasoning agent for question answering.

We start by formulating the video QA task. Given a video v as a sequence of n image frames $I^{1:n}$, where I^i represents the i-th frame of the video, and a question q, a model $f(\cdot)$ assigns a probability to an answer a from the answer space $\mathcal{A}$. This can be formulated as:

$$\hat{a} = \underset{a \in \mathcal{A}}{\arg\max}\, f(a|v, q, \mathcal{A}), \tag{9.1}$$

where $\hat{a}$ is the predicted answer given the video and questions. The answer space $\mathcal{A}$ can be large for open-end QA or preset for multiple-choice QA.

9.1.1 Perceive: Visual Encoding

State-of-the-art visual encoders such as CLIP [191] and BLIP-2 [140] have demonstrated remarkable zero-shot capabilities in generating visual features, making it a common practice [275] to leverage these powerful features for video QA tasks. Despite their success, these models are restricted to extracting vision-only features that ignore other modalities, like textual questions in our task. This leads to question-agnostic visual feature extraction and a lack of contextual relevance to complex questions.

To address this limitation, our PQR employs a question-guided visual feature extraction mechanism. Specifically, we adopt the instruction-aware Q-Former proposed by [57] to extract question-guided features from each frame. In more detail, given a sequence $I^{1:n}$, we encode each frame *independently* to get a sequence $\mathbf{e}_f^{1:n}$ of framewise visual tokens:

$$\mathbf{e}_f^i = \mathrm{VisualEncoder}(I^i, q, \mathcal{A}), \forall i \in \{1, 2, \ldots, n\}, \tag{9.2}$$

where $\mathbf{e}_f^i$ represents the visual features of i-th frame, and $\mathrm{VisualEncoder}$ refers to both the frame encoder and Q-Former for simplicity. This step aims to *perceive* and extract spatial visual information from raw frames.

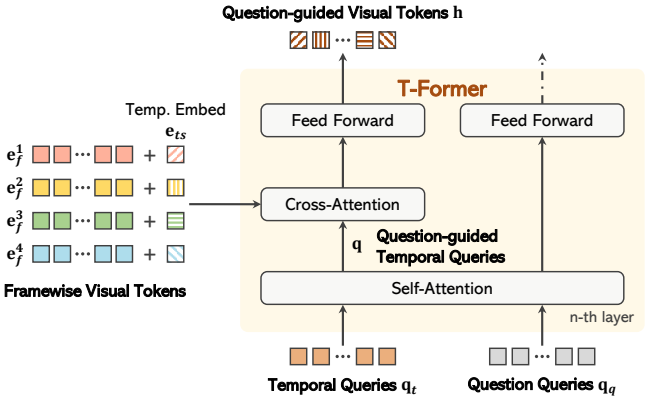


Figure 9.3: Overview of T-Former, a Temporal Querying Transformer. T-Former adopts 1) a self-attention layer between temporal queries and question queries and 2) cross-attention layers between query tokens and the full-length sequence of visual tokens.

9.1.2 Query: T-Former

Handling a full sequence of framewise visual tokens $\mathbf{e}_f^{i:n}$ leads to temporal information redundancy across frames and is also computationally demanding due to the complexity of LLMs. To mitigate these challenges, we propose T-Former, a question-guided ***t***emporal querying trans***former***, designed to select the most relevant visual features across time based on the input question, as presented in Figure 9.3.

T-Former integrates three distinct types of input tokens: 1) the concatenation of framewise visual tokens $\mathbf{e}_f^{1:n}$ from the *perceive* step, 2) temporal queries $\mathbf{q}_t$ of a fixed length to *query* temporally relevant information from the whole framewise visual token sequence, and 3) question queries $\mathbf{q}_q$ to condition the output tokens on the questions. T-Former first contextualizes query tokens with self-attention between $\mathbf{q}_t$ and $\mathbf{q}_q$; and then performs cross-attention between $\mathbf{q}_t$ and $\mathbf{e}_f$ to select question-relevant visual information over time. We elucidate each module as follows.

Temporal Query Initialization. The core of our approach lies in the initialization of a small number of fixed-length temporal queries, $\mathbf{q}_t$, which are specific to the input video and the question. Existing methods [57, 140, 283, 308] typic-

ally employ learnable queries with random initialization to compress features. However, these randomly initialized static queries require extensive video-based pre-training, as they lack any inherent alignment with the spatio-temporal dynamics of video events. Moreover, even after pre-training, these static queries remain identical across all videos, failing to capture video-specific contextual information. To overcome these limitations, we propose a novel strategy that replaces static, pre-trained queries with dynamic queries sampled directly from the input video features. By grounding the query tokens in the unique characteristics of the input video, our approach introduces a more effective inductive bias for T-Former that aligns with Q-Former's image understanding capabilities. This dynamic grounding not only enables a deeper understanding of the video but also facilitates more precise responses to video-related questions.

We investigate four distinct sampling strategies for temporal query initialization: uniform sampling, random sampling, k-means sampling, and k-medoids sampling:

$$\mathbf{q}_t = \text{sampler}(\mathbf{e}_f^1, \mathbf{e}_f^2, ..., \mathbf{e}_f^n) \tag{9.3}$$

The size of temporal queries $\mathbf{q}_t$ is considered a crucial model hyperparameter, governing two essential aspects. Firstly, it defines the sampling ratio of T-Former, thereby controlling how much the input sequence is compressed. Secondly, it influences the length of visual tokens fed into LLMs. Hence, a higher number of temporal queries increases computational overheads within LLMs. In Section sec:design, we investigate the effects of various sampling methods and sampling ratios.

Question-guided Temporal Query Tokens. To condition the temporal queries $\mathbf{q}_t$ based on the questions, we concatenate $\mathbf{q}_t$ with the question queries $\mathbf{q}_q$. These question queries are generated through the tokenization process applied to the question q and the answer options $\mathcal{A}$. To facilitate interaction between different modalities, we employ self-attention layers to contextualize both $\mathbf{q}_t$ and $\mathbf{q}_q$. This process is formulated as follows:

$$\mathbf{Q} = \mathbf{K} = \mathbf{V} = \phi(\text{concat}(\mathbf{q}_t, \mathbf{q}_q)), \tag{9.4}$$

$$\mathbf{q} = \text{self-attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \tag{9.5}$$

where ϕ represents the linear projection function of the attention [230] mechanism and $\mathbf{q}$ denotes the resulting question-guided temporal query tokens.

Query-Frame Cross-attention. The sampled temporal queries still lack spatio-temporal information from videos. Therefore, revisiting the video to enhance the

queries with more detailed and fine-grained video-specific information becomes imperative.

To this aim, given the contextualized queries $\mathbf{q}$, we employ cross-attention between $\mathbf{q}$ and the entire sequence of video features. The process also incorporates learnable timestamp embeddings $\mathbf{e}_{ts}$ to capture temporal dependencies, as formulated in the following:

$$\hat{\mathbf{e}}_f^i = \mathbf{e}_f^i + \mathbf{e}_{ts}, \tag{9.6}$$

$$\mathbf{Q} = \psi_q(\mathbf{q}), \quad \mathbf{K} = \mathbf{V} = \psi_{kv}(\text{concat}(\hat{\mathbf{e}}_f^1, ..., \hat{\mathbf{e}}_f^n)), \tag{9.7}$$

$$\mathbf{h} = \text{cross-attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}). \tag{9.8}$$

Finally, we pass the final T-Former hidden state $\mathbf{h}$ through a feed-forward layer to derive a representation $\mathbf{t}_v$ that integrates video content, question context, and required information for a precise answer. Importantly, our cross-attention mechanism between the temporal queries and the full-length sequence of video tokens enables the T-Former to perform question-guided spatio-temporal feature extraction. This mechanism enriches temporal query tokens by integrating additional relevant information from the full sequence of video tokens, compensating for the partial context captured during initial sampling and ensuring a comprehensive, question-specific understanding of the video.

9.1.3 Reason: LLMs as reasoning agents

The reasoning component of our framework leverages an LLM to reason and extract meaningful insights from the rich visual-temporal representations produced by our T-Former. Aggregated visual features from T-Former, question tokens, and answer option tokens from the LLM tokenizer are fed as inputs to the LLM reasoner. We incorporate a linear projection layer to reformat the visual features and ensure alignment with the LLM's feature space:

$$\mathbf{h}_{vq\mathcal{A}} = \text{LLM}(\text{linear-proj}(\mathbf{t}_v), q, \mathcal{A}). \tag{9.9}$$

The LLM's hidden state $\mathbf{h}_{vqA}$ represents a unified embedding that captures the complex interactions between the contextualized representations of video, question, and answer space $(v, q, \mathcal{A})$. For multiple-choice video QA tasks, we append a linear projection layer, as a common practice, to generate logits for cross-entropy loss computation.

This enhanced reasoning module enables sophisticated multi-modal understanding, leveraging the pre-trained knowledge of the LLM while maintaining efficiency through our selective temporal feature extraction approach.

9.2 Experiments

This section presents a comprehensive evaluation of our proposed PQR. The experiments are designed to assess the model's performance across diverse benchmarks, explore its architectural design choices, and compare it against state-of-the-art methods. We begin by detailing the experimental setup, including the tasks, datasets, and implementation specifics. Following this, we analyze various temporal modeling strategies and investigate critical design decisions that influence the model's performance. Extensive ablation studies are conducted to isolate the contributions of individual components within PQR. Additionally, we visualize the model's attention mechanisms to provide insights into its reasoning capabilities. Finally, we benchmark PQR against existing models, highlighting its superior performance across multiple datasets and reasoning categories, while also addressing concerns such as linguistic bias in video QA datasets.

9.2.1 Experimental Setup

Task and Datasets. We evaluate PQR in a multiple-choice video QA setting, where each question is paired with a fixed number of answer choices. The primary evaluation is conducted on the NExT-QA dataset [252], which serves as the development benchmark for exploring various architectural decisions. To ensure a comprehensive assessment, we extend the evaluation to three additional competitive video QA benchmarks—NExT-QA, STAR [249], and How2QA [141]—and one video event prediction benchmark, VLEP [137]. The NExT-QA dataset comprises 5,440 videos and 52K questions, requiring models to perform temporal, causal, and descriptive reasoning. STAR includes 22K videos and 60K question-answer pairs across four categories: Interaction, Sequence, Prediction, and Feasibility. It emphasizes reasoning about environments and situations. How2QA consists of 9,035 YouTube videos with an average length of 17.45 seconds and 44K crowdsourced questions, each with four candidate answers. VLEP, a future event prediction benchmark, contains 10,234 videos from TV shows and YouTube with 28,726 samples. Here, the task involves predicting the next likely event from two options.

We follow established evaluation protocols [57, 283] for each benchmark to ensure comparability with prior works. Notably, PQR is trained exclusively on target datasets without requiring any video-based pre-training phase.

Implementation Details. Our framework adopts the ViT-G/14 from [74] and the instruction-aware Q-Former from [57] as the pre-trained visual encoder for frame processing. The reasoning component in PQR is powered by a frozen

Dataset	Batch Size	Epochs	Iterations per Epoch	Warmup Epochs	Cooldown Epochs	Initial LR	Warmup LR	Minimum LR
NExT-QA	2	10	2500	1	2	3e-5	8e-6	1e-6
STAR	2	10	5000	1	2	5e-5	1e-5	1e-6
How2QA	4	10	5000	1	5	3e-5	8e-6	1e-6
VLEP	4	10	1000	1	5	2e-5	7e-6	1e-6

Table 9.1: PQR training hyperparameters for different datasets.

Vicuna-7B model [298], an instruction-tuned LLM built on top of the LLaMA architecture [226]. Within PQR, we integrate two layers of our T-Former, which is based on a BERT-structured Transformer. To ensure a smooth integration between T-Former 's visual features and the LLM's feature space, we utilize a linear projection layer. Additionally, a trainable linear layer is appended to the LLM for fine-tuning in the multi-choice video QA setting. During training, both the visual encoder and the LLM remain frozen, and only T-Former and the linear projection layers are trained.

Training Details. Our PQR is trained on a single NVIDIA A6000 48GB GPU. We employ the AdamW optimizer with a linear warmup and cosine annealing scheduler. We provide a detailed overview of the training hyperparameters used across all benchmark datasets in Table 9.1, which presents the optimal values for key parameters, including batch size, total epoch numbers, number of iteration steps per epoch, warm-up and cooldown epochs, and learning rate.

LLM Prompts. For all tasks, we adhere to the protocol of [140, 57] and adopt a canonical context template: "*Question:* [<*Question*>]. *Options:* [<*Option Lists*>]. *Answer:*" used for textual inputs of Q-Former, T-Former, and the LLM reasoner. For the future event prediction task on VLEP, we use the same question template, "*Which event is more likely to happen right after?*", for all samples.

Evaluation Metrics. We report the standard answer accuracy metric for the multiple-choice video QA tasks. Each dataset has a different number of candidate answer options: NExT-QA provides five options per question, STAR and How2QA provide four options, and VLEP offers two options. Additionally, NExT-QA categorizes questions into three types: temporal, causal, and descriptive, while STAR delineates four types: interaction, sequence, prediction, and feasibility.

Sampling Strategy	#Input Frames	NExT-QA			
		Tem.	Cau.	Des.	Avg.
Single Frame	1	67.7	72.3	79.0	71.9
Concatenate	4	66.1	72.3	80.6	71.6
Concatenate	16	69.6	74.0	81.1	73.2
Mean-pooling	4	64.5	69.6	77.2	69.1
Mean-pooling	16	64.6	70.5	79.6	70.0
Spatio-Temporal	4	64.5	70.7	77.5	69.8
Spatio-Temporal	16	67.7	72.6	79.5	72.1
T-Former (Ours)	16	**72.8**	**76.9**	**84.7**	**76.6**

Table 9.2: Comparing T-Former with other temporal modeling methods. T-Former outperforms other sampling methods.

9.2.2 Comparison to Other Temporal Modeling

We compare our T-Former against several temporal modeling baselines on the NExT-QA dataset, including:

- *Single Frame Sampling*: A single frame is randomly sampled from the whole video sequence.
- *Frame Concatenation*: Visual tokens from each frame are concatenated before being fed into the LLM.
- *Mean-Pooling*: Framewise outputs from the visual encoder are merged via mean pooling.
- *Spatio-Temporal Transformers*: The ViT features from all frames are concatenated in sequence and input into the Q-Former, which jointly extracts spatio-temporal features before passing them to the LLM.

Our results, summarized in Table 9.2, demonstrate that T-Former outperforms these baselines by a significant margin, with improvements ranging from 3.4% to 7.5%. Interestingly, simple baseline methods like single-frame sampling achieve decent performance, suggesting that the dataset may still exhibit temporal bias. However, T-Former consistently excels, particularly in temporal and causal reasoning, highlighting its superior temporal modeling capabilities.

Sampling Strategy	NExT-QA			
	Tem.	Cau.	Des.	Avg.
Uniform	70.8	74.9	83.8	75.0
Random	71.7	75.0	84.6	75.4
k-means	71.9	76.7	83.4	76.2
k-medoids	**72.8**	**76.9**	**84.7**	**76.7**

Table 9.3: Ablation studies of different sampling methods for temporal query initialization. The number of input frames is 16. The clustering-based methods outperform other techniques.

Temporal Queries Initialization	NExT-QA			
	Tem.	Cau.	Des.	Avg.
Learnable	63.0	65.2	73.2	65.7
Sampled	**72.8**	**76.9**	**84.7**	**76.7**

Table 9.4: Impact of replacing sampled temporal queries with learnable temporal queries. Results indicate that sampled temporal queries outperform their learnable counterparts.

9.2.3 Exploring Design Choices

We perform a comprehensive analysis of different design choices for T-Former, examining the impact of sampling methods, Transformer settings, and sampling ratios.

Sampling Methods. We compare four methods for initializing queries $\mathbf{q}_t$: uniform sampling, random sampling, k-means sampling, and k-medoids sampling. In our experiments, we fix the input frames at 16. The clustering-based methods, *i.e.*, k-means and k-medoids, outperform other techniques, as shown in Table 9.3. This aligns with our intuition that clustering selects more diverse and crucial frames, providing better initialization.

We also conducted an experiment involving learnable temporal queries akin to the learnable queries in Q-Former. The results, shown in Table 9.4, reveal that our sampling-based approach consistently outperforms learnable temporal queries by a significant margin. This suggests that the initial sampling strategy for temporal queries contributes substantially to the overall performance.

#Hidden Layers	NExT-QA			
	Tem.	Cau.	Des.	Avg.
12	67.1	72.1	80.1	71.7
8	69.1	74.8	83.4	74.3
4	71.1	75.4	83.9	75.3
2	**72.8**	**76.9**	**84.7**	**76.7**
1	70.1	75.3	84.3	75.1

Table 9.5: Effect of the number of hidden layers on T-Former performance.

#Attn. Heads	NExT-QA			
	Tem.	Cau.	Des.	Avg.
2	71.5	75.8	82.6	75.5
4	71.8	75.9	83.7	75.8
8	71.9	76.5	84.1	76.2
12	**72.8**	**76.9**	**84.7**	**76.7**
16	72.7	76.5	84.0	76.4

Table 9.6: Effect of the number of attention heads. Increasing the number of attention heads improves the model's performance.

Investigating T-Former Settings. We investigate the impact of varying our T-Former architecture settings, including the number of hidden layers and attention heads. In Table 9.5, we show that T-Former with two hidden layers represents the most effective solution. We speculate that since Q-Former already captures intricate and abstract visual features, T-Former can easily grasp the temporal modeling of video frames with its lightweight structure.

As shown in Table 9.6, increasing the number of attention heads improves performance, with 12 heads yielding the best results. We believe that having more attention heads helps to encode diverse relationships between video frames, a critical capability for comprehensive temporal reasoning.

Impact of Video-frame Sampling Ratio. We analyze how the sampling ratio, defined as "number of input frames $\rightarrow$ numbers of equivalent output frames", influences the performance of PQR. A higher sampling ratio compresses the input frame sequence into a smaller number of output tokens, referred to as the number of equivalent output frames. Conversely, a lower sampling ratio produces more fine-grained representations, but it also leads to longer input sequences for LLMs.

Sampling Ratio	NExT-QA			
	Tem.	Cau.	Des.	Avg.
$8 \rightarrow 1$	70.9	75.9	82.4	74.8
$16 \rightarrow 1$	71.0	75.3	81.9	74.9
$32 \rightarrow 1$	71.4	75.4	83.2	75.3
$64 \rightarrow 1$	71.5	75.8	82.4	75.4
$128 \rightarrow 1$	71.4	75.3	83.3	75.3
$8 \rightarrow 4$	71.9	76.8	**84.8**	76.4
$16 \rightarrow 4$	72.8	76.9	84.7	**76.6**
$32 \rightarrow 4$	73.0	76.1	84.6	76.4
$64 \rightarrow 4$	72.4	**77.0**	84.3	**76.6**
$128 \rightarrow 4$	**73.2**	76.6	84.4	**76.6**
$8 \rightarrow 8$	71.6	75.8	83.5	75.7
$16 \rightarrow 8$	72.1	**77.0**	84.3	76.5
$32 \rightarrow 8$	72.8	76.7	84.2	**76.6**
$64 \rightarrow 8$	72.3	76.2	84.7	76.3

Table 9.7: We conduct a comprehensive analysis to determine the optimal sampling ratio, defined as "#input frames $\rightarrow$ #equivalent output frames".

As shown in Table 9.7, for a fixed sampling ratio, such as $8 \rightarrow 1$ and $32 \rightarrow 4$, larger output sizes consistently yield better performance by accommodating more information. Conversely, compressing entire video frames into a small number of tokens while keeping the input frame size constant, as in $16 \rightarrow 1$ and $16 \rightarrow 4$, can hinder performance. Moreover, when the output frame size is fixed, as in $16 \rightarrow 4$, $32 \rightarrow 4$, and $64 \rightarrow 4$, incorporating more input frames enhances the performance by introducing additional information. However, there is an upper limit beyond which adding more frames no longer enhances performance.

Our findings confirm that increasing the number of input frames improves the performance, but only up to a certain limit. While additional frames provide more question-aware information, they may also introduce excessive redundancy. Additionally, the number of output frames is critical for achieving optimal results. We found that using four output frames consistently yields the best results, irrespective of the number of input frames. We hypothesize that an insufficient number of output frames may fail to capture enough information, while a large number could introduce redundancy. Longer sequences may also divert the attention of LLMs, potentially compromising performance. Furthermore, the absolute number of output frames outweighs a fixed sampling ratio, highlighting the importance of carefully selecting the number of output frames to ensure optimal performance.

T-Former	Question Guided	Temp. Embed.	NExT-QA				STAR					H2QA	VLEP
			Tem.	Cau.	Des.	Avg.	Int.	Seq.	Pre.	Fea.	Avg.		
✗	✗	✗	70.3	75.8	83.4	75.4	61.0	66.3	70.3	69.8	65.1	81.2	68.6
✓	✗	✗	72.2	76.0	84.6	76.1	60.8	67.2	71.6	70.4	65.4	84.4	69.0
✓	✓	✗	71.5	77.0	83.9	76.3	60.8	66.8	71.2	70.2	65.4	85.6	69.3
✓	✗	✓	**73.5**	75.1	**85.5**	76.2	63.0	68.2	70.2	72.6	66.6	84.5	69.2
✓	✓	✓	72.8	**77.0**	84.7	**76.7**	**62.8**	**69.6**	**72.8**	**70.0**	**67.6**	**85.9**	**69.6**

Table 9.8: Ablation studies of each module of PQR. Ablating different modules including T-Former, question-guided visual features, and temporal embeddings.

#Linear Layers	Bottleneck Dim	NExT-QA			
		Tem.	Cau.	Des.	Avg.
2	3,072	72.4	75.8	81.7	75.7
2	1,536	72.5	**77.0**	82.5	76.4
2	768	**72.8**	76.9	**84.7**	**76.7**
1	768	71.7	75.0	82.9	75.2

Table 9.9: Effect of different feed-forward bottleneck size on model performance.

9.2.4 Ablation Studies

Assessing the Role of T-Former and Question-Guided Queries. To assess the contribution of various components in PQR, we conduct an ablation study, as detailed in Table 9.8. We ablate T-Former, question-guided queries, and temporal embeddings. Our results indicate that all key components contribute to improved performance across multiple datasets. Notably, the absence of T-Former results in significant performance degradation, highlighting its critical role. Additionally, leveraging question tokens to extract question-guided visual features also enhances performance. This collective improvement underscores the success and efficacy of our question-guided, localization-free spatio-temporal feature extraction approach.

T-Former Settings. In this section, we present an additional ablation study to investigate the impact of layer number and intermediate dimensionality in the feed-forward layer of the T-Former, as shown in Table 9.9. Our experiments demonstrate that increasing the number of hidden layers improves model performance, while larger bottleneck dimensionality yields the opposite effect. Our findings suggest that a configuration of 2 hidden layers with a 768-dimensional feed-forward layer yields the best performance.

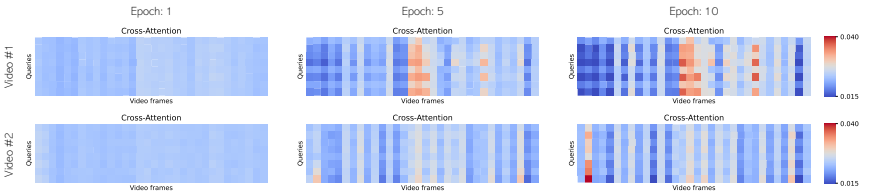


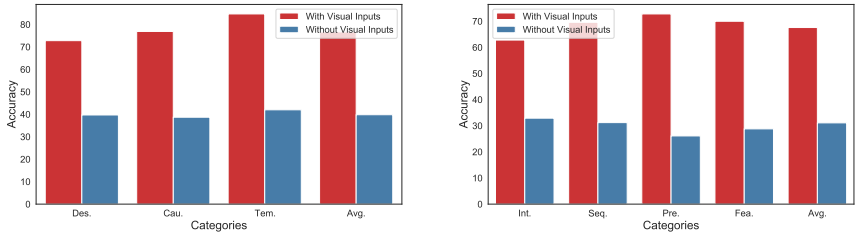
Figure 9.4: For different video samples (on each row), we visualize the attention map between question-guided temporal queries $\mathbf{q}$ and framewise visual tokens $\mathbf{e}_f^{1:n}$ in the last layer of T-Former during epochs 1, 5, and 10.

9.2.5 Visualizing T-Former Attention Mechanisms

To highlight the learning behavior of T-Former, we attempt to visualize the query-video cross-attention maps. Figure 9.4 showcases the attention map between question-guided temporal queries $\mathbf{q}$ and framewise visual tokens $\mathbf{e}_f^{1:n}$ in the last layer of T-Former across epochs 1, 5, and 10. We observe that throughout the training, T-Former effectively learns the temporal dependencies between frames at different timestamps across the entire video sequence. The frames with the highest attention scores are distributed across the video, indicating T-Former's ability to discern the importance of different frames. This also demonstrates that T-Former can attend to and differentiate the significance of frames, enabling it to identify the most crucial frames without relying on explicit localization methods.

9.2.6 Exploring Linguistic Bias

Linguistic bias in video question-answering datasets is a significant concern when using Large Language Models (LLMs). We conduct a comprehensive analysis to verify if the questions in the benchmarks contain biases that enable models to answer correctly without visual input. Figure 9.5 presents the full performance results across different datasets and categories. Our observations indicate that in the absence of visual information, LLM reasoners exhibit modest performance, comparable to a "blind guess". This finding highlights the robustness of the video question-answering benchmarks, ensuring they are minimally influenced by linguistic bias. Notably, the performance gap between our model and LLM reasoners is even more pronounced in categories such as causal and temporal reasoning. This underscores the effectiveness of our approach in leveraging visual information rather than being overly dependent on linguistic cues.



(a) NextQA (b) STAR

Figure 9.5: Exploring Linguistic Bias. We observe that LLM reasoners can only achieve a modest performance, akin to a "blind guess" when visual inputs are absent.

9.2.7 Comparison to State-of-the-art Models

Finally, we conduct a comprehensive comparison of our proposed PQR model with state-of-the-art approaches across all four benchmarks, as presented in Table 9.10. Notably, PQR consistently outperforms existing models.

On NExT-QA, our model achieves the best performance in the overall category, with a 2.9% improvement. It also excels in the three subcategories that emphasize temporal modeling and causal reasoning, demonstrating the effectiveness of the T-Former module. Similarly, on the STAR dataset, PQR delivers a 2.2% performance gain over current models while achieving top results across all subcategories. On How2QA, our model surpasses all other competitors by 2.3%.

Our results reveal a significant performance boost compared to non-LLM baseline models, confirming the potency of LLMs as robust visual reasoners when provided with appropriate visual context. Moreover, PQR surpasses LLM-based competitors, particularly excelling in causal reasoning on NExT-QA and event prediction on STAR, further validating the efficacy of PQR.

Notably, PQR utilizes a smaller LLM as the reasoning backbone and a limited number of learnable parameters compared to other models. Furthermore, unlike most competitors that rely on extensive video-based pre-training, our model is trained exclusively on the target datasets. This highlights the efficiency and effectiveness of our approach.

Model	w/ LLM	#Params	NExT-QA				STAR					How2QA	VLEP
		Train (Tot.)	Tem.	Cau.	Des.	Overall.	Int.	Seq.	Pre.	Fea.	Overall.		
Flamingo-9B 4-shot [2]	✓	- (9B)	-	-	-	-	-	-	-	-	42.8	-	-
Flamingo-9B 32-shot [2]	✓	- (9B)	-	-	-	-	-	-	-	-	41.2	-	-
Flamingo-80B 4-shot [2]	✓	- (80B)	-	-	-	-	-	-	-	-	42.4	-	-
Flamingo-80B 32-shot [2]	✓	- (80B)	-	-	-	-	-	-	-	-	42.2	-	-
FrozenBiLM [266]	✗	30M (30M)	-	-	-	-	-	-	-	-	-	81.5	-
All-in-One [234]	✗	110M (110M)	48.6	48.0	63.2	50.6	47.5	50.8	47.7	44.0	47.5	-	-
ATP [14]	✗	-	53.1	50.2	66.8	54.3	50.6	52.9	49.4	40.6	48.4	-	-
MIST [81]	✗	-	56.6	54.6	66.9	57.1	55.5	54.2	54.2	44.4	51.1	-	-
TranSTR [146]	✗	-	59.7	60.2	70.0	61.5	-	-	-	-	-	-	-
HiTeA [271]	✗	-	58.3	62.4	75.6	63.1	-	-	-	-	-	-	-
InternVideo [241]	✗	1.3B (1.3B)	58.5	62.5	75.8	63.2	62.7	65.6	54.9	51.9	58.7	79.0	63.9
BLIP-2 [140]	✓	188M (7.8B)	68.1	72.9	81.2	72.6	65.4	69.0	59.7	54.2	62.0	82.2	68.6
SeViLa [275]	✓	106M (4.1B)	69.4	74.2	81.3	73.8	63.7	**70.4**	63.1	62.4	64.9	83.6	68.9
LLaMA-VQA [127]	✓	4.5M (7B)	72.7	69.2	75.8	72.0	**66.2**	67.9	57.2	52.7	65.4	-	**71.0**
PQR (Ours)	✓	50.1M (7.8B)	**72.8**	**76.9**	**84.7**	**76.7** (+2.9)	62.8	69.6	**72.8**	**70.0**	**67.6** (+2.2)	**85.9** (+2.3)	69.6

Table 9.10: Comparison with SOTA video QA models. NExT-QA involves *tem*poral, *cau*sal, and *des*criptive question types. STAR contains four question types: *int*eraction, *seq*uence, *pre*diction, and *fea*sibility. We emphasize the best performance in **bold**, and underline the second-best performance. PQR consistently outperforms SOTA models. Particularly, it manifests great temporal and causal reasoning capabilities on NeXT-QA and STAR. These results underscore the effectiveness of PQR's temporal modeling for VideoQA tasks.

Chapter 10

Conclusions

This thesis has explored the intersection of attention-based models, multimodal learning, and visual-semantic understanding. By addressing challenges across a wide spectrum of areas—including the development and application of Vision Transformers (ViTs), multimodal integration, and reasoning—this work contributes to advancing the capabilities of AI systems interpreting and reasoning about visual content. In this concluding chapter, we summarize the key contributions of this thesis, discuss potential directions for future research, and acknowledge those who have contributed to the development of the proposed solutions.

Vision Transformer Architectures

In Chapter 3, we investigated the foundational role of ViTs in Computer Vision, highlighting their ability to model long-range dependencies and capture global relationships in visual data. Despite their transformative impact, ViTs face significant computational efficiency and scalability challenges. To address these issues, we introduced a bidimensional pooling strategy for intermediate patch sequences in ViT architectures. This approach, akin to downsampling in CNNs, reduces computational requirements and memory usage while enhancing hierarchical input representation. Our evaluations on popular classification datasets demonstrated that this method nearly halves the required FLOPs without sacrificing performance and consistently improves classification accuracy. This contribution enhances the practicality of ViTs for real-world applications and provides insights into optimizing self-attention mechanisms for dense prediction tasks.

Self-supervised Pre-training for Vision Transformers

Chapter 4 focused on the critical role of pre-training strategies in enabling effective Vision Transformer models. We proposed MaPeT, a novel self-supervised pre-training approach designed for vision tasks. Our model effectively tackles the limitations of the standard Masked Image Modeling strategy by employing a permutation-based objective to capture the interdependencies among predicted tokens and auxiliary position information to enable the model to access a full sequence of image patches, thus reducing the discrepancy between pre-training and fine-tuning. Moreover, we introduce the k-CLIP tokenizer that can densely capture the semantic information of the visual input by leveraging discretized CLIP features as visual tokens. Experimental results demonstrate that our approach can significantly enhance the performance of ViTs across a range of tasks while reducing reliance on large annotated datasets.

Integrating Attentive Models with Classical Computer Vision Techniques

While attention-based architectures have achieved remarkable success, classical Computer Vision techniques continue to offer valuable insights. In Chapter 5, we investigated the effectiveness of integrating superpixels in Transformer-based architectures for semantic segmentation. We proposed a novel superpixel-based positional encoding strategy that injects superpixel shape and position priors into the ViT encoder features, creating more boundary-aware semantic latent space representations. Our experimental evaluation using different architectures and datasets showed that our proposed superpixel positional encoding approach provides a significant performance improvement. Notably, our approach shows improved performance on classes with low occurrence in the dataset, addressing the challenge of accurate segmentation of rare classes. At the same time, it mitigates the risk of overfitting for classes with higher representation, ensuring a good balance between generalization and specificity. By incorporating superpixel priors into positional encoding strategies, we demonstrated how traditional perception-based approaches can enhance boundary awareness and spatial reasoning in dense prediction tasks like semantic segmentation. This fusion underscores the importance of leveraging complementary strengths from both classical and modern methodologies.

Multimodal Semantic Segmentation

The potential of Transformers extends beyond purely visual tasks into multimodal learning, where vision is integrated with language to enable sophisticated cross-

modal reasoning. A significant challenge in this domain is open-vocabulary semantic segmentation, which involves segmenting unseen categories based on textual descriptions. Addressing this, Chapter 6 introduced FreeDA, a novel training-free approach for unsupervised open-vocabulary segmentation. Our approach leverages visual prototypes and textual keys extracted offline with diffusion-augmented generation and exploits local-global similarities at inference time. FreeDA's contributions are multifaceted. First, it demonstrates the efficacy of leveraging diffusion models for generating synthetic visual data paired with weak localization masks, significantly enhancing the diversity and contextual richness of visual prototypes. Second, by employing self-supervised backbones and integrating superpixel-based segmentation, FreeDA achieves fine-grained boundary delineation while maintaining semantic coherence. Third, its innovative combination of local and global similarity measures ensures robust text-pixel alignment, even in challenging scenarios. Experimentally, we achieve state-of-the-art results on five different datasets, effectively addressing domain shift problems without extensive training or annotated datasets.

Fine-grained Evaluation Metrics for Segmentation

Recognizing limitations in traditional evaluation metrics like mean Intersection over Union ($mIoU$), Chapter 7 proposed a novel fine-grained error analysis framework for semantic segmentation models. Our intuitive error categorization allows for the investigation of the strengths and weaknesses of semantic segmentation models in a quantitative way. We conducted an extensive analysis, including various semantic segmentation architectures, datasets, and learning settings. Extensive analysis revealed high segment error rates in mask-classification-based architectures but showed that combining them with models producing fewer segment errors achieved state-of-the-art performance. Our framework facilitates targeted improvements in segmentation algorithms and offers a more nuanced understanding of model performance across diverse applications.

Multimodal Deepfake Detection

The proliferation of synthetic media generated by advanced generative models such as GANs and diffusion-based architectures has raised significant concerns about misuse. Chapter 8 addressed this challenge by presenting a multimodal setting for deepfake detection and analysis, in which real and generated images sharing the same semantics are paired into semantic clusters. In our setting,

different semantic projections of a given image, expressed through captions, are employed to generate fake images. Employing the popular Stable Diffusion model as a generator, we investigated the performance of contrastive and classification-based visual features, highlighting that diffusion-based deepfakes share common low-level features that make them easily identifiable. Further, we proposed an approach to disentangle semantic and perceptual information, based on supervised contrastive learning. Under this setting, we investigated the classification of authenticity in a semantic space in which low-level cues left by the generator are removed, thus tackling a more challenging scenario. As a complementary contribution, we also collected and released the COCOFake dataset, containing about 1.2M images generated from COCO using both Stable Diffusion 1.4 and 2.0. Our findings demonstrated that combining multimodal features can enhance detection robustness while addressing emerging challenges posed by increasingly realistic generative models.

Multimodal Video Question Answering

In Chapter 9, we extended our exploration into video understanding by introducing T-Former, a novel question-guided temporal querying Transformer designed to address the challenges of dense video processing for multiple-choice video QA tasks. Our integrated framework, named PQR, combines T-Former with Large Language Models (LLMs) as reasoning agents, significantly outperforming other temporal modeling methods. Our approach also achieves substantial improvements over existing LLM-based models across four challenging video QA benchmarks while being trained solely on the target datasets without requiring any video-based pre-training phase. Through extensive experimental evaluations, we showcased the effectiveness of our approach. By leveraging compact, question-guided temporal tokens, we successfully bridge the gap between the complex temporal dynamics in videos and the reasoning capabilities of LLMs.

Future Works and Open Problems

In light of the comprehensive exploration and contributions detailed in this thesis, several promising avenues for future research and open problems emerge. These directions not only aim to extend the current work but also address the evolving challenges in the field of multimodal attentive deep learning architectures for visual-semantic understanding.

Firstly, while ViTs have demonstrated significant potential in modeling long-range dependencies, further research is needed to enhance their scalability and efficiency. Future work could explore more advanced sampling techniques to optimize computational resources without compromising performance. Additionally, developing more robust self-supervised pre-training strategies that can generalize across diverse datasets and tasks remains an open challenge. This includes exploring novel objectives that capture richer semantic information and effectively bridge the gap between pre-training and fine-tuning phases.

The integration of classical Computer Vision techniques with modern Transformer based segmentation architectures has shown to be effective in improving boundary precision and spatial reasoning. However, there is potential to further refine these integrations by exploring faster and more efficient superpixel algorithms that dynamically adjust to varying image resolutions, complexities, and contexts. This could lead to more precise segmentation outcomes, particularly in scenarios with rare or ambiguous classes.

In the realm of multimodal learning, expanding the applicability of open-vocabulary semantic segmentation models to a wider range of domains presents a significant opportunity. Future research could focus on enhancing domain adaptation techniques and exploring unsupervised or semi-supervised learning frameworks that minimize reliance on annotated data. Moreover, addressing the explainability and interpretability of state-of-the-art semantic segmentation models is crucial. Developing methods to retrieve reliable explanations for segmentation errors in a model-agnostic manner is a fundamental research direction.

The challenges associated with deepfake detection continue to evolve as generative models become more sophisticated. Future work could investigate advanced multimodal detection frameworks that incorporate additional modalities such as audio or contextual metadata to improve detection accuracy. Moreover, addressing ethical considerations and developing proactive measures to counteract misuse remain critical areas for ongoing research.

In video understanding, the development of more efficient temporal modeling techniques is essential for handling the vast amounts of data inherent in video content. Extending the capabilities of models like T-Former to support real-time processing and reasoning over longer video sequences could significantly enhance their applicability in practical scenarios. Additionally, exploring the integration of external knowledge sources or commonsense reasoning modules into video QA systems may further improve their ability to handle complex queries and dynamic content.

Final Remarks

This thesis has addressed challenges ranging from low-level feature extraction to high-level reasoning across modalities. By bridging pixels to reasoning through innovative methodologies, this work contributes to advancing artificial intelligence's ability to perceive, interpret, and reason complex data in a human-like manner. The findings presented here not only push the boundaries of what is possible with current AI systems but also lay a strong foundation for future advancements in visual-semantic understanding. As AI continues to evolve, we hope that this research will inspire further innovation at the intersection of vision, language, and reasoning, paving the way toward more intelligent, ethical, and impactful AI systems.

Publications, Achievements, and Acknowledgments

The majority of the works presented in this thesis have been published in international conferences and journals. A detailed list of publications is available in Appendix A. Throughout my PhD journey, I have had the privilege of engaging in several enriching research experiences and collaborations that have shaped my academic and professional development. My internship experiences at NVIDIA and LMU University provided invaluable opportunities to explore innovative research directions, culminating in publications presented at international conferences.

As an ELLIS PhD student, I had the privilege of being part of an international network of researchers at the forefront of artificial intelligence. This experience not only deepened my expertise but also fostered meaningful collaborations within a vibrant research community. My efforts have been recognized through several prestigious awards, which I am deeply honored to have received. These include the Outstanding Reviewer Award at ECCV 2024, the Best Paper Award at ICIAP 2021, and the Best Poster Award at CoNEXT 2020.

Finally, I would like to express my gratitude to everyone who have supported me throughout this transformative journey—my advisors, mentors, colleagues, friends, and family. This thesis is not just a culmination of years of hard work but also a testament to the collective efforts of all those who have walked this path with me. To everyone who I care about and who has played a role—big or small—in this journey, I am profoundly grateful.

Appendix A

List of Publications

The following list of publications includes all conference papers, journal articles, and book chapters published during my Ph.D., as well as pre-prints. An asterisk (*) denotes equal contribution. 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. Roberto Amoroso*, Gengyuan Zhang*, Rajat Koner, Lorenzo Baraldi, Rita Cucchiara, and Volker Tresp. Perceive, query & reason: Enhancing video qa with question-guided temporal queries. In *Winter Conference on Applications of Computer Vision (WACV)*, 2025.

2. Lorenzo Baraldi*, Roberto Amoroso*, Marcella Cornia, Andrea Pilzer, Lorenzo Baraldi, and Rita Cucchiara. Learning to mask and permute visual tokens for vision transformer pre-training. In *Computer Vision and Image Understanding (CVIU)*, 2025.

3. Luca Barsellotti*, Roberto Amoroso*, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Training-free open-vocabulary segmentation with offline diffusion-augmented prototype generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

4. Luca Barsellotti*, Roberto Amoroso*, Lorenzo Baraldi, and Rita Cucchiara. Fossil: Free open-vocabulary semantic segmentation through synthetic references retrieval. In *Winter Conference on Applications of Computer Vision (WACV)*, 2024.

5. Maximilian Bernhard, Roberto Amoroso, Yannic Kindermann, Lorenzo Baraldi, Rita Cucchiara, Volker Tresp, and Schubert Matthias. What's outside the intersection? fine-grained error analysis for semantic segmentation beyond iou. In *Winter Conference on Applications of Computer Vision (WACV)*, 2024.

6. Roberto Amoroso*, Davide Morelli*, Marcella Cornia, Lorenzo Baraldi, Alberto Del Bimbo, and Rita Cucchiara. Parents and children: Distinguishing multimodal deepfakes from natural images. In *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)*, 2024.

7. Roberto Amoroso, Michele Fenzi, Francesco Ferroni, Niloofar Gheissari, Despoina Paschalidou, Laura Leal-Taixé, and Elmar Haussmann. Video search: A large-scale video-text retrieval system for AV. In *NTECH*, 2024. ***Oral Presentation***.

8. Roberto Amoroso, Matteo Tomei, Lorenzo Baraldi, and Rita Cucchiara. Superpixel positional encoding to improve vit-based semantic segmentation models. In *British Machine Vision Conference (BMVC)*, 2023.

9. Luca Barsellotti, Roberto Amoroso, Lorenzo Baraldi, and Rita Cucchiara. Enhancing open-vocabulary semantic segmentation with prototype retrieval. In *International Conference on Image Analysis and Processing (ICIAP)*, 2023.

10. Roberto Amoroso, Lorenzo Pappone, and Flavio Esposito. A federated learning approach to traffic matrix estimation using super-resolution techniques. In *IEEE Consumer Communications and Networking Conference (CCNC)*, 2023.

11. Paolo Bruno, Roberto Amoroso, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. Investigating bidimensional downsampling in vision transformer models. In *International Conference on Image Analysis and Processing (ICIAP)*, 2022. ***Best Paper Award***.

12. Roberto Amoroso, Lorenzo Baraldi, and Rita Cucchiara. Assessing the role of boundary-level objectives in indoor semantic segmentation. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2021. ***Oral Presentation***.

13. Roberto Amoroso, Lorenzo Baraldi, and Rita Cucchiara. Improving indoor semantic segmentation with boundary-level objectives. In *International Work-Conference on Artificial Neural Networks (IWANN)*, 2021. ***Oral Presentation***.

14. Roberto Amoroso, Flavio Esposito, and Maria Luisa Merani. Estimation of traffic matrices via super-resolution and federated learning. In *International Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*, 2020. ***Best Poster Award***.

Bibliography

[1] 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*, 2012. 17, 62, 67, 75, 77, 95

[2] 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. *Advances in Neural Information Processing Systems*, 2022. 150, 166

[3] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 91, 92, 95

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the International Conference on Computer Vision*, 2021. 23

[5] 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. 14, 37, 38, 47, 52, 53, 54, 112

[6] Manuele Barraco, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. With a little help from your own past: Prototypical memory networks for image captioning. In *Proceedings of the International Conference on Computer Vision*, 2023. 130

[7] Wanda Benesova and Michal Kottman. Fast superpixel segmentation using morphological processing. In *Proceedings of the Conference on Machine Vision and Machine Learning*, 2014. 17

[8] Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019. 31

[9] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Semantic segmentation with boundary neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 16

[10] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. In *Proceedings of the European Conference on Computer Vision*, 2020. 19

[11] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Proceedings of the European Conference on Computer Vision*, 2014. 54

[12] 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. 14

[13] Miles Brundage, Shahar Avin, Jack Clark, Helen Toner, Peter Eckersley, Ben Garfinkel, Allan Dafoe, Paul Scharre, Thomas Zeitzoff, Bobby Filar, et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228*, 2018. 123

[14] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the" video" in video-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 23, 166

[15] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 1995. 134

[16] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 81, 87, 90, 92, 109, 114

[17] Davide Caffagni, Manuele Barraco, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Synthcap: Augmenting transformers with synthetic data

for image captioning. In *Proceedings of the International Conference on Image Analysis and Processing*, 2023. 130

[18] Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. The (r)evolution of multimodal large language models: A survey. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2024. 149

[19] 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. 13, 25

[20] 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. 14, 93

[21] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 18, 80, 88, 90, 91, 92

[22] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 116

[23] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *Proceedings of the European Conference on Computer Vision*, 2020. 20

[24] Keshigeyan Chandrasegaran, Ngoc-Trung Tran, and Ngai-Man Cheung. A closer look at fourier spectrum discrepancies for cnn-generated images detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 20

[25] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 102, 115

[26] Liang-Chieh Chen, Jonathan T Barron, George Papandreou, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 16

[27] 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*, 2017. 16, 19, 25

[28] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 16, 111, 112

[29] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, 2018. 16, 111, 112, 116

[30] Liangyu Chen, Tong Yang, Xiangyu Zhang, Wei Zhang, and Jian Sun. Points as queries: Weakly semi-supervised object detection by points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 19

[31] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the International Conference on Machine Learning*, 2020. 14

[32] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*, 2020. 14

[33] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *Proceedings of the International Conference on Learning Representations*, 2023. 150

[34] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision*, 2023. 14, 38, 47, 52, 53, 54

[35] 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. 88

[36] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 14

[37] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision*, 2021. 14

[38] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *Proceedings of the International Conference on Learning Representations*, 2023. 19

[39] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 18, 19, 102, 109, 119

[40] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 103, 111, 112

[41] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 2021. 103, 111, 112

[42] Harry Cheng, Yangyang Guo, Tianyi Wang, Qi Li, Xiaojun Chang, and Liqiang Nie. Voice-face homogeneity tells deepfake. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2023. 20

[43] Robert Chesney and Danielle Citron. Deepfakes and the new disinformation war: The coming age of post-truth geopolitics. *Foreign Affairs*, 2019. 123

[44] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 13, 26

[45] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *Advances in Neural Information Processing Systems*, 2021. 16

[46] 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. *Journal of Machine Learning Research*, 2024. 24

[47] Federico Cocchi, Lorenzo Baraldi, Samuele Poppi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Unveiling the impact of image transformations on deepfake detection: An experimental analysis. In *Proceedings of the International Conference on Image Analysis and Processing*, 2023. 145

[48] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 67, 75, 76, 78, 81, 87, 90, 93, 109, 114

[49] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Explaining transformer-based image captioning models: An empirical analysis. *AI Communications*, 2021. 13

[50] Marcella Cornia, Lorenzo Baraldi, Giuseppe Fiameni, and Rita Cucchiara. Universal captioner: Long-tail vision-and-language model training through content-style separation. *arXiv preprint arXiv:2111.12727*, 2021. 13

[51] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *International Conference on Acoustics, Speech, and Signal Processing*, 2023. 21, 135

[52] Davide Cozzolino, Diego Gragnaniello, Giovanni Poggi, and Luisa Verdoliva. Towards universal gan image detection. In *Proceedings of the*

International Conference on Visual Communications and Image Processing, 2021. 20

[53] Davide Cozzolino, Andreas Rössler, Justus Thies, Matthias Nießner, and Luisa Verdoliva. Id-reveal: Identity-aware deepfake video detection. In *Proceedings of the International Conference on Computer Vision*, 2021. 20

[54] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018. 20

[55] Gabriela Csurka, Diane Larlus, Florent Perronnin, and France Meylan. What is a good evaluation measure for semantic segmentation?. In *Proceedings of the British Machine Vision Conference*, 2013. 18

[56] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, 2020. 31

[57] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, 2023. 24, 150, 153, 154, 157, 158

[58] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 101

[59] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2009. 38, 45, 48, 49, 54, 91

[60] 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 North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, 2019. 12, 14, 25, 37, 39, 47

[61] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, 2021. 21, 123

[62] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 17, 18, 80

[63] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In *Advances in Neural Information Processing Systems*, 2021. 124

[64] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the International Conference on Computer Vision*, 2015. 14

[65] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020. 22

[66] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Peco: Perceptual codebook for bert pre-training of vision transformers. In *Proceedings of the Conference on Artificial Intelligence*, 2023. 14, 53

[67] 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. 13, 14, 25, 26, 29, 31, 39, 47, 52, 62, 64, 125, 133

[68] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 20, 124

[69] Tarik Dzanic, Karan Shah, and Freddie Witherden. Fourier spectrum discrepancies in deep network generated images. In *Advances in Neural Information Processing Systems*, 2020. 20

[70] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021. 14, 38, 53

[71] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in Neural Information Processing Systems*, 2021. 16

[72] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2012 (voc2012) results, 2012. 81, 87, 90, 92, 93, 109, 114

[73] Yuxin Fang, Li Dong, Hangbo Bao, Xinggang Wang, and Furu Wei. Corrupted image modeling for self-supervised visual pre-training. In *Proceedings of the International Conference on Learning Representations*, 2023. 14

[74] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 157

[75] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 2004. 17, 86, 88, 95, 96

[76] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *International Conference on Acoustics, Speech, and Signal Processing*, 2022. 145

[77] Gianni Franchi, Nacim Belkhir, Mai Lan Ha, Yufei Hu, Andrei Bursuc, Volker Blanz, and Angela Yao. Robust semantic segmentation with superpixel-mix. *Proceedings of the British Machine Vision Conference*, 2021. 17

[78] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *Proceedings of the International Conference on Machine Learning*, 2020. 20, 124

[79] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 16

[80] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *Proceedings of the European Conference on Computer Vision*, 2022. 124

[81] Difei Gao, Luowei Zhou, Lei Ji, Linchao Zhu, Yi Yang, and Mike Zheng Shou. Mist: Multi-modal iterative spatial-temporal transformer for long-form video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 23, 166

[82] 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. 17, 80

[83] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Proceedings of the International Conference on Learning Representations*, 2018. 14

[84] Sharath Girish, Saksham Suri, Sai Saketh Rambhatla, and Abhinav Shrivastava. Towards discovery and attribution of open-world gan generated images. In *Proceedings of the International Conference on Computer Vision*, 2021. 20

[85] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 20

[86] Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *Proceedings of the International Conference on Machine Learning*, 2020. 13, 35

[87] Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006. 17

[88] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Are gan generated images easy to detect? a critical analysis of the state-of-the-art. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2021. 20

[89] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent-a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020. 14

[90] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 23

[91] Zhihao Gu, Yang Chen, Taiping Yao, Shouhong Ding, Jilin Li, Feiyue Huang, and Lizhuang Ma. Spatiotemporal inconsistency learning for deepfake video detection. In *Proceedings of the ACM International Conference on Multimedia*, 2021. 20

[92] Bing Han, Xiaoguang Han, Hua Zhang, Jingzhi Li, and Xiaochun Cao. Fighting fake news: Two stream network for deepfake detection via learnable srm. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2021. 20

[93] Bing Han, Jianshu Li, Wenqi Ren, Man Luo, Jian Liu, and Xiaochun Cao. Sigma-df: Single-side guided meta-learning for deepfake detection. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, 2023. 20

[94] Douglas Harris. Deepfakes: False pornography is here and the law cannot protect you. *Duke Law & Technology Review*, 2018. 123

[95] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 16

[96] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 14, 38

[97] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 14

[98] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the International Conference on Computer Vision*, 2017. 25

[99] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 16, 25, 28, 125, 133

[100] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 2022. 80

[101] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 31

[102] AD Hoover, Valentina Kouznetsova, and Michael Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical imaging*, 2000. 109, 114

[103] Zejiang Hou, Fei Sun, Yen-Kuang Chen, Yuan Xie, and Sun-Yuan Kung. Milan: Masked image pretraining on language assisted representation. *arXiv preprint arXiv:2208.06049*, 2022. 15

[104] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 13

[105] Zhongwen Hu, Qin Zou, and Qingquan Li. Watershed superpixel. In *Proceedings of the International Conference on Image Processing*, 2015. 95

[106] Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 14, 38

[107] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 13, 16

[108] Ziyuan Huang, Shiwei Zhang, Liang Pan, Zhiwu Qing, Mingqian Tang, Ziwei Liu, and Marcelo H Ang Jr. Tada! temporally-adaptive convolutions for video understanding. *Proceedings of the International Conference on Learning Representations*, 2022. 150

[109] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 103, 111, 112, 113, 114

[110] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masked transformers for semantic segmentation. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2023. 62, 64

[111] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision*, 2018. 17

[112] 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. 17, 80

[113] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 21

[114] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 2021. 13

[115] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019. 88

[116] Matthew Joslin and Shuang Hao. Attributing and detecting fake images generated by known gans. In *Proceedings of the IEEE Security and Privacy Workshops*, 2020. 20

[117] Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion models for open-vocabulary segmentation. *Proceedings of the European Conference on Computer Vision*, 2024. 18, 90, 91

[118] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 130

[119] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 20

[120] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 20

[121] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning*, 2020. 13, 26

[122] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys*, 2021. 12

[123] Mahyar Khayatkhoei and Ahmed Elgammal. Spatial frequency bias in convolutional generative adversarial networks. In *Proceedings of the Conference on Artificial Intelligence*, 2022. 20

[124] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, 2020. 129

[125] Beomyoung Kim, Sangeun Han, and Junmo Kim. Discriminative region suppression for weakly-supervised semantic segmentation. In *Proceedings of the Conference on Artificial Intelligence*, 2021. 102, 115

[126] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018. 20

[127] Dohwan Ko, Ji Soo Lee, Wooyoung Kang, Byungseok Roh, and Hyunwoo J Kim. Large language models are temporal and causal reasoners for video question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023. 166

[128] Pushmeet Kohli, L'ubor Ladickỳ, and Philip HS Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 2009. 19

[129] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 2022. 24

[130] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the International Conference on Computer Vision*, 2013. 54

[131] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009. 26, 30, 32, 33, 35

[132] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Neural reasoning, fast and slow, for video question answering. In *International Joint Conference on Neural Networks*, 2020. 23

[133] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 2021. 102, 115

[134] Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2023. 23, 150

[135] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 23

[136] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018. 23

[137] Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. What is more likely to happen next? video-and-language future event prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2020. 23, 157

[138] 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. 80

[139] Jing Li, Junsong Fan, and Zhaoxiang Zhang. Towards noiseless object contours for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 102, 115

[140] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Proceedings of the International Conference on Machine Learning*, 2023. 24, 150, 153, 154, 158, 166

[141] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2020. 157

[142] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *Proceedings of the International Conference on Computer Vision*, 2019. 16

[143] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 13

[144] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In *Proceedings of the Conference on Artificial Intelligence*, 2019. 23

[145] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021. 19

[146] Yicong Li, Junbin Xiao, Chun Feng, Xiang Wang, and Tat-Seng Chua. Discovering spatio-temporal rationales for video question answering. In *Proceedings of the International Conference on Computer Vision*, 2023. 23, 166

[147] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 124

[148] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 21, 124

[149] Zhengqin Li and Jiansheng Chen. Superpixel segmentation using linear spectral clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 17

[150] 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. 18, 80

[151] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 17, 18

[152] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 16

[153] 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. 81, 88, 127, 130

[154] Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*, 2023. 24

[155] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 149

[156] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 2023. 150

[157] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2011. 17, 62

[158] 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. 80

[159] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems*, 2021. 37

[160] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 14

[161] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 14, 38

[162] 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. *Proceedings of the International Conference on Computer Vision*, 2021. 13, 16

[163] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 16, 61

[164] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002. 83

[165] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, 2019. 31, 48, 134

[166] Yuhang Lu and Touradj Ebrahimi. Assessment framework for deepfake detection in real-world situations. *EURASIP Journal on Image and Video Processing*, 2024. 145

[167] 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. 80, 91, 92

[168] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *Neurocomputing*, 2022. 150

[169] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 20

[170] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2024. 149

[171] Vaïa Machairas, Etienne Decencière, and Thomas Walter. Waterpixels: Superpixels based on the watershed transformation. In *Proceedings of the International Conference on Image Processing*, 2014. 17

[172] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 54

[173] 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*, 2018. 96

[174] Sara Mandelli, Nicolò Bonettini, Paolo Bestagini, and Stefano Tubaro. Detecting gan-generated images by orthogonal training of multiple cnns. In *Proceedings of the International Conference on Image Processing*, 2022. 20, 135, 146, 147

[175] Francesco Marra, Diego Gragnaniello, Davide Cozzolino, and Luisa Verdoliva. Detection of gan-generated fake images over social networks. In *Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval*, 2018. 20

[176] Fernand Meyer. Color image segmentation. In *Proceedings of the International Conference on Image Processing*, 1992. 76, 77

[177] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, 2019. 13

[178] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 20

[179] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2014. 81, 87, 90, 92

[180] 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. 17

[181] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *Proceedings of the International Conference on Learning Representations*, 2022. 21

[182] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the International Conference on Computer Vision*, 2015. 61

[183] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision*, 2016. 14

[184] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 14

[185] 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*, 2024. 81, 84, 88, 93

[186] Zizheng Pan, Bohan Zhuang, Jing Liu, Haoyu He, and Jianfei Cai. Scalable vision transformers with hierarchical pooling. In *Proceedings of the International Conference on Computer Vision*, 2021. 13, 26, 28, 30, 35, 62, 64

[187] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional

network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 16

[188] Liang Peng, Shuangji Yang, Yi Bin, and Guoqing Wang. Progressive graph attention network for video question answering. In *Proceedings of the ACM International Conference on Multimedia*, 2021. 23

[189] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 15, 38, 45, 46, 48, 52, 53, 54, 56, 57

[190] Samuele Poppi, Tobia Poppi, Federico Cocchi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Safe-clip: Removing nsfw concepts from vision-and-language models. *Proceedings of the European Conference on Computer Vision*, 2024. 123

[191] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, 2021. 15, 17, 23, 45, 80, 86, 88, 89, 93, 125, 130, 134, 137, 153

[192] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. 12, 25

[193] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 21, 123, 124

[194] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the International Conference on Machine Learning*, 2021. 14, 38, 56, 57, 124

[195] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the International Conference on Computer Vision*, 2021. 16, 61, 62, 63, 64, 69, 75, 76, 77

[196] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 25

[197] 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. 80, 91, 92

[198] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 25

[199] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Proceedings of the International Conference on Computer Vision*, 2003. 17, 62

[200] Jonas Ricker, Simon Damm, Thorsten Holz, and Asja Fischer. Towards the detection of diffusion model deepfakes. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2024. 21

[201] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 18, 21, 80, 83, 88, 123, 124, 127, 130

[202] Shenghai Rong, Bohai Tu, Zilei Wang, and Junjie Li. Boundary-enhanced co-training for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 115, 116

[203] 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. 83

[204] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the International Conference on Computer Vision*, 2019. 20, 21, 124

[205] 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*, 2015. 13, 26, 30, 34, 35, 133

[206] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 2022. 21, 123, 124

[207] Sara Sarto, Manuele Barraco, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Positive-augmented contrastive learning for image and video captioning evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 130

[208] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 123

[209] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, 2022. 130, 134

[210] Christoph Schuhmann, Robert Kaczmarczyk, Aran Komatsuzaki, Aarush Katta, Richard Vencu, Romain Beaumont, Jenia Jitsev, Theo Coombes, and Clayton Mullis. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *Proceedings of Neural Information Processing Systems Workshops*, 2021. 134

[211] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image diffusion models. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2023. 21

[212] Gyungin Shin, Weidi Xie, and Samuel Albanie. Reco: Retrieve and co-segment for zero-shot transfer. In *Advances in Neural Information Processing Systems*, 2022. 90, 91, 92

[213] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015. 25, 28

[214] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, 2020. 14, 38, 42

[215] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: A survey on deep learning-based image captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 12

[216] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the International Conference on Computer Vision*, 2021. 111, 112

[217] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 2018. 17, 77

[218] Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, et al. Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2022. 19

[219] 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. 18, 80, 83

[220] Yehui Tang, Yunhe Wang, Yixing Xu, Dacheng Tao, Chunjing Xu, Chao Xu, and Chang Xu. Scop: Scientific control for reliable neural network pruning. In *Advances in Neural Information Processing Systems*, 2020. 13, 26, 34, 35

[221] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, Ali Vosoughi, Chao Huang, Zeliang Zhang, Feng Zheng, Jianguo Zhang, Ping Luo, Jiebo Luo, and Chenliang Xu. Video understanding with large language models: A survey. *arXiv preprint arXiv:2312.17432*, 2023. 149

[222] 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. 24

[223] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 23

[224] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning*, 2021. 13, 25, 31, 34, 35

[225] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *Proceedings of the European Conference on Computer Vision*, 2022. 93

[226] 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. 158

[227] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In *Advances in Neural Information Processing Systems*, 2020. 20

[228] Michael Van den Bergh, Xavier Boix, Gemma Roig, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. *International Journal of Computer Vision*, 2015. 77, 95

[229] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008. 143

[230] 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. 12, 16, 25, 27, 66, 133, 155

[231] Luisa Verdoliva. Media forensics and deepfakes: an overview. *IEEE Journal of Selected Topics in Signal Processing*, 2020. 20

[232] Jie Wang and Xiaoqiang Wang. Vcells: Simple and efficient superpixels using edge-weighted centroidal voronoi tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 17

[233] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 16

[234] Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Kevin Qinghong Lin, Satoshi Tsutsui, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, et al. All in one: Exploring unified video-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 23, 166

[235] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2019. 27

[236] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 20, 146, 147

[237] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the International Conference on Computer Vision*, 2021. 13, 16

[238] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. 102, 115

[239] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 16

[240] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the International Conference on Computer Vision*, 2015. 14

[241] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 23, 166

[242] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 115, 116

[243] Syed Waqas Zamir, Aditya Arora, Akshita Gupta, Salman Khan, Guolei Sun, Fahad Shahbaz Khan, Fan Zhu, Ling Shao, Gui-Song Xia, and Xiang Bai. isaid: A large-scale dataset for instance segmentation in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 109, 114

[244] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 14, 38, 53

[245] Longhui Wei, Lingxi Xie, Wengang Zhou, Houqiang Li, and Qi Tian. Mvp: Multimodality-guided visual pre-training. In *Proceedings of the European Conference on Computer Vision*, 2022. 14, 15, 53

[246] David Weikersdorfer, David Gossow, and Michael Beetz. Depth-adaptive superpixels. In *Proceedings of the International Conference on Pattern Recognition*, 2012. 17

[247] Moritz Wolter, Felix Blanke, Raoul Heese, and Jochen Garcke. Wavelet-packets for deepfake image analysis and detection. *Machine Learning*, 2022. 21

[248] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 134

[249] Bo Wu, Shoubin Yu, Tenenbaum Joshua B Chen, Zhenfang, and Chuang Gan. Star: A benchmark for situated reasoning in real-world videos. In *Advances in Neural Information Processing Systems*, 2021. 23, 157

[250] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the International Conference on Computer Vision*, 2021. 62, 64

[251] 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. *Proceedings of the International Conference on Computer Vision*, 2023. 18, 80

[252] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 23, 157

[253] Junbin Xiao, Pan Zhou, Tat-Seng Chua, and Shuicheng Yan. Video graph transformer for video question answering. In *Proceedings of the European Conference on Computer Vision*, 2022. 23

[254] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision*, 2018. 49, 112

[255] Enze Xie, Wenhai Wang, Zhiding Yu, Anima 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. 16, 61, 62, 63, 64, 69, 71, 73, 75, 78, 111, 112

[256] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: a simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 14

[257] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the ACM International Conference on Multimedia*, 2017. 23

[258] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 17, 18, 80, 90, 91, 92

[259] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 18, 116

[260] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 18, 80, 91, 92

[261] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 17

[262] 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. 80

[263] Peng Xu, Xiatian Zhu, and David A Clifton. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 149

[264] Zhiwei Xu, Thalaiyasingam Ajanthan, and Richard Hartley. Refining semantic segmentation with superpixel by transparent initialization and sparse encoder. *arXiv preprint arXiv:2010.04363*, 2020. 17

[265] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the International Conference on Computer Vision*, 2021. 23

[266] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 2022. 23, 166

[267] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel segmentation with fully convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 17, 62

[268] Wenyuan Yang, Xiaoyu Zhou, Zhikai Chen, Bofei Guo, Zhongjie Ba, Zhihua Xia, Xiaochun Cao, and Kui Ren. Avoid-df: Audio-visual joint learning for detecting deepfake. *IEEE Transactions on Information Forensics and Security*, 2023. 20

[269] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, 2019. 14, 38, 41, 42, 44

[270] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *Proceedings of the International Conference on Learning Representations*, 2023. 24

[271] Qinghao Ye, Guohai Xu, Ming Yan, Haiyang Xu, Qi Qian, Ji Zhang, and Fei Huang. Hitea: Hierarchical temporal-aware video-language pre-training. In *Proceedings of the International Conference on Computer Vision*, 2023. 166

[272] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 16

[273] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 16

[274] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the International Conference on Computer Vision*, 2019. 20

[275] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 2024. 23, 150, 151, 153, 166

[276] Wenyi Yu, Changli Tang, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Connecting speech encoder and large language model for asr. In *International Conference on Acoustics, Speech, and Signal Processing*, 2024. 150

[277] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the Conference on Artificial Intelligence*, 2019. 23

[278] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the International Conference on Computer Vision*, 2021. 13

[279] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the International Conference on Computer Vision*, 2019. 31

[280] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, and Yifan Liu. Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems*, 2022. 17

[281] Gengyuan Zhang, Jisen Ren, Jindong Gu, and Volker Tresp. Multi-event video-text retrieval. In *Proceedings of the International Conference on Computer Vision*, 2023. 149

[282] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 16

[283] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023. 24, 149, 150, 154, 157

[284] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Muller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2022. 13

[285] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations*, 2018. 31

[286] 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. 86

[287] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 13

[288] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proceedings of the European Conference on Computer Vision*, 2016. 14

[289] Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. Ternarybert: Distillation-aware ultra-low bit bert. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2020. 13

[290] Xinyu Zhang, Jiahui Chen, Junkun Yuan, Qiang Chen, Jian Wang, Xiaodi Wang, Shumin Han, Xiaokang Chen, Jimin Pi, Kun Yao, Junyu Han, Errui

Ding, and Jingdong Wang. Cae v2: Context autoencoder with clip target. *Transactions on Machine Learning Research*, 2023. 15

[291] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *Proceedings of the International Workshop on Information Forensics and Security*, 2019. 20, 124

[292] Yifan Zhang, Bo Pang, and Cewu Lu. Semantic segmentation by early region proxy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 17

[293] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 80

[294] Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. Mmicl: Empowering vision-language model with multi-modal in-context learning. *Proceedings of the International Conference on Learning Representations*, 2024. 150

[295] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 16, 111, 112

[296] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision*, 2018. 16

[297] Anlin Zheng, Yuang Zhang, Xiangyu Zhang, Xiaojuan Qi, and Jian Sun. Progressive end-to-end object detection in crowded scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 19

[298] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 2023. 24, 158

[299] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao 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/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 13, 25, 61, 63, 68, 69, 111, 112

[300] Yaoyao Zhong, Junbin Xiao, Wei Ji, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022. 23, 149

[301] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the Conference on Artificial Intelligence*, 2020. 31

[302] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 48, 50, 67, 75, 78, 81, 87, 90, 93, 97, 109, 114

[303] 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*, 2019. 48, 50, 81, 87, 90, 93, 97

[304] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *Proceedings of the European Conference on Computer Vision*, 2022. 18, 80, 90, 91, 92

[305] Yizhou Zhou, Xiaoyan Sun, Dong Liu, Zhengjun Zha, and Wenjun Zeng. Adaptive pooling in multi-instance learning for web video annotation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017. 150

[306] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support Workshop*, 2018. 101

[307] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 1997. 134

[308] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *Proceedings of the International Conference on Learning Representations*, 2024. 150, 154

[309] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the International Conference on Computer Vision*, 2017. 20

[310] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *Proceedings of the International Conference on Learning Representations*, 2021. 13