UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

University of Modena and Reggio Emilia

XXXVIII cycle of the International Doctorate School in
Information and Communication Technologies (ICT)

Learning Across Time, Tasks, and Models: Knowledge Transfer in Evolving Systems

Aniello Panariello

Supervisor: Prof. Simone Calderara
PhD Course Coordinator: Prof. Luigi Rovati

Modena, A.Y. 2024/2025

Supervisor:

Prof. Simone Calderara — University of Modena and Reggio Emilia

Director of the Doctoral School:

Prof. Luigi Rovati — University of Modena and Reggio Emilia

Review Committee:

Prof. Andrew D. Bagdanov — University of Florence

Dr. Joost van de Weijer — Computer Vision Center
Universitat Autònoma de Barcelona

Tesi di dottorato finanziata dall'Unione europea - Next Generation EU, Missione 4, componente 2 "Dalla Ricerca all'Impresa" - Investimento 3.3 "Introduzione di dottorati innovativi che rispondono ai fabbisogni di innovazione delle imprese e promuovono l'assunzione dei ricercatori dalle imprese".

Abstract

With the growing accessibility of artificial intelligence and machine learning technologies, modern learning systems increasingly operate in dynamic environments, where data distributions, tasks, and objectives evolve over time. Traditional static learning paradigms struggle to keep pace with this evolution, often leading to degraded performance, loss of previously acquired knowledge, or costly retraining from scratch. Addressing these challenges requires learning mechanisms that can transfer, preserve, and recombine knowledge across evolving conditions, spanning sequential data, streams of tasks, and even collections of trained models.

This dissertation investigates how learning systems can evolve alongside their environments by leveraging structured information and prior knowledge at multiple scales. It follows a coherent progression that begins with temporal learning from data, extends to continual adaptation across tasks, and culminates in model composition and merging. Across these settings, the central question is how information acquired in one context can be reused or adapted in another without restarting the learning process.

The first part focuses on temporal learning from visual data, treating video streams as structured time series. It introduces a consistency-based formulation for weakly supervised temporal anomaly localization, showing how temporal coherence can compensate for missing frame-level supervision. It then recasts multi-object tracking as

conditional density estimation through probabilistic and flow-based modeling of data association (TrackFlow), before introducing monocular per-object distance estimation (DistFormer), which combines object-centric reasoning with Masked Object Modeling to learn geometric representations that remain robust under occlusions and domain shifts. Together, these contributions demonstrate how temporal and object-level structure can be exploited at increasing levels of granularity, from sequence-level regularities to identity persistence and geometric reasoning.

The second part investigates continual learning, where data arrive as a stream of evolving tasks. CHARON presents an efficient continual learning framework for skeleton-based action recognition that combines masking and compression to improve stability and memory efficiency. CGIL introduces a generative replay strategy in embedding space for continual prompt learning in large vision-language models, preserving zero-shot capabilities while enabling incremental adaptation. Collectively, these works reinterpret continual learning as structured temporal evolution in task space.

The final part explores knowledge transfer across models through model composition, merging, and transport. Instead of adapting a single model over time, this setting studies how multiple trained models or parameter-efficient updates can be combined to synthesize new capabilities. PASTA demonstrates scenario-specialized modular composition in parameter space for tracking, MoDER shows how class- and task-specialized textual experts can be recomposed to improve recognition of unseen classes, Core Space introduces an accurate and efficient low-rank framework for merging parameter-efficient updates, and GradFix addresses task-vector transport across different pretrained models through gradient-sign filtering. This reframes adaptation as evolution in model space rather than data space.

Overall, this dissertation presents a unified perspective on knowledge transfer in evolving systems. By connecting temporal learning, continual adaptation, and model composition, it frames learning as the manipulation of structured representations across time, tasks, and models. The resulting framework highlights the role of structure, modularity, and reuse in building scalable, adaptive, and resilient learning systems that not only operate in changing environments but also evolve with them.

Sommario

Con la crescente accessibilità delle tecnologie di intelligenza artificiale e di apprendimento automatico, i sistemi di apprendimento moderni operano sempre più spesso in ambienti dinamici, in cui le distribuzioni dei dati, i task e gli obiettivi evolvono nel tempo. I paradigmi di apprendimento statici tradizionali faticano a tenere il passo con questa evoluzione, portando spesso a un degrado delle prestazioni, alla perdita delle conoscenze precedentemente acquisite o a costosi riaddestramenti da zero. Affrontare queste sfide richiede meccanismi di apprendimento in grado di trasferire, preservare e ricombinare la conoscenza in condizioni in evoluzione, che spaziano dai dati sequenziali ai flussi di task, fino a collezioni di modelli addestrati.

Questa tesi indaga come i sistemi di apprendimento possano evolvere insieme ai loro ambienti sfruttando informazioni strutturate e conoscenza pregressa a più scale. Segue un percorso coerente che inizia con l'apprendimento temporale dai dati, si estende all'adattamento continuo tra task e culmina nella composizione e fusione di modelli. In questi contesti, la questione centrale è come le informazioni acquisite in un contesto possano essere riutilizzate o adattate in un altro senza riavviare il processo di apprendimento.

La prima parte si concentra sull'apprendimento temporale da dati visivi, trattando i flussi video come serie temporali strutturate. Introduce una formulazione basata sulla coerenza per la localizzazione temporale di anomalie in regime debolmente su-

pervisionato, mostrando come la coerenza temporale possa compensare l'assenza di supervisione a livello di *frame*. Riformula poi il *multi-object tracking* come un problema di stima di densità condizionata, tramite una modellazione probabilistica e basata su flussi dell'associazione tra oggetti (TrackFlow), prima di introdurre la stima monoculare della distanza per singolo oggetto (DistFormer), che combina ragionamento centrato sugli oggetti e *Masked Object Modeling* per apprendere rappresentazioni geometriche robuste a occlusioni e cambi di dominio. Nel loro insieme, questi contributi mostrano come la struttura temporale e quella a livello di oggetto possano essere sfruttate a livelli crescenti di granularità, dalle regolarità a livello di sequenza alla persistenza delle identità e al ragionamento geometrico.

La seconda parte indaga il *continual learning*, in cui i dati arrivano come un flusso di task in evoluzione. CHARON presenta un *framework* efficiente di continual learning per il riconoscimento di azioni basato su scheletri, che combina mascheramento e compressione per migliorare stabilità ed efficienza in memoria. CGIL introduce una strategia di *generative replay* nello spazio delle rappresentazioni per il *continual prompt learning* in grandi modelli visione-linguaggio, preservando le capacità *zero-shot* e consentendo al contempo un adattamento incrementale. Nel complesso, questi lavori reinterpretano il continual learning come un'evoluzione temporale strutturata nello spazio dei task.

La parte finale esplora il trasferimento di conoscenza tra modelli tramite composizione, fusione e trasporto nello spazio dei parametri. Invece di adattare un singolo modello nel tempo, questo contesto studia come più modelli addestrati o aggiornamenti *parameter-efficient* possano essere combinati per sintetizzare nuove capacità. PASTA dimostra una composizione modulare specializzata per scenario nel tracking, MoDER mostra come esperti testuali specializzati per classe o task possano essere ricombinati per migliorare il riconoscimento di classi mai osservate, Core Space introduce un framework accurato ed efficiente a basso rango per la fusione di aggiornamenti parameter-efficient, e GradFix affronta il trasporto di task vector tra diversi modelli pre-addestrati tramite filtraggio basato sul segno del gradiente. Questo riformula l'adattamento come un'evoluzione nello spazio dei modelli piuttosto che nello spazio dei dati.

Nel complesso, questa tesi presenta una prospettiva unificata sul trasferimento di conoscenza nei sistemi in evoluzione. Collegando apprendimento temporale, adattamento continuo e composizione dei modelli, inquadra l'apprendimento come la manipolazione di rappresentazioni strutturate attraverso tempo, task e modelli. Il

framework risultante mette in evidenza il ruolo di struttura, modularità e riuso nella costruzione di sistemi di apprendimento scalabili, adattivi e resilienti che non solo operano in ambienti in cambiamento, ma evolvono insieme a essi.

Contents

Introduction

1

Introduction

1.1 Learning in Evolving Systems

Modern machine learning systems are increasingly deployed in environments that evolve over time. Data distributions shift, tasks change, models are updated, and operational constraints limit the ability to retrain from scratch or store all historical data. As a result, learning systems must continuously adapt while preserving previously acquired knowledge, often under strict computational, memory, or privacy constraints.

A common thread underlying the three main parts of this dissertation is the presence of *non-stationarity*. Learning is performed in settings where some aspect of the problem changes over time, yet the system is expected to remain robust, efficient, and reusable. While *temporal learning*, *continual learning*, and *model merging* are traditionally studied as distinct research areas, this dissertation argues that they can be understood within a shared technical framework centered on *knowledge transfer under evolution*.

Across all these settings, the core challenge is not merely learning, but learning without starting over: how to reuse, preserve, and recombine knowledge as the learning context changes.

1.1.1 A unifying abstraction

Consider a parametric model $f(\cdot;\theta)$ operating on data drawn from a distribution p. Learning can be formalized as the minimization of an expected risk:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \; \mathbb{E}_{(x,y)\sim p} \left[\mathcal{L}(f(x;\theta), y) \right]. \tag{1.1}$$

In evolving settings, however, the data distribution p is not fixed. Instead, learning is exposed to a sequence of distributions $\{p_s\}_{s\in\mathcal{S}}$, where the index s may represent time steps, tasks, domains, or even different pretrained models. The central technical question addressed in this dissertation is how to update, adapt, or combine models across changes in p_s while preserving useful information acquired under previous distributions.

Across the three parts of the dissertation, evolution acts on different objects:

- in **temporal learning**, p_s varies implicitly through sequential structure in the input;
- in **continual learning**, p_s changes explicitly across tasks, with restricted access to past data;
- in **model merging**, p_s is fixed, but knowledge is distributed across multiple parameter vectors $\{\theta_s\}$.

Taken together, these settings describe a progression from evolution in the input stream, to evolution in the task sequence, and finally to evolution at the level of the model itself. Despite these differences, all settings require mechanisms to control interference, retain knowledge, and enable transfer across changing conditions.

1.1.2 Structured information as an inductive constraint

Throughout the dissertation, *structured information* is used as an inductive constraint to stabilize learning under evolution. Rather than relying on additional supervision, structure is embedded into the learning process through architectural choices, loss functions, or parameterizations.

In temporal domains, structure arises from ordering, persistence, and smoothness in sequential data. Let $\{x_t\}_{t=1}^{T}$ denote a sequence. Many objectives introduced in this dissertation can be interpreted as enforcing invariances of the form:

$$f(\mathcal{A}_1(x_{1:T});\theta) \approx f(\mathcal{A}_2(x_{1:T});\theta), \tag{1.2}$$

where $\mathcal{A}_1$ and $\mathcal{A}_2$ are transformations that preserve the underlying temporal semantics. Such constraints are closely related to self-supervised learning principles and exploit temporal coherence as a supervisory signal.

In object-centric and tracking settings, structure is expressed through entities and associations. Given a latent association variable z, learning often targets a conditional distribution

$$p(z \mid x_{1:t}), \tag{1.3}$$

where heterogeneous cues such as appearance, motion, and geometry interact jointly. Explicitly modeling these interactions, rather than assuming independence, improves robustness and generalization.

In continual learning and model merging, structure manifests at the level of the model parameters, but in fundamentally different ways. In continual learning, adaptation proceeds through standard optimization, while forgetting is mitigated by shaping the learning signal, for example via replay or regularization. In model merging, instead, adaptation is performed *without optimization*: multiple trained parameter vectors are combined algebraically to construct new models. Across the dissertation, these forms of structure act as anchors that limit interference under non-stationarity, enabling reuse and transfer without requiring full retraining.

1.1.3 Knowledge transfer across different applications

Despite the diversity of problem settings, the contributions in this dissertation repeatedly rely on a small set of recurring transfer mechanisms:

Consistency and invariance. Local invariances reduce the effective hypothesis space and stabilize learning. In temporal problems, this yields objectives that align predictions across time or views. In parameter space, similar ideas appear as constraints that limit deviations from previously learned solutions.

Memory and replay. When past distributions $p_{<s}$ are unavailable, transfer requires an explicit memory mechanism. Replay-based methods approximate the ideal joint objective

$$\sum_{s} \mathbb{E}_{(x,y)\sim p_s} \mathcal{L}(f(x;\theta), y), \tag{1.4}$$

either through stored samples or through generated representations. Operating in learned embedding spaces improves both efficiency and scalability.

Parameter-space composition. Model merging treats knowledge as a displacement in parameter space. Given models $\theta_i = \theta_0 + \tau_i$, composition constructs a new model as

$$\theta_c = \theta_0 + \sum_i \lambda_i \tau_i, \tag{1.5}$$

with suitable coefficients λ_i. This formulation reframes transfer as a purely algebraic operation and connects task arithmetic, low-rank adaptation, and modular fine-tuning.

1.1.4 Efficiency and constraints

Efficiency is treated as a first-class requirement across all parts of the dissertation. Temporal models must process long sequences, continual learners face memory and privacy constraints, and model merging is only viable if it avoids retraining and additional inference cost. These pressures motivate learning signals that are local, compact, and compositional.

1.1.5 Common evaluation principles

Although different metrics are used across parts, evaluation consistently targets robustness under change: stability over time, retention across tasks, and compositional generalization across models. Success is measured not only by peak performance, but by the ability to adapt without collapse or prohibitive computational cost.

Taken together, these principles motivate the organization of the dissertation into three parts, each addressing a different manifestation of evolution while relying on shared technical foundations.

1.2 Organization of the dissertation

This dissertation presents the research contributions of the candidate and his collaborators (*we* in the following), addressing the challenges of knowledge transfer in evolving systems through multiple settings and applications.

The dissertation is organized into three main parts, each focusing on a different aspect of knowledge transfer in evolving systems. At the beginning of each part, a background chapter introduces the relevant concepts. Each chapter within the parts corresponds to a published or submitted research work, with the candidate as one of the authors.

Part I - Learning Across Time The first part focuses on temporal learning from visual data, where video streams are treated as structured time series. It explores how temporal coherence and spatial structure can be leveraged to model dynamic phenomena.

- Chapter 2 provides an overview of temporal learning, self-supervised learning, and video understanding.
- Chapter 3 introduces a consistency-based framework for temporal anomaly localization that exploits self-supervision to identify irregular events in weakly labeled videos.
- Chapter 4 presents TrackFlow, a probabilistic formulation of multi-object tracking built on normalizing flows for modeling temporal dynamics in complex scenes.
- Chapter 5 addresses monocular per-object distance estimation through DistFormer, which integrates object-centric reasoning into a unified masking-based learning paradigm.

Part II - Learning Across Tasks The second part investigates continual and incremental learning, where models must acquire new knowledge from a sequence of tasks while preserving previously learned information. It focuses on efficiency, memory constraints, and transferability across evolving domains.

- Chapter 6 reviews continual learning, incremental learning, and prompt learning in vision-language models.
- Chapter 7 describes CHARON, a continual learning framework for skeleton-based action recognition that employs masking and compression to enhance stability and memory efficiency.
- Chapter 8 introduces CGIL, a continual generative training approach for incremental prompt learning in CLIP models, which preserves zero-shot capabilities while supporting incremental adaptation.

Part III - Learning Across Models The third part explores knowledge transfer across models through model merging and task arithmetic. It investigates how pretrained models can be combined and adapted to create new capabilities without full retraining.

- Chapter 9 surveys model merging, task arithmetic, and parameter-efficient fine-tuning techniques.
- Chapter 10 presents PASTA, a modular tracking framework that composes specialized model components in parameter space to generalize across domains.
- Chapter 11 introduces MoDER, a model merging technique that facilitates know-

ledge transfer between different architectures.

- Chapter 12 introduces Core Space, a framework for low-rank adaptation of pre-trained models that enables efficient fine-tuning on new tasks.
- Chapter 13 introduces GradFix, a method for stabilizing gradient updates in multi-task learning scenarios.

Part I

Learning through Time

2

Background on Temporal Learning from Video

This part of the dissertation considers *temporal learning from video*, in which observations are naturally organized as sequences and time provides a fundamental source of structure. Rather than treating video frames as independent samples, temporal learning explicitly exploits continuity, persistence, and dynamics to infer meaningful patterns and maintain coherent predictions over time.

Three application domains are considered: video anomaly detection, multi-object tracking, and distance estimation from monocular video. While anomaly detection operates at the level of global scene dynamics, tracking and distance estimation adopt an object-centric perspective, modeling temporal persistence and geometric consistency at the level of individual entities.

This chapter introduces the common temporal learning principles underlying these problems, presents the shared mathematical background, and details the datasets and evaluation protocols adopted throughout this part.

2.1 An informal overview

Before introducing the formal background, it is useful to ask: *what does it mean to learn from time?* Across these chapters, the model does not interpret isolated frames, but exploits continuity, persistence, and motion across nearby observations.

In the anomaly-detection chapter (Chapter 3), this means recognizing when the temporal evolution of a scene breaks its usual pattern. A violent event or an accident may be ambiguous in a single frame, but becomes easier to detect when the model can compare what happens before, during, and after the event.

In the tracking and distance-estimation chapters (Chapters 4 and 5), the same principle becomes object-centric. A tracker must recognize that a partially occluded pedestrian remains the same person across frames, while distance estimation benefits from how an object's appearance and geometry evolve over time. These examples motivate the formal treatment below: time acts as a structural constraint that supports learning under weak, noisy, or incomplete supervision.

2.2 Temporal learning from video

Let $x_{1:T} = \{x_1, \dots, x_T\}$ denote a video sequence, where each $x_t \in \mathcal{X}$ represents a frame or a short clip. Unlike static image understanding, video understanding must model dependencies across time, induced by physical continuity, object persistence, and structured dynamics.

A recurring assumption throughout this part is *temporal coherence*: under normal conditions, changes across time are smooth and predictable, whereas anomalies, identity switches, or geometric inconsistencies violate this regularity. This principle appears as global temporal regularity in anomaly detection, identity persistence in tracking, and geometric consistency in distance estimation.

Temporal learning therefore aims to extract stable representations from sequential observations, using time not merely as an ordering variable but as an inductive bias that constrains the hypothesis space.

2.3 Mathematical background

This section introduces the mathematical abstractions that underpin the temporal learning methods developed in the following chapters.

2.3.1 Sequential modeling

Many temporal learning problems can be described through latent variables evolving over time. Introducing latent states z_t that summarize relevant information at time t, a general factorization takes the form

$$p(x_{1:T}, z_{1:T}) = p(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \prod_{t=1}^{T} p(x_t \mid z_t), \tag{2.1}$$

which underlies a wide class of models, including hidden Markov models and state-space models. While the methods proposed in this dissertation do not explicitly instantiate this formulation, the assumptions of temporal dependence and state persistence inform the design of learning objectives, architectures, and inference strategies.

2.3.2 Weak supervision and multiple instance learning

In weakly supervised temporal problems, labels are provided at the sequence level rather than at the frame level. This setting can be formalized within the multiple instance learning (MIL) framework, in which a video is treated as a bag of instances.

Given instance-level anomaly scores $a_{1:T}$ and a video-level label $y \in \{0, 1\}$, MIL objectives typically enforce constraints of the form

$$\max_t a_t \approx y. \tag{2.2}$$

Despite its simplicity, this aggregation mechanism implicitly assumes that temporally adjacent instances exhibit correlated behavior, often combined with regularization terms encoding temporal sparsity or smoothness. This formulation highlights how global supervision can be leveraged to recover localized temporal structure despite the absence of explicit frame-level annotations.

2.3.3 Probabilistic modeling and density estimation

Probabilistic modeling plays a central role in multi-object tracking. Given a track history $\mathcal{T}$ and a candidate detection d, the association problem can be cast as estimating

$$p(d \mid \mathcal{T}), \tag{2.3}$$

which quantifies the compatibility between motion, appearance, and geometric cues.

Learning such conditional distributions allows replacing hand-crafted association costs with data-driven likelihoods, enabling principled handling of uncertainty.

2.3.4 Temporal consistency as a learning signal

A unifying principle across the considered tasks is temporal consistency. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ denote transformations that preserve the semantics of a sequence. Consistency-based objectives enforce

$$f(\mathcal{A}_1(x_{1:T})) \approx f(\mathcal{A}_2(x_{1:T})), \tag{2.4}$$

thereby exploiting temporal structure without requiring additional labels. This principle connects temporal learning in video to broader self-supervised learning frameworks.

2.4 Applications

2.4.1 Video anomaly detection

Video anomaly detection aims to identify temporal intervals that deviate from patterns observed during training. In realistic settings, only weak supervision is available, with videos labeled as normal or anomalous at the sequence level. Temporal context is therefore crucial, since the same visual pattern may be benign or abnormal depending on what precedes and follows it.

Formally, given a sequence $x_{1:T}$ and a video-level label y, the goal is to infer latent anomaly scores $a_{1:T}$, where a_t reflects the likelihood of an anomaly occurring at time t. This problem is inherently ill-posed and relies on assumptions such as rarity and temporal localization of anomalous events.

Datasets

We conduct experiments on:

- **XD-Violence** [279]: a large-scale multi-modal benchmark containing 4754 videos (approximately 217 hours), including 2405 violent and 2349 non-violent samples. The official split comprises 3954 training videos and 800 test videos. All experiments use RGB data only.

Evaluation metrics

Performance in video anomaly detection is evaluated using threshold-free metrics that are robust to severe class imbalance. Let $s_i \in \mathbb{R}$ denote the predicted anomaly score for video i, and let $y_i \in \{0, 1\}$ be the corresponding ground-truth label, where $y_i = 1$ indicates the presence of at least one anomalous event.

- **Area Under the ROC Curve (AUC).** The Receiver Operating Characteristic (ROC) curve plots the *True Positive Rate* (TPR) against the *False Positive Rate* (FPR) as the decision threshold τ applied to the anomaly scores s_i varies. Formally, for a given threshold τ,

$$\mathrm{TPR}(\tau) = \Pr(s_i \geq \tau \mid y_i = 1)\,, \qquad \mathrm{FPR}(\tau) = \Pr(s_i \geq \tau \mid y_i = 0)\,. \tag{2.5}$$

  The AUC is defined as the area under the ROC curve:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}\big(\mathrm{FPR}^{-1}(u)\big)\; du, \tag{2.6}$$

  and quantifies the ability of the model to rank anomalous videos higher than normal ones, independently of any specific operating threshold. Equivalently, AUC can be interpreted as the probability that a randomly chosen anomalous sample receives a higher anomaly score than a randomly chosen normal sample.
- **Average Precision (AP).** Average Precision summarizes the precision–recall curve obtained by varying the decision threshold over s_i. Let P_n and R_n denote precision and recall at the n-th threshold. AP is defined as:

$$\mathrm{AP} = \sum_n (R_n - R_{n-1}) P_n, \tag{2.7}$$

  which corresponds to a Riemann sum approximation of the area under the precision–

recall curve. Unlike AUC, AP is sensitive to class imbalance and emphasizes correct detection of rare anomalous samples, making it particularly informative in highly skewed datasets.

Metrics are computed at the video level; frame-level scores are used only for analysis.

2.4.2 Multi-object tracking

Multi-object tracking (MOT) addresses the problem of maintaining object identities across time. Given detections $D_t = \{d_t^1, \dots, d_t^{N_t}\}$ at each time step, tracking consists of associating detections across frames into trajectories.

Datasets

We evaluate tracking performance on:

- **MOTSynth** [72]: 764 synthetic full-HD sequences, each 1800 frames long, covering diverse urban scenarios.
- **MOT17** [183]: real-world pedestrian tracking benchmark with moderate crowd density.
- **MOT20** [57]: highly crowded real-world scenarios with severe occlusions.

Evaluation metrics

Multi-object tracking performance must jointly account for *detection accuracy* and *identity consistency over time*. To this end, modern evaluation protocols measure not only whether objects are localized correctly in individual frames, but also whether their identities are maintained consistently throughout the sequence.

Let IDTP, IDFP, and IDFN denote identity true positives, false positives, and false negatives, respectively. These quantities count correctly matched identities, identity mismatches, and missed identity associations across all frames.

- **IDF1.** IDF1 measures identity preservation over time and is defined as the harmonic mean of identity precision and identity recall:

$$\mathrm{IDF1} = \frac{2 \cdot \mathrm{IDTP}}{2 \cdot \mathrm{IDTP} + \mathrm{IDFP} + \mathrm{IDFN}}. \tag{2.8}$$

  A high IDF1 score indicates that tracked trajectories maintain consistent identities across frames, penalizing identity switches, fragmentations, and re-identification

failures. This metric is particularly sensitive to long-term tracking consistency and is therefore widely adopted in pedestrian tracking benchmarks.

- **HOTA.** Higher-Order Tracking Accuracy (HOTA) provides a unified measure of tracking performance by explicitly decomposing it into detection and association components. It is defined as

$$\mathrm{HOTA} = \sqrt{\mathrm{DetA} \cdot \mathrm{AssA}}, \tag{2.9}$$

where DetA (Detection Accuracy) quantifies how well objects are detected and localized in individual frames, while AssA (Association Accuracy) measures the quality of identity associations across time. By combining these two aspects multiplicatively, HOTA captures the trade-off between spatial accuracy and temporal consistency, offering a more balanced evaluation than metrics that focus on only one of these dimensions.

2.4.3 Distance estimation from monocular video

In contrast to anomaly detection and tracking, distance estimation introduces explicit geometric structure into temporal learning, providing a complementary signal that reduces ambiguity in object-centric reasoning under occlusions and crowded scenes. Distance estimation aims to infer metric object distances from monocular video by exploiting temporal cues such as motion consistency and object persistence. In this dissertation, distance estimation is studied in conjunction with tracking, enabling temporal stabilization of predictions.

Datasets

We evaluate distance estimation on:

- **NuScenes** [26]: 1000 urban driving scenes with dense 3D annotations.
- **MOTSynth** [72]: synthetic pedestrian sequences with 3D skeleton annotations.
- **KITTI** [84]: autonomous driving benchmark with LiDAR-based distance ground truth.

Evaluation metrics

Distance estimation quality is evaluated using complementary metrics that capture relative accuracy, absolute error, and robustness under challenging visibility conditions.

Let d_i and $\hat{d}_i$ denote the ground-truth and predicted distances for the i-th object, respectively, and let N be the number of evaluated instances.

- **Threshold accuracy ($\delta_{<\tau}$).** Threshold accuracy measures the proportion of predictions whose relative error falls below a predefined tolerance τ. It is defined as:

$$\delta_{<\tau} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1} \left(\max \left(\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i} \right) < \tau \right). \tag{2.10}$$

  This metric is scale-invariant and is widely adopted in monocular depth and distance estimation benchmarks, as it emphasizes relative correctness rather than absolute magnitude.

- **Root Mean Squared Error (RMSE).** RMSE quantifies the average absolute deviation between predicted and ground-truth distances:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{d}_i - d_i)^2}. \tag{2.11}$$

  By squaring the error, RMSE penalizes large deviations more strongly, making it sensitive to catastrophic failures and long-range distance errors.

- **Average Localization Precision (ALP$_{@\tau}$).** ALP$_{@\tau}$ measures the percentage of predictions whose absolute distance error is below a fixed threshold τ, expressed in meters:

$$\text{ALP}_{@\tau} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1} \left(|\hat{d}_i - d_i| < \tau \right). \tag{2.12}$$

  Unlike $\delta_{<\tau}$, this metric directly reflects performance in metric space and is particularly relevant for safety-critical applications such as autonomous driving.

- **Average Localization of Occluded Objects Error (ALOE).** ALOE evaluates robustness under partial or severe occlusions by restricting the evaluation to objects within a predefined occlusion range $\mathcal{O}$:

$$\text{ALOE} = \frac{1}{|\mathcal{O}|} \sum_{i \in \mathcal{O}} |\hat{d}_i - d_i|. \tag{2.13}$$

  This metric isolates the impact of visibility degradation on distance estimation performance and enables fine-grained analysis of failure modes in crowded or occluded scenes.

3

Consistency-based Self-supervised Learning for Temporal Anomaly Localization

3.1 Video anomaly detection

Video anomaly detection addresses the problem of identifying and localizing irregular events within a video stream. Such events often correspond to unusual or unsafe human activities, including violence, accidents, or illicit behaviors. Owing to the widespread deployment of surveillance systems, particularly CCTV cameras, this task has become increasingly relevant in real-world applications, where manual inspection of video data is impractical due to the sheer volume of data and automated analysis is required to make the process feasible.

Early approaches to video anomaly detection relied on low-level handcrafted features, extracted either from visual cues [55, 11, 118] or from object trajectories [28, 185].

Publication. **Aniello Panariello**, *et al.* *Consistency-Based Self-Supervised Learning for Temporal Anomaly Localization.* ECCVW, 2022 [194].

Candidate contribution. Idea, methodology, implementation, experiments, and writing.

While effective in controlled settings, these methods often exhibit limited robustness when confronted with complex scenes or unseen behaviors [107, 180]. As a result, research has gradually shifted toward learning-based formulations.

A prominent class of methods models normality by learning a representation of regular patterns from data containing only normal events. Under this paradigm, anomalies are detected as deviations from the learned model of regularity [93]. Deep reconstruction-based approaches, typically based on autoencoders or related architectures, have become particularly popular in this context [320, 268]. These methods often incorporate explicit regularization of the latent space to encourage compactness and stability of normal representations.

More recent work has explored the use of *weak supervision*, where anomalous videos are available during training but annotations are provided only at the video level [248]. In this setting, the model is informed about the presence of an anomaly somewhere in the sequence, without access to its temporal extent. This formulation naturally leads to multiple instance learning (MIL) frameworks [253, 279, 75], in which a video is treated as a bag of instances and additional constraints are introduced to infer frame-level anomaly scores from coarse labels. Common assumptions include temporal sparsity and smoothness of anomaly scores, which are enforced through regularization.

Within this line of research, temporal structure emerges as a key source of information. Recent advances in self-supervised learning and consistency regularization [246, 282, 42, 22] have shown that meaningful learning signals can be obtained by enforcing invariance across different views of the same data. In sequential domains, this principle can be extended by generating alternative views of a video sequence and encouraging the model to produce consistent predictions across them.

Following this perspective, the work presented in this chapter introduces a regularization strategy, based on consistency, tailored to video sequences. Instead of restricting temporal smoothness to adjacent frames, consistency is enforced across temporally distant but semantically related sub-sequences, leveraging temporal coherence as an implicit supervisory signal. This formulation complements existing MIL-based objectives and provides a principled way to exploit temporal structure under weak supervision.

The effectiveness of this approach is evaluated on the XD-Violence dataset [279], demonstrating that temporal consistency can significantly improve frame-level localization performance in weakly supervised anomaly detection settings.

3.2 Related work

Traditional video anomaly detection relied on handcrafted features or object trajectories [55, 11, 118, 28, 185], but these approaches lack robustness to unseen scenarios, motivating the shift to deep learning methods. Recent unsupervised methods use deep autoencoders to model normality and detect anomalies via reconstruction error [93, 1]. Due to the scarcity of fine-grained labels, weakly supervised approaches based on Multiple Instance Learning (MIL) [248] have become popular, using video-level labels and constraints such as sparsity and smoothness, with attention mechanisms further improving localization [320].

Temporal action localization methods are typically fully or weakly supervised, with one-stage [166, 152] and two-stage [284, 302, 154] pipelines; proposal generation uses anchors, sliding windows, or boundary detection [36, 78, 291, 238, 153, 154]. Weakly supervised temporal localization has been advanced by UntrimmedNet [265], STPN [189], and AutoLoc [239], and recently adapted to anomaly detection [278, 171]. The approach presented in this chapter builds on the weakly supervised MIL formulation, while introducing temporal consistency as an additional self-supervised regularization signal.

3.3 Proposed method

In the following, we first present the proposed model and then its training objective. The final part is dedicated to explaining the process of proposal generation, which consists of a post-processing step grouping adjacent similar scores into contiguous intervals.

3.3.1 Model

In our setup, each video is split into segments of 16 frames, with no overlap between consecutive segments. To extract video-level features, some works [300, 33] combine 2D CNNs and recurrent neural networks; instead, we feed each segment to a pretrained I3D network [32]. Indeed, the authors of [128] have shown that I3D features can be effective for video anomaly detection even when a shallow classifier such as XGBoost [41, 29] is used for downstream classification. In our case, the I3D network is pretrained on the Kinetics [114] dataset and not fine-tuned on our target data.

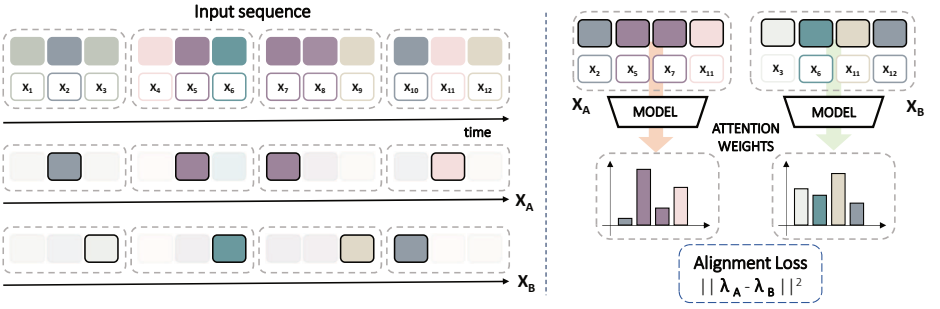


Figure 3.1: Overview of the proposed framework. (left) Augmentation function sampling two slightly different sequences out of a single one. The original sequence gets split into windows and for each, we randomly sample a single feature vector. This is done twice to obtain two sequences X^A and X^B. (right) Both sequences are separately fed to the model, obtaining two sequences of attention weights λ_A and λ_B, pulled closer together by the alignment loss.

Each example is represented by a variable-length sequence of T feature vectors $\mathcal{X} = (x_1, x_2, \ldots, x_T)$. During training, each example is associated with a label y indicating whether an anomalous event appears in the sequence at least once. Hence, given a training set of examples $\{(\mathcal{X}_j, y_j)\}_{j=1}^{N}$ constructed as described above, we seek to train a neural network $f(\cdot;\theta)$ that minimizes the empirical error:

$$\min_{\theta} \quad \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{cl}(f(\mathcal{X}_i;\theta), y_i), \tag{3.1}$$

where $\mathcal{L}_{cl}(\cdot,\cdot)$ denotes the binary cross entropy (BCE) loss. For the architectural design of $f(\cdot;\theta)$, we took inspiration from [189]. It consists of two main parts, discussed in the following paragraphs: namely, the computation of attention coefficients and the creation of an aggregate video-level representation.

Attention coefficients

The aim of this module is to assign a weight $\lambda_t \in [0, 1]$ to each element of the input sequence. These weights will identify the most salient segments of the video *i.e.*, the likelihood of having observed an abnormal event within each segment. The module initially performs a masked temporal 1D convolution [136] to allow each feature vector to encode information from the past. Such a transformation, which does not alter the number of input feature vectors, is followed by two fully connected layers activated by ReLU functions, except for the last layer where a sigmoid function is employed.

Video-level representation

Once we have the attention values, we exploit them to aggregate the input feature vectors. Such an operation, which resembles a temporal weighted average pooling presented in [204], produces a single feature vector with unchanged dimensionality; formally:

$$\mathbf{x} = \sum_{t=1}^{T} \lambda_t x_t. \tag{3.2}$$

We finally feed $\mathbf{x}$ to a classifier $g(\cdot)$, composed of two fully connected layers. The final output represents the guess of the network for the value of y.

3.3.2 Training objective

As mentioned before, we train our network in a weakly supervised fashion, *i.e.*, only video-level labels are provided to the learner. However, to provide a stronger training signal and to encourage attention coefficients to highlight salient events, we follow recent work [27, 296] and use additional regularization terms that encode prior knowledge about the dynamics of anomalous events.

Often, anomalous activities are characterized by sparsity and smoothness. Namely, anomalies appear rarely (*i.e.*, normal events dominate) and transitions between the two modalities usually occur across multiple frames. This prior can be enforced on the scores learned by the model: most of them should be close to zero and vary smoothly across neighboring video segments. In formal terms, the first constraint is injected by penalizing the l_1 norm [34] of the attention weights, as follows:

$$\mathcal{L}_{sp} = \|\lambda\|_1, \tag{3.3}$$

while the second one can be carried out by imposing adjacent coefficients to vary as little as possible:

$$\mathcal{L}_{sm} = \sum_{t=1}^{T-1} (\lambda_t - \lambda_{t+1})^2. \tag{3.4}$$

Alignment loss

Our main contribution consists in adding a consistency-based regularization term to the overall objective function. Overall, the idea is to generate two slightly different

sequences out of a single one and, then, to encourage the model to produce the same attention coefficients for the two inputs.

To do so, we introduce a data augmentation function, shown in Fig. 3.1, that allows us to forge different versions $\mathcal{X}_A$ and $\mathcal{X}_B$ from the same example $\mathcal{X}$. In more detail, we split each sequence $(x_1, x_2, \ldots, x_T)$ into fixed-size blocks, whose length L is a hyperparameter we always set to 3; afterward, we randomly choose a feature vector within each block.

Once the variants $\mathcal{X}_A$ and $\mathcal{X}_B$ have been created, we ask the network to minimize the following objective function:

$$\mathcal{L}_a = \sum_{t=1}^{T} \left(\lambda_t^A - \lambda_t^B\right)^2, \tag{3.5}$$

where λ^A and λ^B are respectively the attention coefficients computed by the network for $\mathcal{X}_A$ and $\mathcal{X}_B$. With this additional regularization term, we seek to enforce that not only adjacent time-steps should have the same weight, but also those lying within a wider temporal horizon.

Overall objective

Finally, the objective function will be:

$$\mathcal{L} = \mathcal{L}_{cl} + \alpha\mathcal{L}_{sp} + \beta\mathcal{L}_{sm} + \gamma\mathcal{L}_a, \tag{3.6}$$

where the parameters α, β, and γ weight the regularization terms, and $\mathcal{L}_{cl}$ is the BCE classification loss.

3.3.3 Temporal proposal

During inference, we refine the anomaly scores with a post-processing step. Often, two segments considered important by the network are separated by "holes", mostly due to noisy acquisitions or weak representations. The purpose of this phase is to merge temporally close detections into a single candidate. In particular, as in [315], we first remove from the candidate set all time steps whose attention scores fall below a threshold. The remaining non-zero scores are then used to generate the temporal proposal.

	Video Level		Segment Level		Frame Level Proposal		Frame Level	
Align Loss	AUC %	AP %	AUC %	AP %	AUC %	AP %	AUC %	AP %
-	**97.91**	**98.36**	84.39	66.75	85.14	68.01	84.57	65.96
✓	97.79	98.28	**85.49**	**66.87**	**90.23**	**71.68**	**85.65**	**66.05**

Table 3.1: Results of our model with and without the proposed alignment loss across different evaluation levels and metrics. Leveraging the proposed term improves almost all metrics.

To generate the proposals, we do not use the raw coefficients, but a refined version of them. In particular, we compute a 1D activation map in the temporal domain, called the Temporal Class Activation Map (T-CAM) [315], which indicates the relevance of segment t for predicting one of the two classes involved (*normal* vs. *anomalous*). Each value a_t of this activation map is computed as $a_t = g(x_t)$, *i.e.*, the prediction of the classifier $g(\cdot)$ (introduced in Sec. 3.3.1) when the contributions of all time steps except the t-th one are masked. Furthermore, we extract the weighted T-CAM, which combines the attention weight and the T-CAM activation value, *i.e.*, $\psi_t = \lambda_t \cdot a_t$. This operation lets us emphasize the most important features for generating the proposal.

The last operation involves interpolating the weighted scores in the temporal axis and taking the bounding box that covers the largest connected component [189] to generate the final proposal [315]. The anomaly score for each proposal is:

$$s(t_{start}, t_{end}) = \frac{1}{t_{end} - t_{start} + 1} \sum_{t=t_{start}}^{t_{end}} \psi_t, \tag{3.7}$$

where $s(t_{start}, t_{end})$ denotes the anomaly score assigned to the proposal $[t_{start}, t_{end}]$.

3.4 Experiments

Following a weakly supervised setting, we use only video-level annotations during training. During evaluation, we exploit the segment-level and frame-level ground truth provided by XD-Violence [279] to assess the model's ability to localize anomalous events with fine temporal granularity. We evaluate anomaly detection performance using AUC and AP, following standard practice.

Supervision	Method	AP %
Unsupervised	SVM baseline	50.78
	OCSVM [227]	27.25
	Hasan et al. [93]	30.77
Weakly Supervised	Sultani et al. [248]	75.68
	Wu et al. [279]	75.41
	RTFM [253]	77.81
	Ours	71.68

Table 3.2: We report the frame-level AP score on XD-Violence for both unsupervised and weakly-supervised methods. All the competitors exploit the I3D network for extracting features from RGB frames.

Training details

We use the Adam optimizer [120] with a learning rate of 10^{-4} for the first 10 epochs and 10^{-5} for the remaining 40 epochs. The hyperparameters for the loss components are set to $\alpha = 2 \times 10^{-8}$, $\beta = 0.002$, and $\gamma = 0.5$. The threshold for discarding low weighted T-CAM scores is set to 0.35, and the batch size is 8.

Results

Tab. 3.1 compares the baseline approach with and without the proposed alignment objective. Adding the objective yields a remarkable improvement in most metrics. In particular, we observe a gain of about 1% in AUC for the segment-level and frame-level metrics, while AP remains almost unchanged. The largest improvement occurs for the temporal proposal metric, where we gain 5 points in AUC and around 4 points in AP. The video-level metrics remain approximately unchanged, but still very high.

When comparing our approach with other recent works (see Tab. 3.2), it can be seen that it outperforms the unsupervised state-of-the-art methods; however, it is in turn surpassed by the weakly supervised ones. We conjecture that such a gap is mainly due to the bag representations inherent in these approaches, which could confer superior robustness; therefore, we leave to future works the extension of our idea to these methods.

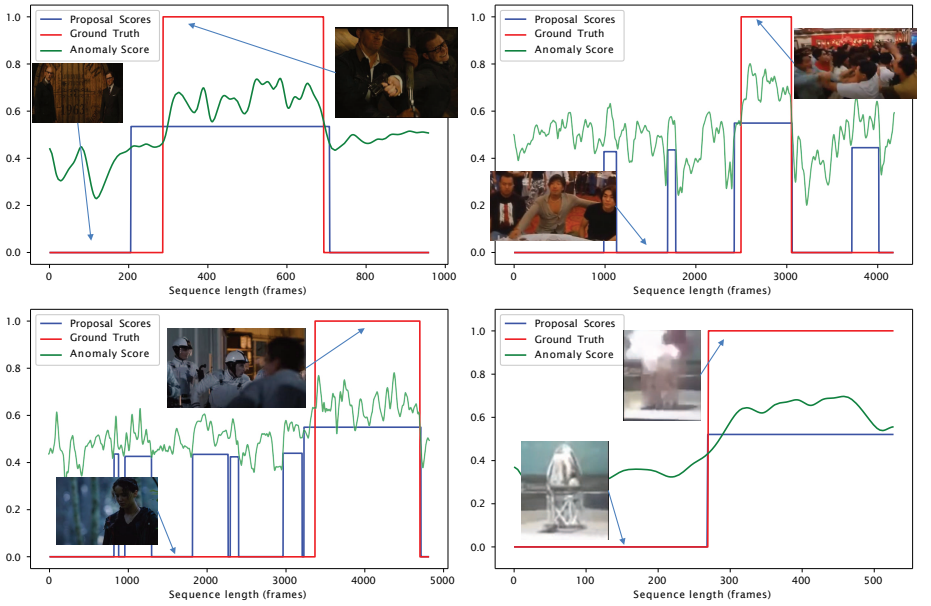


Figure 3.2: Qualitative examples of the capabilities of our model to perform anomaly localization. The temporal proposal scores are indicated with a blue line, while the weighted T-CAM scores and the ground truth are shown in green and red respectively.

Qualitative analysis

Fig. 3.2 presents several qualitative results, showing two fight scenes in the first row and a riot and an explosion in the second. We observe that the scores increase when an anomalous action begins. Unfortunately, they remain close to the uncertainty regime (around a score of 0.5) and never move toward clear-cut decisions.

By inspecting the original videos, we can also explain why the scores produced by the model are noisy and subject to local fluctuations. Sudden scene changes or camera movements are often mistaken for real anomalies by the model, which is sensitive to these visual discontinuities because segment-level annotations are unavailable during training. By contrast, when the entire video sequence is considered, the model can recognize them correctly.

3.5 Conclusions

This chapter studied weakly supervised video anomaly localization through the lens of *temporal consistency*. By introducing a consistency-based self-supervised regularization term, the proposed approach exploits temporal coherence as an implicit supervisory signal, complementing sparsity and smoothness priors commonly used in MIL-based formulations.

While not designed to outperform specialized architectures, the results show that enforcing consistency across temporally related views improves frame-level localization and proposal quality. This confirms temporal consistency as a principled mechanism for stabilizing learning under weak supervision and extracting fine-grained temporal information from coarse labels.

At the same time, this formulation operates at the sequence level, assigning anomaly scores to time steps without explicitly modeling the entities that generate the observed dynamics. This limitation motivates a transition toward *object-centric temporal learning*. Multi-object tracking addresses this complementary problem by enforcing temporal coherence at the level of persistent object identities and trajectories, rather than frame-level regularity. The following chapter builds on this shift by introducing probabilistic tracking models that explicitly represent object persistence and temporal dynamics.

4

TrackFlow: Multi-Object Tracking with Normalizing Flows

4.1 Multi-object tracking as conditional density estimation

Multi-object tracking (MOT) provides an object-centric perspective on temporal learning. Unlike anomaly detection, which assigns scores to time steps by modeling deviations from regularity, tracking aims to maintain persistent identities over time, despite occlusions, missed detections, and clutter. This makes temporal coherence explicit at the level of trajectories: a valid track is one whose motion and appearance remain consistent across the sequence. This problem is central in applications ranging from autonomous driving to visual surveillance that require reliable tracking under strong ambiguity and crowding.

Modern trackers can be broadly categorized into tracking-by-detection, tracking-

Publication. Gianluca Mancusi, **Aniello Panariello**, *et al. TrackFlow: Multi-Object Tracking with Normalizing Flows*. ICCV, 2023 [174].

Candidate contribution. Idea, methodology, implementation, experiments, and writing.

by-regression, and attention-based end-to-end approaches [19, 12, 117, 74, 100, 162, 181, 286, 277]. In this chapter we focus on the tracking-by-detection setting, in which detections are produced per frame and the challenge is data association across time.

Although tracking-by-regression and attention-based end-to-end approaches have recently attracted significant interest, *tracking-by-detection* remains competitive thanks to its simplicity, reliability, and the availability of accurate object detectors [83]. Motivated by these considerations, we strengthen *tracking-by-detection* algorithms by enriching the information they typically leverage, *i.e.*, the displacement between estimated and actual bounding boxes [275, 312], with additional cues. Indeed, as shown by several works on multi-modal tracking [46, 310], visual appearance is only one possible source of information. Skeletal pose [51], depth maps [212, 58], and even thermal measurements [132] can improve robustness because they encode a deeper understanding of the scene. In particular, since humans move and interact in a three-dimensional space, one goal of this work is to provide the tracker with the predicted distance from the camera, thus resembling what is commonly referred to as "2.5D". To achieve this, we train a per-object distance regressor on MOTSynth [72], a synthetic dataset displaying immense variety in scenes, lighting/weather conditions, pedestrians' appearance, and behaviors.

However, the fusion of multi-modal representations raises a key question: how should the contribution of each input modality be weighted in the overall cost? This is a crucial design choice, as it directly affects the subsequent assignment problem. Existing works often resort to hand-crafted formulas and heuristics; for example, DeepSORT [275] computes two different cost matrices and combines them through a weighted sum. Notably, the authors of [212] build on a probabilistic formulation, which recasts the cost $c_{i,j}$ as the likelihood of the event "the i-th detection belongs to the j-th tracklet". The task then becomes estimating a density function over correct associations, termed *inliers*. We investigate whether data-association costs can be learned as a conditional likelihood model, replacing hand-crafted fusion rules with a single probabilistic score that adapts to track history and scene context. Although these fusion strategies may appear reasonable, they hide several practical and conceptual pitfalls:

- They introduce additional hyperparameters, which require careful tuning on a separate validation set and hence additional labeled data.
- A single choice of these hyperparameters cannot fit different scenes perfectly, as

these typically display different dynamics in terms of pedestrians' motion and spatial density, the camera's position/motion, and lighting/weather conditions. Therefore, the right trade-off is likely to be scenario-dependent;
- Common approaches (*e.g.*, a simple weighted summation) assume the input modalities to be independent, thus overlooking their interactions.

We address these weaknesses through a dedicated parametric density estimator, termed TrackFlow, which summarizes several input costs/displacements into a single score, *e.g.*, the probability that a specific detection D belongs to a particular track T. To approximate the underlying conditional probability distribution $\mathcal{P}(D \in T | \, T)$ over the input costs, we draw on deep generative models, in particular normalizing flows [62, 63, 122]. These models provide a flexible and effective tool for density estimation. Importantly, the module also relies on an additional context-level representation that informs the model about scene-level characteristics. In this way, likelihood computation is conditioned on cues that capture scenario-specific properties.

Extensive experiments on MOTSynth [72], MOT17 [183], and MOT20 [57] show that the naive cost metric, *i.e.*, the 2D intersection between predicted and candidate bounding boxes, can be replaced by the score provided by our approach, with a remarkable performance gain.

This chapter focuses on the formulation of multi-object tracking as a probabilistic density estimation problem, in which data association is learned rather than handcrafted. The objective is not to propose a universal tracking system, but to study how flexible generative models can capture temporal dynamics and uncertainty in crowded scenes, providing a principled alternative to heuristic association costs.

4.2 Related work

Most modern multi-object tracking methods follow the tracking-by-detection paradigm, combining object detectors with hand-crafted motion and appearance models for data association [275, 18, 312, 318, 169]. Recent works have explored learning data association directly, either through end-to-end architectures or probabilistic formulations that model uncertainty explicitly [14, 181, 301, 31]. These methods highlight the importance of temporal modeling beyond deterministic matching.

Normalizing flows have emerged as a powerful class of generative models for density estimation in high-dimensional spaces [63, 218], and have recently been applied to

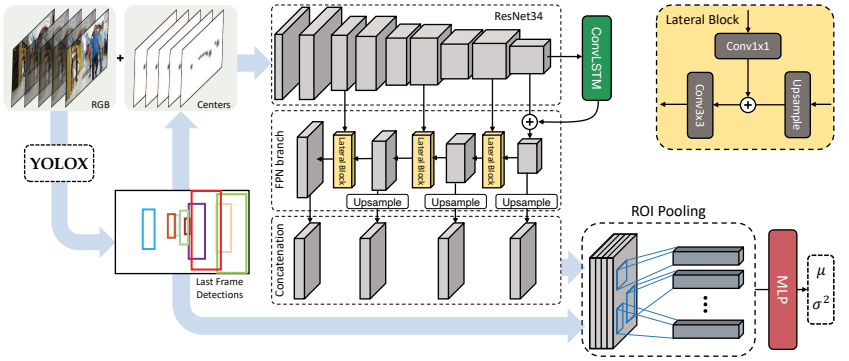


Figure 4.1: Overview of the camera distance estimator. DistSynth predicts per-object distances from a short video clip and serves as the distance estimation module within TrackFlow. We further provide the centers of each bounding box as an additional input channel. After several convolutional blocks processing each frame independently: (i) a temporal module is devised to extract temporal patterns; (ii) the activation maps undergo the FPN branch, in order to preserve local details. Finally, feature maps from distinct layers are stacked and passed to the RoI pooling layer. The latter produces per-pedestrian vector representations, which we finally use to predict the pedestrians' expected distance μ and uncertainty σ^2.

motion modeling and trajectory prediction [184, 116]. This chapter builds on these ideas by using normalizing flows to model track dynamics and association likelihoods in multi-object tracking.

4.2.1 Problem formulation

Let $\mathcal{T}_{t-1}$ denote the state of a track up to time $t-1$, and let d_t be a detection at time t. Multi-object tracking can be formulated as estimating the conditional likelihood

$$p(d_t \mid \mathcal{T}_{t-1}), \tag{4.1}$$

which measures the compatibility between a detection and an existing track. Learning this distribution enables data association to be performed by likelihood maximization rather than heuristic cost design. In the remainder of the chapter, this likelihood is instantiated as $\mathcal{P}_\theta(D \in T \mid T)$, parameterized by a normalizing flow over association cues.

4.3 Method

Our architecture comprises two main building blocks:

- A deep neural regressor that, given a monocular image, estimates the distance of each pedestrian from the camera (see Sec. 4.3.1). We called it **DistSynth**, as we train only on synthetic images from MOTSynth [72].
- A deep density estimator, termed **TrackFlow** (Sec. 4.3.2), which has to merge 2D cues (*e.g.*, the spatial displacement between bounding boxes) with the 3D localization information obtained through DistSynth.

4.3.1 DistSynth: estimating per-instance distance from a monocular image

As the output of the distance estimator is meant to further refine the association cost between detections and tracks, it is crucial to handle temporary occlusions and noisy motion patterns. Therefore, as discussed below, our model integrates temporal information (*e.g.*, a short collection of past frames) with visual cues to achieve smoother and more reliable distance predictions.

In detail, the network is fed with a short video clip $\mathbb{R}^{T \times C \times W \times H}$, where T is the clip length, C is the number of channels, and W and H are the frame width and height. Since we are not interested in a dense prediction for the entire scene but in pedestrian-level predictions, we ask the network to focus on a restricted set of locations, namely those around the bounding boxes provided by an off-the-shelf detector such as YOLOX [83]. To do so, we concatenate an additional channel to the RGB frames representing the center of each bounding box.

The architecture mainly follows the design of residual networks [95] (in our experiments, we used ResNet-34 pretrained on ImageNet [59]). Importantly, we apply two modifications to the feature extractor to enhance its capabilities, discussed in the following two sub-paragraphs.

Exploiting temporal information. While related works [319, 94] focus solely on the last frame of interest, we propose to condition the predictions of camera distances on a small window of previous frames, thus encompassing temporal dynamics. The main goal is to provide a much more robust prediction when the target pedestrian is partially or temporarily occluded in the current frame but visible in the previous ones. In that case, his/her history would compensate and smooth the prediction. Therefore,

we equip the backbone with a layer capable of processing the sequence of past feature maps: precisely, a Convolutional Recurrent Neural Network, *i.e.*, a ConvLSTM [237], whose output is a single-frame feature map encoding all the history of past frames. We insert such a module in the deeper layer of the network *i.e.*, to the exit of the last residual block of our backbone.

Improving spatial representations. Standard CNNs usually exploit pooling layers to progressively downsample the activation maps. For classification, there is little doubt that this operation provides advantageous properties (*e.g.*, translation invariance, high-level reasoning, *etc.*). For per-object distance estimation, however, we argue that a strong reliance on pooling can be detrimental. In particular, for people far away from the camera (*i.e.*, those enclosed by tiny bounding boxes), pooling layers may over-subsample the corresponding spatial regions, leading to a significant loss of visual cues.

To avoid this issue, we equip the feature extractor with an additional branch based on a Feature Pyramid Network (FPN) [155]. In practice, it begins with the encoding produced by the temporal module, then proceeds in the reverse direction (*i.e.*, from deeper layers to those closer to the input), and restores the original resolution through up-sampling layers and residual paths from the forward flow. Fig. 4.1 provides an overview of the architecture.

Output and loss function. Once the feature maps have been processed through the temporal module and the pyramid, we again exploit the bounding boxes and perform RoI pooling [85] to obtain a feature vector for each pedestrian. The result is a $\mathbb{R}^{N \times H \times K \times K}$ feature map, where N indicates the number of detected pedestrians, H the number of hidden channels, and $K = 4$ is the dimension of the RoI pooling window. We process these feature maps through a multilayer perceptron (MLP), which outputs the predicted distances. Finally, rather than producing a point estimate, we ask the network to place a Gaussian distribution over the expected distance, thus obtaining the model's aleatoric uncertainty [17, 60]. In practice, this means yielding two values, $d \equiv d_\mu$ and d_{σ^2}, and optimizing the Gaussian Negative Log Likelihood (GNLL) [190], as follows:

$$\text{GNLL}(d_{true} | d, d_{\sigma^2}) = \frac{1}{2} \left(\log(d_{\sigma^2}) + \frac{(d - d_{true})^2}{d_{\sigma^2}} \right).$$

4.3.2 TrackFlow: modeling the density of correct associations through normalizing flows

In a nutshell, the *tracking-by-detection* paradigm usually relies on the Kalman filter [244, 18] to estimate the next 2D spatial position $\mathbf{p}_j^{t+1} = [x_j^{t+1}, y_j^{t+1}]$ of a certain pedestrian j in the next $t+1$-th frame. The prediction $\hat{\mathbf{p}}_j^{t+1}$ depends upon the set of previous observations, contained in a short track $T_j = [\mathbf{p}_j^t, \mathbf{p}_j^{t-1}, \dots, \mathbf{p}_j^{t-|T|+1}]$ recording the past matched locations of the pedestrian j. Afterward, given a new set of detections $D_i = [\mathbf{p}_i, \mathbf{w}_i, \mathbf{h}_i] \;\; i = 1, 2, \dots, |D|$ (with $\mathbf{w}_i$ and $\mathbf{h}_i$ being the width and the height of the bounding box respectively), the *cost* of a candidate association between D_i and the track T_j can be computed as the displacement $\Delta_p \equiv \Delta_p(T_j, D_i)$ between the predicted location and the candidate one, *i.e.*, $\Delta_p = \mathrm{d}(\hat{\mathbf{p}}_j^{t+1}, \mathbf{p}_i)$. In such a notation, $\mathrm{d}(\cdot, \cdot)$ stands for any function penalizing such a displacement, as the Euclidean distance $||\hat{\mathbf{p}}_j^{t+1} - \mathbf{p}_i||_2^2$. Similarly, the variation of the sizes of the bounding box, *i.e.*, $\Delta_{w,h} \equiv \Delta_{w,h}(T_j, D_i)$ could be taken into account.

Furthermore, thanks to the regressor introduced in Sec. 4.3.1, we can also exploit discrepancies in camera distance, $\Delta_d = \mathrm{d}(\hat{d}_j^{\,t+1}, d_i)$, where $\hat{d}_j^{\,t+1}$ is the estimated one-step-ahead distance for track T_j and d_i is the distance of candidate detection D_i inferred through DistSynth. To ease the notation, from now on we denote T_j and D_i simply as T and D.

Once these costs have been computed (though additional cues could also be considered), we define an aggregated cost function $\Phi(T, D) = f(\Delta_p, \Delta_{w,h}, \Delta_d)$ that jointly computes the cost of the candidate association $D \in T$. Among the possible approaches, we build on the probabilistic formulation proposed in [212] and define the cost Φ as the negative log-conditional likelihood:

$$\begin{aligned}\Phi(T, D) &= -\log \mathcal{P}_\theta(D \in T \mid T) \\ &= -\log f([\Delta_p, \Delta_{w,h}, \Delta_d] \mid T, \theta).\end{aligned}$$

In that formulation, the target conditional probability distribution $\mathcal{P}_\theta(\cdot)$ is parameterized as a learnable function $f(\cdot \mid \theta)$, whose parameters θ are optimized by maximizing the likelihood of correct associations (often referred to as *inliers*). To simplify optimization, the authors of [212] factorized the density above, assuming that each marginal

distribution is independent, such that $\mathcal{P}_\theta(D \in T \mid T) \propto \mathcal{P}_p \, \mathcal{P}_{w,h} \, \mathcal{P}_d$. Therefore:

$$\Phi(T, D) = -\log \mathcal{P}_{\theta_1}(\Delta_p) - \log \mathcal{P}_{\theta_2}(\Delta_{w,h}) - \log \mathcal{P}_{\theta_3}(\Delta_d).$$

As discussed in the next subsection, we do not impose such an assumption but approximate, via Maximum Likelihood Estimation (MLE), the joint conditional distribution with a deep generative model $f(\cdot \mid T, \theta)$.

Overview of the architecture

Among many possible choices (*e.g.*, variational autoencoders [121], generative adversarial networks [91] or the most recent diffusion models [102]), we borrow the design of $f(\cdot|T, \theta)$ from the family of normalizing flow models [62, 218, 63]. Notably, they provide an exact estimate of the likelihood of a sample, in contrast with other approaches that yield a lower bound (as the variational autoencoder and its variants [259, 254]). Moreover, normalizing flow models grant high flexibility, as they do not rely on a specific approximating family for the posterior distribution. The latter is instead a peculiar trait of the variational methodology, which may suffer if the approximating family does not contain the true posterior distribution.

Briefly, a normalizing flow model creates an invertible mapping between a simple factorized base distribution with known density (*e.g.*, a standard Gaussian in our experiments) and an arbitrary, complex and multi-modal distribution, which in our formulation is the conditional distribution $\mathcal{P}(D \in T \mid T)$ underlying correct associations. The mapping between the two distributions is carried out through a sequence of L invertible and differentiable transformations $g_l(\cdot \mid T)$ (with parameters θ_l, omitted in the following), which progressively refines the initial density through the rule for change of variables. In formal terms, our proposal named **TrackFlow** takes the following abstract form:

$$f([\Delta_p, \Delta_{w,h}, \Delta_d] \mid T, \theta) = g_L^{-1} \circ \cdots \circ g_2^{-1} \circ g_1^{-1}, \tag{4.2}$$

where

$$\begin{aligned} \text{forward pass}: \quad & \mathbf{z}_l = g_l(\mathbf{z}_{l-1} \mid T); \ \mathbf{z}_L \sim \mathcal{P}_\theta(D \in T \mid T) \\ \text{inverse pass}: \quad & \mathbf{z}_{l-1} = g_l^{-1}(\mathbf{z}_l \mid T); \ \mathbf{z}_0 \sim \mathcal{N}(0, 1) \end{aligned}$$

are the forward pass (*i.e.*, used when sampling) and the inverse pass (*i.e.*, used to evaluate densities) of TrackFlow. The model can be learned via Stochastic Gradient Descent (SGD), by minimizing the negative log-likelihood on a batch of associations sampled from the true $\mathcal{P}(D \in T \mid T)$ (*i.e.*, corresponding to valid tracks). The loss function exploits the inverse pass and takes into account the likelihood under the base distribution [125] plus an additive term for each change of variable occurred through the flow.

Base architecture. Regarding the design of each layer $g_l(\cdot \mid T)$, we make use of several well-established building blocks, such as normalization layers, masked autoregressive layers [198], and invertible residual blocks [40]. In particular, our model features a cascade of residual flows [9], which we preferred to other valuable alternatives (*e.g.*, RealNVP [63]) in light of their expressiveness and proven numerical stability. For the sake of conciseness we are omitting the inverse functions, but the overall representation of the forward pass of the l-th block proceeds as follows:

$$\begin{aligned} \text{residual block}: \quad & z = \mathrm{MLP}_l(\mathbf{z}_{l-1}) + \mathbf{z}_{l-1}, \\ \text{act. norm}: \quad & z = s_l \odot z + b_l, \\ \text{masked auto. flow}: \quad & \mathbf{z}_l = \mathrm{MAF}_l(\mathrm{concat}[z \mid\mid \mathrm{e}_l]), \end{aligned}$$

where e_l refers to an auxiliary learnable representation discussed below, by which we take into account the dependence on the external context (*e.g.*, the track T).

Context encoder

Dependence on temporal cues. As stated in Eq. 4.2, the inverse pass (but also the forward one) of TrackFlow depends also on the observed track T. By introducing such a conditioning information, the model could learn to assign higher likelihood to the candidate associations that exhibit motion patterns coherent with those observed in the recent past. To introduce such an information, we take inspiration from [274, 228] and condition each invertible layer on an additional latent representation e_l, given by a temporal encoder network e_{θ_l} s.t. $e_l = \mathrm{e}_{\theta_l}(T)$ fed with the observed track T.

Importantly: (i) as advocated in several recent works [229, 8], we provide the encoder network with relative displacements between subsequent observations rather than absolute coordinates; (ii) regarding the design of the encoder network, it can be any module that extracts temporal features (*e.g.*, Gated Recurrent Units (GRU) [47, 228]

or transformers [260, 184]). In this work, the context encoder follows a subset of the Temporal Fusion Transformer (TFT) [151], a well-established and flexible backbone for time-series analysis and forecasting. In particular, we start from the original architecture and discard the decoding modules, retaining only the layers needed to encode the previous time steps (referred to as "*past inputs*" in the original paper) of track T.

Dependence on scene-related visual information. Importantly, one of the main issues we aim to address is the lack of adaptation to the scene under consideration. Existing approaches typically use the same aggregated cost function for all conditions, which we argue may clash with the variability expected in real-world settings. Since different scenarios may display substantial differences (*e.g.*, night/day, camera orientation, moving/stationary camera, *etc.*), some costs should be weighted differently.

To provide this signal, we further condition the estimated density $f(\cdot \mid T, \theta)$ on a visual representation of the whole current frame. In particular, we exploit the variety of MOTSynth [72] (comprising more than five hundred scenarios) and encode each frame x^t through CLIP's visual encoder [210], thus leveraging its well-known zero-shot capabilities. On top of the extracted representations, we run the k-means algorithm and split them into $|C| = 16$ clusters, each of which represents an abstract hyper-scenario. We then introduce the cluster index $\hat{\mathbf{c}}$ as an additional conditioning variable:

$$f \equiv f([\Delta_p, \Delta_{w,h}, \Delta_d] \mid T,\ \hat{\mathbf{c}}\ , \theta) \tag{4.3}$$

$$\text{where } \hat{\mathbf{c}} = \text{argmin}_{i=1,\dots,|C|}\ ||\,\text{CLIP}_v(x^t) - c_i||_2^2, \tag{4.4}$$

and c_i are the $|C|$ centroids retrieved through the k-means pass. Such a formulation also allows inference on novel scenarios, unseen during the training stage. To practically condition the model, we simply extend the context encoder $\text{e}_{\theta_l}(\cdot)$ to take an additional learnable embedding $\text{emb}_l[\hat{c}]$ as input, s.t. $\text{e}_l \equiv \text{e}_{\theta_l}(T, \text{emb}_l[\hat{c}])$. In practice, in light of the TFT [151] layout employed by our context encoder, it becomes natural to include scene embeddings $\text{emb}_l[\hat{c}]$ as static covariates [151, 273] – *i.e.*, something holding time-independent information about the time-series. We kindly refer the reader to the original paper [151] for all the important details regarding how the TFT uses covariates to influence the forward pass of each layer.

Normalization of the cost matrix

Once the density estimator $f(\cdot \mid T,\ \hat{\mathbf{c}}\ , \theta)$ has been trained, we exploit its output to fill the cost matrix $\Phi(D_j, T_i)$. Following Bastani *et al.* [7], we apply a further normalization step, defined as follows:

$$\Phi^{\text{row}} = \frac{e^{\Phi(D_j,T_i)/\sigma}}{\sum_k e^{\Phi(D_j,T_k)/\sigma}}, \qquad \Phi^{\text{col}} = \frac{e^{\Phi(D_j,T_i)/\sigma}}{\sum_k e^{\Phi(D_k,T_i)/\sigma}}, \tag{4.5}$$

$$\hat{\Phi}(D_j, T_i) = \min(\Phi^{\text{row}}(D_j, T_i), \Phi^{\text{col}}(D_j, T_i)). \tag{4.6}$$

In practice, we compute softmax (smoothed through a temperature hyperparameter σ) along rows and columns of Φ; afterward, we take the cell-wise minimum between the two cost matrices. We finally pass the normalized cost matrix $\hat{\Phi}$ to the Hungarian algorithm for solving the associations.

4.4 Experiments

Experimental protocol. Datasets and evaluation metrics used in this chapter are introduced in Chapter 2. Here we report only TrackFlow-specific protocol choices (data cleaning, splits, and evaluation settings). Specifically, we evaluate on MOTSynth, MOT17, and MOT20.

4.4.1 Evaluation metrics

We adopt the standard tracking and distance estimation metrics discussed in Chapter 2. In addition, we introduce a dedicated metric, ALOE, to evaluate distance estimation robustness under occlusions.

Tracking. We report IDF1 and HOTA following standard MOT evaluation practice.

Distance estimation. We report standard distance metrics (*i.e.*, τ-Accuracy, $\text{ALP}_{@\tau}$, Abs. Rel., Sq. Rel., RMSE, $\text{RMSE}_{\log}$) and introduce **ALOE** (Average Localization of Occluded objects Error), which measures average absolute distance error restricted to objects whose occlusion level falls within a specified range.

ALOE: Average Localization of Occluded objects Error (ours). Let d_i be the predicted distance and d_i^{GT} the ground-truth distance for object i, and let $o_i \in [0, 1]$

	Easy		Moderate		Hard		*All*	
Metrics	HOTA ↑	IDF1 ↑	HOTA ↑	IDF1 ↑	HOTA ↑	IDF1 ↑	HOTA ↑	IDF1 ↑
SORT [18]	63.48	79.40	50.31	62.11	37.48	45.13	48.42	59.05
+ TrackFlow GT	**+4.37**	**+7.41**	**+5.33**	**+9.09**	**+6.54**	**+10.88**	**+5.49**	**+9.62**
+ TrackFlow	**+0.31**	**+0.97**	**+0.81**	**+1.63**	**+0.74**	**+1.56**	**+0.54**	**+1.22**
ByteTrack [312]	63.22	80.84	49.91	62.46	37.61	46.15	48.21	59.79
+ TrackFlow GT	**+3.76**	**+2.82**	**+5.47**	**+5.51**	**+5.08**	**+4.60**	**+4.75**	**+4.54**
+ TrackFlow	**+0.13**	**+1.80**	**+0.47**	**+1.21**	**+0.88**	**+1.81**	**+0.49**	**+1.41**
OC-SORT [30]	65.56	81.61	52.42	63.50	38.10	45.48	49.96	60.16
+ TrackFlow GT	**+2.41**	**+3.76**	**+4.88**	**+7.70**	**+6.18**	**+9.55**	**+4.67**	**+7.67**
+ TrackFlow	**+0.44**	**+0.84**	**+0.60**	**+1.09**	**+1.17**	**+1.96**	**+0.31**	**+0.70**

Table 4.1: Tracking results on MOTSynth. For each tracker, we report its extended version using either our distance estimator (*i.e.*, TrackFlow) and ground-truth distances (*i.e.*, TrackFlow GT). For a wider comparison, we also report two tracking-by-regression approaches.

	MOT17		**MOT20**	
Metrics	HOTA ↑	IDF1 ↑	HOTA ↑	IDF1 ↑
SORT [18]	64.17	72.98	60.56	74.30
+ TrackFlow	**+1.78**	**+1.41**	**+0.15**	**+0.22**
ByteTrack [312]	67.73	79.81	58.94	74.89
+ TrackFlow	**+0.40**	**+0.23**	**+0.54**	**+0.06**
OC-SORT [30]	66.22	77.74	55.18	71.22
+ TrackFlow	**+0.35**	**+1.12**	**+0.53**	**+0.76**

Table 4.2: Tracking results on the validation set of MOT17 and the train set of MOT20 [57].

denote its occlusion level. For an occlusion bin $[\tau_1, \tau_2] \subseteq [0, 1]$, we define:

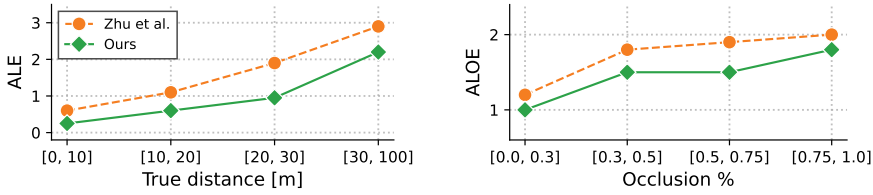
$$\text{ALOE}_{[\tau_1:\tau_2]} = \frac{1}{|\mathcal{I}_{\tau_1:\tau_2}|} \sum_{i \in \mathcal{I}_{\tau_1:\tau_2}} |d_i - d_i^{\text{GT}}|, \qquad \mathcal{I}_{\tau_1:\tau_2} := \{i \mid o_i \in [\tau_1, \tau_2]\}. \tag{4.7}$$

This metric isolates failure cases that are underrepresented by aggregate errors, highlighting whether temporal modeling stabilizes distance predictions during partial and severe occlusions.

Implementation details. We feed the distance estimator with video clips of 6 frames, sampled with a uniform stride of length 8; this way, each clip lasts approximately 2 seconds. We adopt 1280×720 as input resolution, thus further preserving the visual cues. We set the batch size to 4 and use Adam [120] as the optimizer, with a learning

			ALP ↑			ALOE ↓		
Method	$\delta_{<1.25}$ ↑	RMSE ↓	@0.5m	@1m	@2m	[0.3:0.5]	[0.5:0.75]	[0.75:1.]
SVR	26.7%	12.5	3.4%	6.8%	13.8%	-	-	-
DisNet [94]	27.5%	12.1	3.8%	7.5%	14.6%	-	-	-
Zhu et al. [319]	94.7%	2.15	34.5%	56.2%	78.5%	1.78	1.95	2.03
DistSynth	99.1%	1.91	48.0%	68.9%	86.1%	1.39	1.41	1.78

Table 4.3: Comparison of various distance estimators on MOTSynth [72]. Our DistSynth exhibits superior performance across all the metrics reported. We highlight the enhancements observed in terms of ALOE, confirming an improved ability to withstand occlusions.



(a) (b)

Figure 4.2: The ALE metric and the proposed ALOE metric are evaluated on MOTSynth. Specifically, (a) our approach reduces ALE within the reported distance range shown in the plot, and (b) our method displays increased stability during occlusion events, resulting in superior performance, which can be attributed to our temporal approach.

rate of 5×10^{-5}. The density estimator, on the other hand, is trained with a batch size of 512 and Adam [120] with learning rate 1×10^{-3}. The normalizing flow consists of 16 flow blocks, each comprising 64 hidden neurons. For context conditioning, we fix the number of observed past observations to $|T|= 8$ and the number of visual clusters to $C = 16$. Unless otherwise specified, both networks are trained only on synthetic data (*i.e.*, the training set of MOTSynth); we leave the question of transfer learning strategies to future work.

4.4.2 Impact on tracking-by-detection

In this section, we empirically show that our proposed method, applied to popular state-of-the-art *tracking-by-detection* techniques, improves upon the MOTSynth and the MOTChallenge benchmarks (see Tabs. 4.1 and 4.2).

On MOTSynth, we focus on three trackers (*i.e.*, SORT [18], Bytetrack [312], and OC-SORT [30]) and adhere to the following common evaluation pipeline: (i) we compute

predicted bounding boxes through YOLOX [83]; (ii) as our approach requires an estimate of per-pedestrian camera distances, we exploit YOLOX bounding boxes by providing them to the distance estimator DistSynth (Sec. 4.3.1); (iii) we finally integrate our density estimator TrackFlow into the pipeline of each tracker, applying the normalization described in Sec. 4.3.2 before the Hungarian algorithm.

Additionally, we report an upper bound termed **TrackFlow GT** (*i.e.*, *ground-truth*). Like standard TrackFlow, it relies on YOLOX detections to compute 2D displacements, but it leverages ground-truth distances (available in MOTSynth) instead of DistSynth predictions. This allows us to assess the potential of TrackFlow under near-perfect distance estimates. From a methodological standpoint, this comparison isolates the contribution of the distance signal from the rest of the association pipeline, since the detection backbone and the motion cues are left unchanged. The gap between TrackFlow and TrackFlow GT can therefore be interpreted as an estimate of the room for improvement that could be unlocked by more accurate monocular distance prediction.

We provide the results of this comparison in Tab. 4.1. Our results indicate that TrackFlow enhances the performance of the considered trackers on all MOTSynth splits, *i.e.*, easy, moderate, and hard, highlighting the benefits of our method across three levels of complexity. The improvements are substantial when ground-truth distances are employed as expected; nevertheless, a consistent gain is also observed when leveraging estimated distances, leading to improved identity accuracy reflected by a steady enhancement of the IDF1 metric.

As reported in Tab. 4.2, the evaluation on the MOT17 and MOT20 benchmarks further shows that TrackFlow consistently improves the considered trackers in even more realistic scenarios (notably, SORT benefited the most from our approach). While evaluating on the MOT20 benchmark, we rely on the same YOLOX [83] model employed for MOT17. This particular YOLOX model was trained on two distinct datasets, namely CrowdHuman [235] and the initial half of MOT17, which aligns with the training methodology adopted in ByteTrack [312].

As mentioned above, both TrackFlow and DistSynth have been trained solely on synthetic data without additional fine-tuning, yet still achieve competitive results on real data. This suggests that the geometric cues learned by our framework transfer reasonably well across the synthetic-to-real domain gap, despite the evident differences in appearance and scene statistics. This finding opens the door to future research on how components such as camera distance can be used to advance multi-object tracking.

4.4.3 Distance estimation: comparison with the state of the art

To assess the merits of the proposed distance estimator, we compare it with baselines and valid competitors from the current literature. We report the results of such a comparison in Tab. 4.3.

Comparison with Support Vector Regressor (SVR). SVR consists of a simple shallow baseline based on a support vector regressor, which exploits the dimensions of the bounding boxes (*i.e.*, height and width). Through the comparison with such a naive approach, we would like to emphasize the gap w.r.t. the bias present in the task at hand *i.e.*, the smaller the bounding box, the farther the pedestrian from the camera. As expected, the SVR approach yields low performance w.r.t. our method, due to its inability to generalize to objects with different aspect ratios.

Comparison with DisNet. DisNet [94] consists of an MLP of 3 hidden layers, each of 100 hidden units with SeLU [124] activations. The network is fed with the relative width, height, and diagonal of bounding boxes, computed w.r.t. the image dimension; these features are then concatenated with three corresponding reference values (set to 175 cm, 55 cm, and 30 cm). As can be seen, the improvements of DisNet are marginal w.r.t. SVR, but its results are substantially lower than those obtained by both Zhu *et al.* and our approach.

Comparison with Zhu *et al.*. The model proposed by Zhu *et al.* [319] shares some similarities with our approach, as it relies on ResNet as feature extractor and RoI pooling to build pedestrian-level representations. However, thanks to the additional modules our model reckons on (*i.e.*, the temporal module and the FPN branch), it is outperformed by our approach under all the considered metrics. Our advancements concerning ALE and ALOE, compared to Zhu *et al.*, are illustrated in Fig. 4.2.

4.4.4 Analysis of TrackFlow

We next examine the advantages of conditioning our density estimator on scene information. To do so, we focus on a single tracker (*i.e.*, SORT) and compare how its tracking performance changes if the context encoder of TrackFlow (see Sec. 4.3.2) considers only time-dependent information about the tracks, thus discarding the scene-related visual information provided through cluster centroids c_i. From the results reported in Tab. 4.4 (second and third rows), it can be observed that visual conditioning (*i.e.*, the row marked with ✓) leads to a lower negative log-likelihood on both the MOTSynth

		MOTSynth	MOT17		
	cond.	NLL ↓	NLL ↓	HOTA ↑	IDF1 ↑
SORT [18]	-	-	-	64.17	72.98
TrackFlow	✗	-1.48	-5.66	65.34	74.77
TrackFlow	✓	-1.80	-5.81	65.95	75.71
$\text{TrackFlow}_{\text{FT}}$	✗	-0.10	-7.29	65.94	75.97
$\text{TrackFlow}_{\text{FT}}$	✓	-0.12	-7.50	65.70	76.22

Table 4.4: For MOT17, ablative study w/o scenario-level conditioning (cond.) and w/o fine-tuning ($\text{TrackFlow}_{\text{FT}}$). Performance is reported in terms of negative log-likelihood (NLL) and HOTA/IDF1 for the evaluation of the resulting tracker.

and MOT17 validation sets, as well as better HOTA and IDF1 results on MOT17. We interpret these findings as confirmation of the benefits of designing a cost function that is aware of the scene.

Finally, we note that only synthetic data were used to train our models. One might ask whether additional fine-tuning on real-world data could help. To shed light on this question, we select the best-performing model obtained on MOTSynth and carry out a final fine-tuning stage on the MOT17 training set for a further 20 epochs with a lower learning rate. We report the performance of the resulting model (*i.e.*, $\text{TrackFlow}_{\text{FT}}$) without visual conditioning. Two major findings emerge from the last two rows of Tab. 4.4: (i) as in the frozen-model setting, the introduction of visual cues leads to better results (with the only exception of HOTA on MOT17); (ii) in general, additional training steps can profitably adapt TrackFlow to real-world scenarios, as confirmed by both the lower negative log-likelihood attained after fine-tuning (-7.50, compared with -5.81 before fine-tuning) and the higher tracking results.

4.5 Conclusions

This chapter framed multi-object tracking as a problem of probabilistic temporal learning, in which the core operation is data association conditioned on track history and contextual cues. Instead of relying on hand-crafted fusion rules, TrackFlow learns a single likelihood-based association score via normalizing flows, integrating standard 2D motion cues with monocular distance estimates produced by DistSynth.

Experiments on MOTSynth, MOT17, and MOT20 demonstrate that replacing heur-

istic association costs with a learned conditional likelihood consistently improves identity preservation and overall tracking quality. Beyond aggregate performance gains, the proposed ALOE metric reveals that explicit temporal modeling stabilizes distance estimates under occlusions, exposing failure modes that are often masked by conventional error measures.

More broadly, the results of this chapter support a central theme of this part: temporal coherence constitutes a powerful learning signal, and probabilistic modeling provides a principled mechanism for converting temporal structure into robust decision rules. By explicitly modeling identity persistence, motion regularity, and association likelihoods over time, TrackFlow shows how temporal structure can be exploited to resolve ambiguity at the level of observations, even in the presence of noise and partial visibility.

More broadly, these findings suggest that temporal coherence alone is not the only useful signal in video understanding. Once identity persistence is modeled reliably, geometric information becomes essential for resolving ambiguity under occlusion and viewpoint change.

The next chapter builds on this shift from temporal persistence to structured geometry by studying monocular per-object distance estimation. Rather than associating objects across frames, it focuses on learning object-centric representations that support robust geometric reasoning in dynamic scenes.

5

Monocular Per-Object Distance Estimation via Object-Centric Masked Modeling

5.1 Object-centric geometry as a complement to temporal learning

The previous chapter addressed temporal learning in video through object persistence: given a stream of detections, tracking aims to maintain coherent identities over time by exploiting motion, appearance, and temporal continuity. However, object-centric reasoning in video is not only about *who* is present and *where* they move, but also about *how the scene is structured* geometrically. In crowded scenes and under occlusions, geometric cues such as distance-to-camera help disambiguate interactions, enforce physically plausible associations, and stabilize predictions when appearance and motion

Publication. Aniello Panariello, *et al. Monocular Per-Object Distance Estimation with Masked Object Modeling*. CVIU, 2025 [196].

Candidate contribution. Idea, methodology, implementation, experiments, and writing.

become unreliable.

In this chapter, we study *monocular per-object distance estimation*, *i.e.*, predicting a metric distance value for each object instance given a single RGB frame (and its bounding box). Although related to dense depth estimation, per-object distance estimation is a targeted alternative that focuses computation on objects of interest and remains meaningful even under partial visibility. This makes it particularly relevant for downstream video understanding pipelines, in which object-level signals must be robust to occlusions, viewpoint changes, and domain shifts.

Beyond distance prediction itself, the chapter is motivated by a broader question that will recur throughout the dissertation: *how can we learn object-centric representations that are stable and reusable under distribution shifts, without relying on expensive retraining?* To this end, we introduce a masked learning objective tailored to multi-object settings, designed to regularize object representations by forcing models to reconstruct informative object regions from partial observations.

Recently, Masked Image Modeling (MiM) has become a dominant strategy for representation learning, in which models are trained to reconstruct missing content from partially observed inputs (*e.g.*, MAEs [97]). While effective for learning global representations, standard masking strategies are not naturally aligned with multi-object downstream tasks: masking is applied on the full image grid, and reconstruction tends to be dominated by frequent background patterns. As a result, the learned features may under-emphasize fine-grained object cues that are critical for instance-level predictions such as distance.

To address this limitation, we propose **Masked Object Modeling (MoM)**, a masking strategy skewed toward *instance-level* representation learning. Instead of masking raw image patches, we defer masking until after region-based extraction and apply it independently to each object representation. The decoder is trained to reconstruct only the object region, and the reconstruction objective is jointly optimized with the supervised distance loss using a shared backbone. This design (i) focuses learning on object-relevant details, (ii) acts as a strong regularizer under occlusions, and (iii) yields representations that transfer more reliably across domains.

We incorporate MoM into a hybrid architecture termed **DistFormer**, combining convolutional feature pyramids for high-resolution feature extraction with transformer blocks for object-level reasoning. DistFormer predicts a distribution over distances, enabling uncertainty-aware estimates in ambiguous conditions. We validate the approach on KITTI [84], NuScenes [26], and MOTSynth [72], showing consistent gains

in accuracy and robustness, including synthetic-to-real transfer.

Conceptually, this chapter completes the perspective of Part I by complementing sequence-level temporal regularity (anomaly detection) and identity-level temporal coherence (tracking) with an *object-centric geometric signal*. Importantly, it also anticipates later parts of the dissertation: the instance-level masked objective shows how structured self-supervision can produce reusable components, a theme that will reappear when knowledge is preserved across tasks (Part II) and composed across models through parameter-space operations (Part III).

5.2 Related work

Estimating object distances from a single RGB image (*i.e.*, monocular distance estimation) is a crucial task for many computer vision applications [3, 86, 214, 87, 213, 138, 319, 94]. A first family of methods performs per-object distance estimation from geometry alone, without exploiting visual features. Among them, the Support Vector Regressor (SVR) [89] finds the best-fitting hyperplane from bounding-box geometry, while Inverse Perspective Mapping (IPM) [173, 217] converts image points to bird's-eye-view coordinates. However, IPM introduces image distortion, making distance prediction harder for far objects or curved roads.

Successive methods [94, 89] use deep networks on bounding-box geometry, improving upon purely algorithmic techniques but remaining limited across object classes: a car and a person with similar boxes can be at very different distances. Zhu *et al.* [319] addressed this issue with a Faster R-CNN-inspired architecture [216] that extracts visual features through *RoIPool* [85], while DistSynth [174] further leveraged multiple frames to improve temporal consistency.

A different line of work is monocular per-pixel depth estimation [71, 87, 138, 213, 147], which predicts a depth map from a single RGB image. Representative approaches use cascaded networks [71] or multi-resolution depth maps [138]. However, these methods are computationally expensive and difficult to deploy in real-time systems such as autonomous driving, and converting depth maps into object distances is nontrivial in the presence of occlusions and loose bounding boxes. Our approach instead is lighter and can predict distance for partially occluded objects.

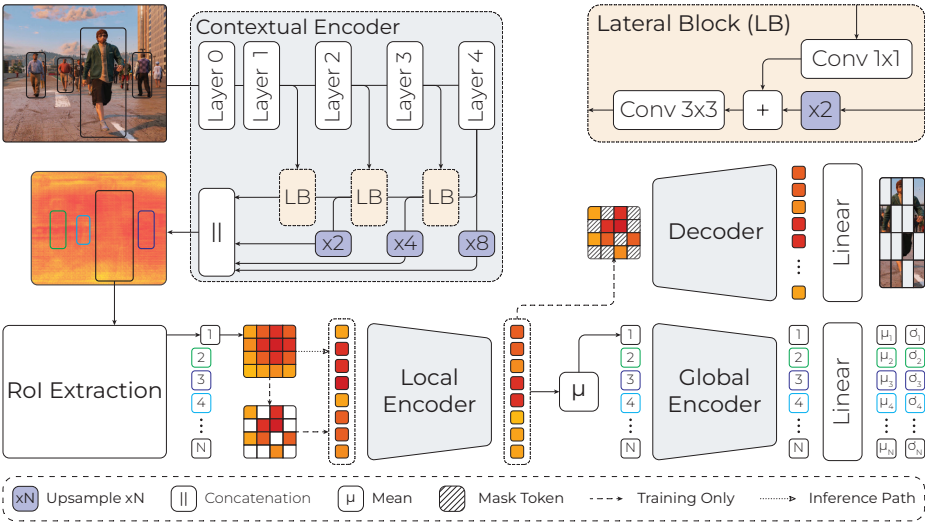


Figure 5.1: Overview of DistFormer. The image passes through the Contextual Encoder and RoI extraction to obtain per-object representations, which are then processed by the Local and Global Encoders. Finally, we predict a Gaussian distribution modeling distance and uncertainty.

5.3 Method

DistFormer, shown in Fig. 5.1, includes three main modules: the Contextual Encoder, the Local Encoder, and the Global Encoder. First, the **Contextual Encoder** $f(\mathbf{x};\theta_f)$ (Sec. 5.3) produces a feature map from an image $\mathbf{x}$, encoding visual features. This network is a CNN equipped with a Feature Pyramid Network [155], which extracts high-level features while retaining fine-grained details.

Secondly, given the bounding box for each instance, we extract per-object representations with standard region-based pooling to obtain a structured grid of activations for each target object, denoted as **latent patches** (or tokens). Then, we apply our masking strategy to these latent patches, treating each instance independently. Unmasked tokens are fed to the **Local Encoder** $LE(f(\cdot);\theta_L)$ (Sec. 5.3), which further enhances local visual reasoning and promotes the extraction of localized fine-grained details. Specifically, it performs self-attention between latent patches of the same object, disregarding other objects. Notably, the Local Encoder interacts with the **Decoder** network and, based on that, receives a self-supervised training signal (**MoM**, Sec. 5.3.1).

Unlike standard MAEs, our approach jointly optimizes the pretext and downstream tasks (*i.e.*, multi-target distance estimation) in a single training stage. In this context,

there is a third and final component, the **Global Encoder** (Sec. 5.3), trained to predict object distances. Since our task benefits from modeling mutual distances, the Global Encoder applies self-attention to representations from distinct objects.

This multi-object analysis also plays an essential role in human perception, as stated by the *adjacency principle* [88]. Indeed, an object's apparent size or position in the field of view is determined by the size and distance cues provided by adjacent objects.

Contextual Encoder. We feed our backbone f with an RGB frame $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$ where C is the number of channels and (H, W) are the frame resolution. We adopt a CNN as our backbone, specifically we use ConvNeXt pre-trained on ImageNet-22k [95, 165]. While convolutional networks offer benefits such as translation invariance and hierarchical reasoning [130], the pooling layers and other resolution reduction techniques can hinder distance estimation, as distant objects may be represented by only a few pixels.

To avoid such shortcomings, we employ Feature Pyramid Networks (FPN) [155], similar to previous works [174, 135, 39, 289]. FPN-based networks consist of a forward branch for downsampling and a backward branch that progressively upscales the output. The backward branch utilizes Lateral Blocks (LB) to upscale the feature maps from the forward pass, concatenated into a single feature map.

Local Encoder. Next, the goal is to extract fixed-size latent representations, one for each object. To do so, we start from the feature maps processed by the Contextual Encoder and then apply the *RoIAlign* [96] operation[1], which extracts the portions of the feature map covered by the target objects. Denoting by N the number of bounding boxes, this operation yields feature vectors $\mathcal{F}_{i|i\in\{1,\dots,N\}} \in \mathbb{R}^{c \times h \times w}$, where c is the number of channels of the feature map and (h, w) are the dimensions of the RoI quantization (8×8 in our experiments).

To better encode information about the target object, we employ a module termed Local Encoder (LE), which consists of the final 6 layers of a pretrained ViT-B/16 model [64]. To feed it, we rearrange the feature map of each object $\mathcal{F}_i$ into a vector – *i.e.*, $\mathbb{R}^{c \times (h \times w)} \rightarrow \mathbb{R}^{c \times (h \cdot w)}$ – treating each cell of the activation map as a token. The LE then performs self-attention on the object's tokens. This operation aims to encode informative intra-object features and to encourage the model to focus on the most critical portions of the object, *e.g.*, the non-occluded ones.

[1] Compared to *RoIPool*, commonly used in this task [319, 146], *RoIAlign* avoids misalignments thanks to a more accurate interpolation strategy.

Figure 5.2: Corresponding reconstructions from the Decoder trained with **+ MoM** .

Masking	Time	FLOPs	RMSE
DistFormer			
+ all tokens	67.6ms	380 G	2.87
+ mask 30%	66.5ms	360 G	2.89
+ **mask 50%**	65.2ms	345 G	2.91
+ mask 80%	64.4ms	330 G	3.14

Table 5.1: Computational cost analysis on KITTI (all classes).

Global Encoder. Since the Local Encoder is based on ViT layers, it outputs $h \cdot w$ tokens for each bounding box, which we aggregate along the token axis through global average pooling – $\mathbb{R}^{c \times (h \cdot w)} \rightarrow \mathbb{R}^{c \times 1}$ – to obtain a single representation. This representation is fed to the Global Encoder (GE), structured as a two-layer ViT architecture. Its function is to improve the modeling of inter-object relationships within the scene. Similarly to the Local Encoder, the Global Encoder employs multiple layers of attention-based operations. However, it performs self-attention between tokens corresponding to different objects. This operation enables each token to integrate information from other objects, including partially occluded ones. Consequently, each object representation in $\mathbb{R}^c$ is passed to a Multi-Layer Perceptron (MLP) to predict its distance.

5.3.1 Masked object modeling (MoM)

We devised a self-supervised learning approach called **Masked Object Modeling (MoM)** in our architecture. During training, only 50% of the input tokens are fed into the Local Encoder. Subsequently, we employ a two-layer Decoder network $D(\cdot, \theta_D)$ to reconstruct only the input image area covered by the bounding box. As in original MAEs [97], the unmasked tokens are taken directly from the encoder output, while a learned control token substitutes the absent masked tokens. Finally, the Masked Object Modeling (MoM) objective is:

$$\mathcal{L}_{\text{MoM}} = \mathbb{E}_{(\mathbf{x}_i, \mathcal{F}_i) \in \mathcal{X}} \left[||D(LE(\mathcal{F}_i, \theta_{LE}), \theta_D) - \mathbf{x}_i||_2^2 \right], \tag{5.1}$$

where $\mathbf{x}_i$ is the i-th object image portion and $\mathcal{X}$ is the whole set of target objects.

Overall objective. Given the intrinsic uncertainty of the task, we opt to predict a Gaussian distribution over the expected distance instead of providing a point estimate. The mean of the distribution represents the distance, while its variance is the model's

aleatoric uncertainty [17, 60], which refers to the inherent noise contained in the observations. To do so, the MLP mentioned in Sec. 5.3 outputs two scalars for each object; the supervised part of the training signal is then obtained by minimizing the Gaussian Negative Log Likelihood (GNLL) [190] of the ground-truth distances. The final objective combines Eq. 5.1 with the GNLL:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{MoM}} + \mathcal{L}_{\text{GNLL}}, \tag{5.2}$$

where α is a hyperparameter balancing the importance of the MoM objective.

MoM acts as an object-level feature regularizer. Adopting the MoM objective provides several noteworthy advantages, as discussed below. In Sec. 5.4.3, we demonstrate its efficacy in enhancing zero-shot capabilities, synthetic-to-real transfer, and robustness to noisy bounding boxes. In this respect, we argue that our masking strategy encourages the Local Encoder to focus on **more stable cues**. To support this claim, we report examples of reconstructions in Fig. 5.2: the decoder appears to exclude non-essential details (*e.g.*, colors) that are irrelevant to distance estimation. Therefore, by prioritizing task-related cues, our model gains resilience to unexpected and unimportant visual variations, which are particularly problematic under domain shift.

MoM allows flexible inputs. Applying MoM enables us to reduce the number of latent tokens provided to the Local Encoder. Significantly, this reduction can be applied not only during training but also at inference time. As reported in Tab. 5.1, leveraging masking at inference reduces both wall-clock time and memory footprint while still producing accurate distance estimates ($\rightarrow$ low root mean squared error).

5.3.2 Comparison with related works

In Sec. 5.2, we pointed out similarities with dense depth estimation, which focuses on predicting depth maps of images. For example, works such as [147, 138, 213, 292] commonly employ end-to-end transformer architectures. Nevertheless, there are several distinctions:

- **Memory.** We defer self-attention layers until the Region of Interest (RoI) stage, applying self-attention only to targets instead of processing the entire input patch set. This leads to a significant reduction in memory footprint: while methods such as [147] demand 8 V100 GPUs for training, our method requires only a single 2080 Ti GPU with the same batch size, aligning with sustainability constraints.

	MOTSynth				NuScenes			
Method	ABS↓	SQ↓	RMSE↓	$\delta_{<1.25}$ ↑	ABS↓	SQ↓	RMSE↓	$\delta_{<1.25}$ ↑
SVR	54.67%	6.758	12.61	26.08%	57.65%	10.48	19.18	32.49%
DisNet	8.73%	0.266	2.507	94.15%	18.47%	1.646	8.270	76.60%
Zhu *et al.*	4.40%	0.116	2.131	98.71%	14.95%	1.244	7.507	84.54%
DistSynth	3.71%	0.073	1.567	99.13%	-	-	-	-
Monoloco†	3.59%	0.064	1.488	99.69%	-	-	-	-
DistFormer (no MoM)	3.36%	0.046	1.152	99.31%	11.16%	0.807	6.363	91.10%
DistFormer (+**MoM**)	**2.81%**	**0.037**	**1.081**	**99.70%**	**8.13%**	**0.533**	**5.092**	**95.33%**

Table 5.2: Comparison on the NuScenes and MOTSynth datasets. (†) Uses GT poses.

- **Speed.** Moreover, in our approach, the number of tokens for self-attention depends on the number of objects in the frame, thus enhancing scalability and flexibility (as discussed in Sec. 5.4.2).
- **Adaptability.** Due to its decoupled design, which separates detection from distance estimation, our model can easily adapt to new detectors.
- **Flexibility.** Per-object distance estimation enables predicting distances for partially occluded objects, a critical task for tracking and autonomous driving.

Regarding the accuracy in estimating distances, we will report the results of a current state-of-the-art dense depth estimator such as Depth Anything V2 [292] in Tab. 5.3.

5.4 Experiments

5.4.1 Experimental setting

Evaluation Setting. We adhere to the widely adopted benchmark [319, 94, 112, 174, 146]. Namely, the model is supplied with ground truth bounding boxes (or poses) during inference, along with the input image, to disentangle the detector's performance from the distance estimator's.

Metrics. We rely on standard metrics for per-object distance estimation [71, 319, 80, 240, 158, 174], such as τ-**Accuracy** (δ_τ) [134] (*i.e.*, the maximum allowed relative error), the percentage of objects whose relative distance error falls below a threshold ($< k\%$) [146], and classical error metrics [319]: absolute relative error (**ABS**), square relative error (**SQ**), root mean squared error in linear space (**RMSE**), average localization error (**ALE**), and average localization of occluded objects error [174] (**ALOE**).

Method	ABS ↓	SQ ↓	RMSE ↓	$\delta_{<1.25}$ ↑
SVR	147.2%	90.14	24.25	37.90%
IPM	39.00%	274.7	78.87	60.30%
DisNet	25.30%	1.81	6.92	69.83%
Zhu *et al.*	54.10%	5.55	8.74	48.60%
+ classifier	25.10%	1.84	6.87	62.90%
DepthAnythingV2-B	27.37%	2.39	6.11	72.10%
DepthAnythingV2-L	27.22%	2.32	5.65	74.33%
DistFormer-RN	11.40%	0.39	3.42	91.98%
DistFormer (no MoM)	10.61%	0.34	3.17	93.43%
DistFormer (+MoM)	**10.39%**	**0.32**	**2.95**	**93.67%**

Table 5.3: Evaluation on KITTI, following the setting in [319].

Baselines. Our comparison includes **geometric methods**, *i.e.*, SVR [89], IPM [257], DisNet [94], and Monoloco [17], which exploits human pose to infer distance, as well as **feature-based methods**, *i.e.*, Zhu *et al.* [319], CenterNet [70], PatchNet [172], Jing *et al.* [112], and DistSynth [174].

Experimental details. We train every approach with ground-truth bounding boxes except Monoloco [17], which is trained with ground-truth human poses. For the experiments involving NuScenes and MOTSynth, we use the same ConvNeXt backbone for all methods, which operates on full-resolution images and extracts object features via RoIAlign with an 8×8 window. Tokens are randomly masked with a 50% ratio before the Local Encoder, which consists of the last six layers of a ViT-B/16 pretrained on ImageNet. The MoM decoder and the Global Encoder are two-layer transformer encoders with eight attention heads. Since the code bases of the other competitors are unavailable for KITTI, we also provide results with a ResNet-50 (the DistFormer-RN row) to ensure a fairer comparison. We train end-to-end on an NVIDIA 2080 Ti for 24 hours on NuScenes and MOTSynth and for 6 hours on KITTI, applying early stopping.

5.4.2 Distance estimation

Tabs. 5.2 and 5.3 present the results of our approach and previous work. Results on KITTI (Tab. 5.3) are taken from the respective papers (apart from DepthAnythingV2), while we implemented the other methods from scratch for NuScenes and MOTSynth (Tab. 5.2). We draw the following overall conclusions, which we expand below: (i)

	MOTSynth				KITTI			
	ALE ↓	ALOE ↓			ALE ↓	ALOE ↓		
Method	0m-100m	30-50	50-75	75-100	0m-100m	30-50	50-75	75-100
Zhu *et al.*	1.127	1.29	1.44	1.57	2.084	1.86	2.19	2.21
DistSynth	0.835	1.08	1.15	1.41	-	-	-	-
DistFormer	0.675	0.81	0.88	1.07	1.909	1.76	2.00	2.12
No Global Enc.	0.711	0.86	0.96	1.13	1.994	1.92	1.94	2.03
+ MoM	**0.617**	**0.76**	**0.85**	**0.99**	**1.854**	**1.71**	**1.89**	**1.94**

Table 5.4: ALE and ALOE comparison on KITTI and MOTSynth (using ConvNeXt).

DistFormer achieves competitive or state-of-the-art performance on the three datasets under consideration; (ii) notably, adding the MoM objective (**+ MoM**) further improves the accuracy of our approach with a consistent gain. In Sec. 5.4.4, we analyze this evidence through ablation studies to disentangle the contribution of the components involved in DistFormer.

NuScenes presents unique challenges due to its dynamic scenarios, complex traffic situations, and distances up to 150 meters. Despite these challenges, our proposed approach demonstrates robust performance, achieving state-of-the-art results across all metrics. The MOTSynth dataset, instead, focuses on the pedestrian class. However, its extensive range of landscapes and viewpoints renders it a comprehensive benchmark. In Tab. 5.2, our proposed method shows a remarkable -27% in RMSE w.r.t. Monoloco and -49% w.r.t. Zhu *et al.*.

Regarding KITTI (Tab. 5.3), we report the average results (**All**) on the three classes examined (*i.e.*, cars, pedestrians, cyclists). Our approach surpasses the state-of-the-art across all classes, except for the car class, which is on par. In this respect, we remark that the methods matching our performance leverage multiple input frames (*e.g.*, Jing *et al.* [112]), or they are designed to handle the class *car*. In contrast, our approach generalizes over all classes without further adjustments. We also tested Depth Anything V2 [292] on KITTI using the original pretrained weights for metric depth estimation. Our method outperforms it in object-level distance estimation, underscoring the difference between per-object and dense distance estimation tasks. Additionally, our model ($\approx 195M$ parameters) runs $6\times$ faster than the Base version ($\approx 97M$ parameters) and $20\times$ faster than the Large version ($\approx 335M$ parameters) on the same GPU.

Long-range evaluation. We further evaluate performance on pseudo long-range subsets of KITTI and NuScenes [146], focusing on objects beyond 40 meters in Tab. 5.5.

Method	Dataset (Long Range)	LiDAR	Lower is better ABS	SQ	RMSE	Higher is better $< 5\%$	$< 10\%$	$< 15\%$
DisNet	KITTI	-	10.6%	1.55	10.4	37.1%	65.0%	77.7%
Zhu *et al.*	KITTI	-	8.7%	0.88	7.7	39.4%	65.8%	80.2%
Zhu *et al.*	KITTI	✓	8.9%	0.97	8.1	41.1%	66.5%	78.0%
R4D	KITTI	✓	7.5%	0.68	6.8	46.3%	72.5%	83.9%
Ours	KITTI	-	**5.2%**	**0.22**	**3.3**	**56.3%**	**88.3%**	**97.3%**
DisNet	NuScenes	-	10.7%	1.46	10.5	29.5%	58.6%	75.0%
Zhu *et al.*	NuScenes	-	8.4%	0.91	8.6	40.3%	66.7%	80.3%
Zhu *et al.*	NuScenes	✓	9.2%	1.06	9.2	37.7%	63.5%	77.2%
R4D	NuScenes	✓	7.6%	0.75	7.7	44.2%	71.1%	84.6%
Ours	NuScenes	-	**7.3%**	**0.65**	**6.8**	**47.3%**	**75.4%**	**88.6%**

Table 5.5: Comparison on the Pseudo Long-Range KITTI and NuScenes datasets. Here $< k\%$ is accuracy below $k\%$ error.

Without additional tuning, DistFormer remains competitive and consistently outperforms methods requiring auxiliary sensors, highlighting the benefit of modeling global inter-object relations via self-attention.

5.4.3 The impact of masked object modeling

MoM enhances transfer learning. In Sec. 5.3.1, we conjectured that our masking strategy encourages the Local Encoder to prioritize the most consistent patterns (*e.g.*, shapes, but not appearance styles). This enables the model to suppress input variations that do not contribute valid information for estimating the distance of target objects. This reduced sensitivity to unimportant variations is advantageous in the case of domain shifts, as it enhances the robustness of the final distance predictor.

To investigate this aspect, we assess the model under domain shift, moving from a synthetic scenario (*i.e.*, MOTSynth) to real-world ones (*i.e.*, KITTI and NuScenes)[2]. Specifically: (i) we train two models on MOTSynth, one with the MoM objective and one without it; (ii) we then move to KITTI and NuScenes and compare the performance of the two models on the *pedestrian* class (*i.e.*, the only class present in all datasets). We evaluate under two settings: **zero-shot**, without any refinement on the target dataset, and **fine-tuning**, which allows a few training steps on a variable number of target examples.

[2]Notably, the intrinsic camera parameters reported by the authors of these datasets differ substantially.

	Zero-shot (no training)				Fine-tuning			
	ABS↓	SQ↓	RMSE↓	$\delta_{<1.25}$↑	ABS↓	SQ↓	RMSE↓	$\delta_{<1.25}$↑
Masking	MOTSynth → KITTI							
-	18.51%	0.56	2.95	70.44%	6.05%	0.12	1.89	97.58%
+ MoM	17.56%	0.47	2.87	83.57%	5.42%	0.12	1.48	99.16%
	MOTSynth → NuScenes							
-	20.74%	1.93	9.10	44.07%	15.62%	1.10	6.27	80.42%
+ MoM	19.94%	1.74	8.74	46.70%	10.28%	0.64	5.22	92.23%

Table 5.6: Masked Object Modeling impact in domain-shifts.

Tab. 5.6 reports the results of the two models (without and with **+ MoM**) under the evaluation protocol described above. Notably, there is an impressive gain from MOTSynth to KITTI in the zero-shot scenario (+13% in $\delta_{<1.25}$), showing that our masking strategy extracts features that are better aligned with real-world domains. Similarly, in the fine-tuning protocol, we observe a +12% gain in $\delta_{<1.25}$, proving that **MoM** provides a better starting point for training on new domains, and that keeping the objective (*i.e.*, object-level reconstruction) further improves the transfer capabilities of our approach.

Furthermore, in Fig. 5.3, we report RMSE and $\delta_{<1.25}$ for the fine-tuning experiment with varying numbers of adaptation samples. Specifically, we observe that the model with **+ MoM** reaches convergence much faster and is more stable than standard fine-tuning, showing that this strategy can also reduce the training-time requirements of the adaptation phase.

MoM aids in handling occlusions. **MoM** yields an advantage even for handling partially occluded objects. Specifically, we evaluate the accuracy at different occlusion levels by evaluating the ALOE [174] metric on MOTSynth and KITTI (Tab. 5.4). The proposed masking strategy provides a stable and reliable improvement over standard training, showcasing its efficacy. **MoM**'s efficacy in addressing occlusions can be ascribed to its distinctive approach to object representation during model training, resulting in more discernible and stable representations, enabling the model to differentiate between objects and background elements effectively.

MoM yields robustness to noisy bounding boxes. While employing ground truth bounding boxes is a common practice in this setting, one might question the model's

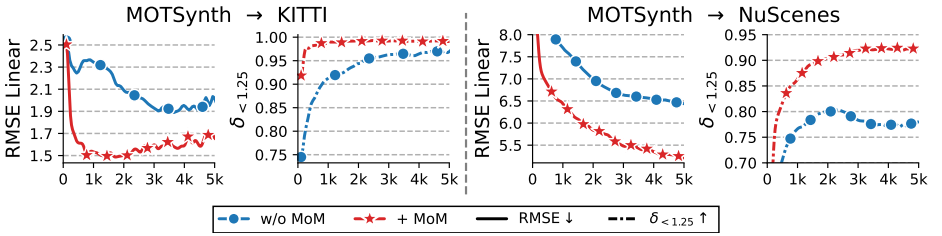


Figure 5.3: Resulting performance on the class *pedestrian* after fine-tuning with varying training set sizes.

performance in cases where bounding boxes are predicted by a detector, potentially influenced by errors. To this end, we employed YOLOX [83] as a state-of-the-art detector for MOTSynth, allowing us to gauge the system's resilience in real-world scenarios. Our findings show that incorporating MoM improves the system's performance, nearly reaching the upper bound in terms of the $\delta_{<1.25}$ while achieving a notable reduction in RMSE. Additionally, we purposely perturbed the geometry of ground truth bounding boxes, such that the noisy box and the original one have at least IoU equal to r. This experiment simulates real-world conditions where the exact bounding box might be imprecise, showing the benefits of **MoM**.

5.4.4 Ablation studies

We report an extensive ablation study (Tab. 5.7) of our model on the MOTSynth dataset.

Bounding box center prior. We additionally evaluate the effect of providing a coarse spatial prior by augmenting the input with a heatmap encoding bounding box centers. This signal yields a small but consistent improvement across metrics on MOTSynth, suggesting that simple object-centric spatial cues can complement RoI-based representations (Tab. 5.8).

Local Encoder and MoM. The experiments with the ResNet34-FPN and the ConvNeXt-S-FPN highlight the role of the LE to process local cues. Indeed, its application (✓) improves all metrics. Moreover, our masking (+ MoM) further improves results, confirming our claims regarding its regularizing effect.

Global Encoder. Discarding the Global Encoder worsens performance, confirming the discussion in Sec. 5.3. We also assess the merits of the ViT layers by comparing them with Graph Attention Networks (GAT) [261, 24]. Notably, GAT improves over

Contextual Enc.	Local Enc.	Global Enc.	ABS ↓	SQ ↓	RMSE ↓	$\delta_{<1.25}$ ↑
Ablative studies on the Contextual Encoder						
ViT-B/16	✓	ViT	7.88%	0.460	3.973	92.78%
	+ MoM	ViT	6.81%	0.316	3.473	94.90%
ResNet34	✓	ViT	4.14%	0.107	2.078	98.93%
	+ MoM	ViT	4.36%	0.094	1.826	98.94%
ResNet34-FPN	-	-	4.45%	0.102	1.975	98.91%
	✓	-	3.44%	0.056	1.363	99.53%
	-	ViT	3.30%	0.054	1.302	99.59%
	✓	ViT	3.15%	0.050	1.302	99.70%
	+ MoM	GAT	3.49%	0.049	1.213	99.51%
	+ MoM	ViT	3.00%	0.040	1.146	99.70%
Ablative studies on Local Encoder & MoM						
ConvNeXt-S-FPN	-	-	3.38%	0.055	1.289	99.31%
	✓	-	3.41%	0.055	1.275	99.34%
	-	ViT	3.38%	0.052	1.236	99.43%
	✓	ViT	3.36%	0.046	1.152	99.31%
	+ MoM	-	3.31%	0.053	1.290	99.38%
	+ MoM	GAT	3.26%	0.048	1.221	99.63%
ConvNeXt-S-FPN	**+ MoM**	ViT	**2.81%**	**0.037**	**1.081**	**99.70%**

Table 5.7: Ablation of the backbone and modules on MOTSynth.

the variant without a Global Encoder. However, ViT layers consistently outperform their GAT counterparts, underscoring their efficacy for multi-object analysis.

Contextual Encoder. Using ResNet leads to worse results (especially when removing the FPN layers), indicating the importance of stronger and larger feature maps. Notably, we observe a severe degradation when using ViT-B/16[3], showcasing the efficacy of convolutional networks in extracting valuable features for multi-object tasks.

[3]Due to its significant memory footprint at full resolution (*i.e.*, 720×1280), we resort to the standard resolution of 224×224.

Centers	ABS ↓	SQ ↓	RMSE ↓	$\delta_{<1.25}$ ↑
	3.07%	0.051	1.266	99.52%
✓	**2.81%**	**0.037**	**1.081**	**99.70%**

Table 5.8: Contribute of the centers mask on MOTSynth.

5.5 Conclusions

This chapter studied monocular per-object distance estimation as an object-centric learning problem and introduced **DistFormer**, a hybrid architecture that combines high-resolution convolutional features with transformer-based object reasoning. We proposed **Masked Object Modeling (MoM)**, an instance-level masked objective that reconstructs object regions from partially observed object tokens and is jointly optimized with the supervised distance loss. Across KITTI, NuScenes, and MOTSynth, MoM consistently improves accuracy and robustness, strengthening transfer under domain shifts, resilience under occlusions, and stability under noisy detections.

From a broader perspective, the contribution of this chapter goes beyond distance prediction. MoM acts as an object-level representation regularizer: by forcing reconstruction at the instance level, it biases learning toward stable cues and reduces reliance on incidental appearance details. This yields a reusable geometric signal that complements temporal cues and supports object-centric video understanding in challenging conditions.

Finally, this chapter provides a bridge to the rest of the dissertation. In Part I, we progressed from sequence-level temporal regularity to object-level persistence; here we showed that object-level learning can be further stabilized through structured self-supervision and geometry. The same principle, namely encoding knowledge in a form that remains reusable under change, will recur in Part II, in which knowledge must persist across evolving task distributions, and in Part III, in which such knowledge is explicitly composed and transported across models without full retraining.

Summary of Part I: Learning through temporal structure

Part I framed video understanding as learning from structured temporal information, progressively moving from sequence-level irregularity detection to object-centric temporal modeling. The first chapter introduced temporal consistency as a learning signal for weakly supervised anomaly localization, showing how coherent predictions across temporally related views can compensate for the lack of frame-level supervision. The second chapter shifted to multi-object tracking, modeling data association as conditional density estimation and leveraging probabilistic generative modeling to fuse heterogeneous cues and improve identity continuity over time. The third chapter further strengthened the object-centric perspective by introducing Masked Object Modeling for multi-target settings and a dedicated architecture for per-object distance estimation, highlighting how self-supervised reconstruction can regularize representations and inject geometric cues that benefit downstream temporal reasoning.

Together, these contributions establish a coherent progression in which temporal structure is exploited at increasing levels of granularity, from video-level dynamics to object-level and geometric reasoning. Across all three problems, a common theme emerges: temporal structure can act as an implicit supervisory signal, enabling learning even when annotations are weak, incomplete, or noisy. Consistency, masking, and

probabilistic modeling provide complementary mechanisms to stabilize representations over time, whether the underlying entities are events, objects, or metric quantities.

This perspective motivates the transition to the next part of the dissertation, in which evolution is no longer confined to the data stream but extends to the learning process itself. In the continual learning setting, models are exposed to sequences of tasks rather than sequences of observations, and the central challenge becomes how to adapt over time while preserving previously acquired knowledge in the presence of distributional shift and limited access to past data.

Part II

Learning through Tasks

6

Background on Continual Learning from Data Streams

This part of the dissertation focuses on learning systems that operate under continual exposure to data, with the training distribution evolving over time. Unlike the temporal learning problems addressed in Part I, in which time indexes observations within a sequence, continual learning treats time as an ordering over *tasks* or *data distributions*. In this setting, the challenge is not to model temporal dynamics within the input, but to adapt a model incrementally while preserving previously acquired knowledge.

This chapter introduces the continual learning perspective adopted in Part II, clarifies the assumptions under which the problem is studied, and reviews the technical background required to understand the contributions presented in the following chapters.

6.1 An informal overview

At an intuitive level, continual learning asks how a model can keep learning *new* tasks without undoing what it already knows. The difficulty is not that the tasks are individually hard, but that they arrive sequentially: after adapting to the present task,

the model is still expected to perform well on earlier ones, even though their data is no longer fully available.

The chapters in this part illustrate this challenge with two concrete examples. In CHARON (Chapter 7), a model first learns some classes of human actions from skeleton sequences and is later exposed to new actions. If it focuses only on the new task, it risks overwriting motion patterns that were useful for the old ones. Continual learning therefore requires deciding what to keep, what to compress, and how to replay prior information under a tight memory budget.

In CGIL (Chapter 8), the setting shifts to large pretrained vision–language models, in which the goal is to adapt to new visual classes while preserving the rich zero-shot capabilities of the original model. Rather than relearning everything from scratch, the method operates at the level of prompts and representations, showing that continual learning can also mean protecting a pretrained interface while adding new knowledge on top of it. These examples motivate the formal view below, in which the core issue is learning under sequential distribution shift without catastrophic forgetting.

6.2 Learning under distributional shift

Let $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_T$ denote a sequence of data distributions arriving over time. In continual learning, a model is trained sequentially on samples drawn from each $\mathcal{D}_t$, without assuming persistent access to data from previous distributions. The goal is to learn a parameterized function $f(\cdot;\theta)$ that performs well on all tasks encountered so far, despite being optimized in a strictly sequential manner.

A defining challenge of this setting is *catastrophic forgetting*: when optimizing the model on data from $\mathcal{D}_t$, performance on earlier distributions $\mathcal{D}_{<t}$ may degrade substantially. This phenomenon arises because gradient-based updates overwrite parameters that were previously important for older tasks.

In contrast to static learning, in which a single stationary distribution is assumed, continual learning must explicitly account for the temporal structure of data arrival and the resulting non-stationarity of the optimization process.

6.3 Problem formulations in continual learning

Continual learning problems are commonly formalized under different assumptions regarding task identity and supervision. In the *task-incremental* setting, the task index t

is known at inference time, and the model can exploit this information to disambiguate predictions. In contrast, the *class-incremental* setting introduces new classes over time while requiring the model to recognize all previously seen classes without access to task identifiers at test time. This formulation is widely regarded as the most challenging and realistic, as it prevents any explicit conditioning on task boundaries during inference.

More generally, continual learning can be viewed as learning from a non-stationary data stream in which samples arrive sequentially and distribution shifts correspond to the introduction of new concepts or classes. In the class-incremental case, the data stream is organized into a sequence of tasks $\{\mathcal{T}_i\}_{i=1}^{T}$, each associated with a disjoint label space, but the model is ultimately evaluated on the union of all classes encountered so far.

Class-Incremental Learning (Class-IL). Formally, in Class-IL a deep model $f(\cdot;\theta)$ parametrized by θ is presented with a sequence of tasks $\mathcal{T}_i$, with $i \in \{1,\dots,T\}$. The i-th task provides N_i labeled samples $\{(x_i^{(n)}, y_i^{(n)})\}_{n=1}^{N_i}$, where $y_i^{(n)} \in \mathcal{Y}_i$ and the label spaces are disjoint across tasks, *i.e.*, $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$. The learning objective corresponds to minimizing the empirical risk over all tasks:

$$\mathcal{L}_{\text{Class-IL}} = \sum_{i=1}^{T} \mathbb{E}_{(x,y)\sim\mathcal{T}_i} \left[\mathcal{L}(f(x;\theta), y)\right], \tag{6.1}$$

where $\mathcal{L}$ denotes the task loss (typically cross-entropy for classification). Since the model observes tasks sequentially and does not retain direct access to past data, naive optimization leads to catastrophic forgetting. As a result, tailored strategies are required to preserve previously acquired knowledge while learning new tasks.

In this part we will adopt a class-incremental perspective. The objective is to incrementally adapt representations over time while maintaining performance on previously learned classes, without relying on task identity or architectural expansion. In this context, the emphasis is placed on preserving representational knowledge across tasks, rather than enforcing explicit constraints on parameter updates or freezing subsets of the network.

6.4 Mitigating forgetting through replay

One of the most widely adopted strategies to mitigate catastrophic forgetting in class-incremental learning is *replay*. Replay-based methods approximate joint training by

reintroducing information from past tasks during optimization on new data. This is typically achieved either by storing a limited subset of previous samples in an episodic memory (*exemplar replay*) or by training a generative model to synthesize representative samples from earlier tasks (*generative replay*).

Under this perspective, learning at time step t is performed by optimizing an objective of the form

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_t}\big[\ell(f(x;\theta),y)\big] \;+\; \mathbb{E}_{(x,y)\sim\tilde{\mathcal{D}}_{<t}}\big[\ell(f(x;\theta),y)\big], \tag{6.2}$$

where $\mathcal{D}_t$ denotes the data distribution of the current task and $\tilde{\mathcal{D}}_{<t}$ is an approximation of past data distributions obtained through stored samples or generation. In rehearsal-based methods, this second term is often implemented through an additional regularization component $\mathcal{L}_{\mathcal{M}}$ computed on memory samples. Accordingly, the objective optimized at the current task $\mathcal{T}_c$ can be written as

$$\hat{\mathcal{L}}_{\text{Class-IL}} = \mathbb{E}_{(x,y)\sim\mathcal{T}_c}\left[\mathcal{L}(f(x;\theta),y)\right] + \mathcal{L}_{\mathcal{M}}. \tag{6.3}$$

In this formulation, parameter updates are not explicitly constrained. Forgetting is mitigated by reshaping the training signal to resemble joint optimization over all tasks observed so far, rather than by restricting the update space. This replay-based view is central to the methods explored in Part II, in which the focus lies on how prior knowledge can be efficiently retained and reused through data-driven mechanisms under strict memory and computational constraints.

6.5 Representation learning under continual adaptation

Beyond maintaining output performance, continual learning can be interpreted as a problem of learning stable and reusable representations. From this viewpoint, catastrophic forgetting is a symptom of representational drift, in which features learned for earlier tasks lose their semantic meaning as new tasks are introduced.

Several approaches therefore emphasize preserving representation quality over time, either by regularizing feature spaces, aligning representations across tasks, or replaying latent variables instead of raw inputs. This perspective becomes particularly relevant when dealing with large pre-trained models, for which re-learning representations

from scratch is computationally prohibitive.

In this dissertation, continual learning is studied as a process of *structured temporal progression in representation space*, in which adaptation is driven by data streams and controlled through replay mechanisms rather than explicit parameter isolation.

6.6 Continual learning in large-scale models

Recent advances in foundation models have raised new challenges for continual learning. Large vision and vision-language models exhibit strong generalization capabilities but are costly to retrain and sensitive to distributional shifts. Adapting such models incrementally requires methods that preserve zero-shot or pre-trained capabilities while enabling task-specific refinement.

The chapters in Part II address this setting by exploring replay and modular adaptation strategies that operate at the level of embeddings and latent representations, enabling continual learning without compromising the original capabilities of large pre-trained models.

6.7 Benchmarks and datasets

Standard benchmarks for Class-IL are typically derived from image classification datasets, such as CIFAR-100 [131] and ImageNet [59]. In these settings, a classification problem involving C classes is decomposed into T sequential tasks, where each task i contains C_i classes and the total number of classes satisfies $C = \sum_{i=1}^{T} C_i$. In the following, we summarize the datasets most frequently employed in the Continual Learning literature.

6.7.1 Natural domain datasets

Natural domain datasets are composed of images representing real-world objects and scenes. They are widely used in computer vision and serve as canonical benchmarks for evaluating Continual Learning methods due to their diversity and availability.

Seq. MNIST. MNIST [137] consists of 70 000 grayscale images of handwritten digits from 0 to 9, with 60 000 training samples and 10 000 test samples. Each image has resolution 28×28. Owing to its simplicity, Split MNIST is commonly adopted for preliminary and small-scale Continual Learning experiments.

Seq. SVHN. The Street View House Numbers dataset (SVHN) [188] contains approximately 600 000 color images of digits extracted from Google Street View, with resolution 32×32. Compared to MNIST, SVHN poses a more challenging recognition problem due to cluttered backgrounds and the presence of multiple digits. In Continual Learning, it is commonly partitioned into 5 binary tasks.

Seq. CIFAR-10 & Seq. CIFAR-100. These benchmarks are derived from CIFAR-10 and CIFAR-100 [131], respectively, and are among the most widely used datasets in Continual Learning. Split CIFAR-10 is typically divided into 5 binary tasks, while Split CIFAR-100 is commonly organized into 5, 10, or 20 tasks, providing a flexible trade-off between task granularity and computational cost.

Seq. ImageNet-R. ImageNet-R [101] includes 30 000 images spanning 200 ImageNet classes, depicting non-photorealistic renditions such as sketches, cartoons, and paintings. All images are disjoint from the original ImageNet training set. This dataset is commonly adopted to assess robustness to domain shifts when starting from ImageNet-pretrained models.

6.7.2 Fine-grained classification datasets

Fine-grained datasets consist of classes that are visually similar and often characterized by limited training samples, making the learning problem more challenging.

Seq. CUB-200. This benchmark is based on the CUB-200-2011 dataset [263], which contains 11 788 images across 200 bird species. It is typically split into 10 tasks of 20 classes each. Due to its difficulty, Continual Learning experiments often initialize models with ImageNet pretraining [38, 299], as several classes overlap semantically with ImageNet categories.

Seq. Cars-196. The Cars-196 dataset [129] includes 16 185 images of 196 car models, divided into 8144 training and 8041 test samples. In Continual Learning settings, it is commonly partitioned into 10 tasks, with 20 classes per task except for the final task, which contains 16 classes.

6.7.3 Satellite and medical image datasets

Although natural and fine-grained datasets dominate Continual Learning benchmarks, their similarity to common pretraining sources [210, 191], such as ImageNet, limits

their ability to capture real-world distribution shifts. For this reason, satellite and medical imaging datasets are increasingly used as out-of-distribution benchmarks.

Seq. EuroSAT. EuroSAT consists of 27 000 images acquired by the Sentinel-2 satellite [13], covering 10 land use and land cover classes. Each sample has resolution 64×64 and includes 13 spectral bands. In Continual Learning, experiments typically retain only the RGB channels and split the dataset into 5 binary tasks.

Seq. RESISC45. The NWPU-RESISC45 dataset [44] contains 31 500 RGB images of size 256×256 across 45 land use classes, collected from satellite and aerial imagery with varying spatial resolution. It is commonly divided into 9 tasks, each containing 5 classes.

Seq. ISIC. Based on ISIC 2018 [53], this dataset includes dermoscopic images of skin lesions. To mitigate class imbalance, the most frequent class is usually excluded, yielding 6 classes with 2648 training and 662 test samples. The dataset is then partitioned into 3 binary tasks.

Seq. ChestX. ChestX-ray8 [267] contains 108 948 chest radiographs originally annotated for multi-label classification. For single-label Continual Learning, it is commonly reduced to 6 non-overlapping disease classes and split into 2 tasks.

Seq. NTU-60 and Seq. NTU-120. This benchmark is built upon the NTU-RGB+D dataset [233] and was introduced in [21]. It targets sequential classification of 3D skeleton-based actions and is typically organized into 6 tasks of 10 classes each for the Seq. NTU-60 version. Later, NTU-RGB+D has been expanded into 120 classes [161] and used to define the Seq. NTU-120 benchmark [22], which comprises 12 tasks of 10 classes each.

Seq. Food-101N. Based on Food-101N [139], this dataset contains 310 009 web-collected images from 101 food categories, sharing the same classes as Food-101 [23] but with substantial instance-level label noise. In Continual Learning, it is commonly split into 5 tasks of 20 classes each.

6.8 Evaluation metrics

Model performance in Continual Learning is most commonly assessed using the *Final Average Accuracy* (FAA), which measures the average accuracy across all tasks after completing the final task. Let a_i^j denote the accuracy on task i evaluated after learning

task j. The FAA is defined as:

$$\text{FAA} \triangleq \frac{1}{T}\sum_{i=1}^{T} a_i^T . \tag{6.4}$$

While FAA reflects the final state of the model and aligns with the desiderata, it does not capture performance dynamics throughout training. To this end, several complementary metrics have been proposed:

- *Backward Transfer* (BWT) [167] quantifies how learning a task t influences performance on a previous task $t' < t$. Negative BWT corresponds to catastrophic forgetting.
- *Forward Transfer* (FWT) [167] measures how learning task t affects performance on a future task $t' > t$. Its computation assumes that predictions on unseen tasks are possible, which may not always hold.
- *Final Forgetting* (FF) [37] captures the average drop in performance for each task after training on all tasks:

$$\text{FF} \triangleq \frac{1}{T-1}\sum_{i=1}^{T-2} \max_{l\in\{0,\dots,T-2\}} \left(a_i^l - a_i^{T-1}\right) . \tag{6.5}$$

This metric reflects relative degradation but ignores absolute accuracy levels.

Throughout this dissertation, we primarily report FAA, as it remains the most widely adopted metric in the Continual Learning literature [215, 25, 245, 179, 10].

7

Mask and Compress: Efficient Skeleton-based Action Recognition in Continual Learning

7.1 Memory-efficient continual skeleton action recognition

Human Action Recognition (HAR) has become critical in various domains such as surveillance [156, 194], rehabilitative healthcare [293], and sports analysis [126, 234]. Early HAR approaches focused on RGB or grayscale videos due to their widespread availability. However, recent advances have explored alternative modalities, including skeletal joints [66, 141, 293], depth [226], point clouds [73], acceleration [133], and WiFi signals [250]. Among these, **skeleton-based action recognition** stands out as particularly efficient and concise, especially for actions that do not involve objects or

Publication. Matteo Mosconi, Andriy Sorokin, **Aniello Panariello**, *et al. Mask and Compress: Efficient Skeleton-Based Action Recognition in Continual Learning.* ICPR (Oral Presentation), 2024 [186].

Candidate contribution. Idea, methodology, implementation, experimental design, and writing.

scene context. Skeleton sequences capture the trajectories of key points (*i.e.*, joints) in the human body (*e.g.*, elbows, knees, wrists) [283]. Since joints can be represented by 2D or 3D spatial coordinates, skeletal data are more efficient than images because of the sparsity of skeleton graphs. Moreover, this representation is robust to changes in appearance, cluttered backgrounds, and occlusions while remaining inherently privacy-preserving [250].

The traditional learning approach to HAR assumes that all necessary data is readily available during training. However, this assumption often does not hold in real-world contexts, as instances or classes may emerge incrementally over time. In such a dynamic scenario, deep neural networks struggle to acquire new knowledge without displacing capabilities learned in earlier stages. This phenomenon, widely known as *catastrophic forgetting*, degrades performance and lies at the core of **continual learning**. In this setting, models must adapt to a sequence of tasks while preserving performance on previously seen ones.

While tasks such as classification [123, 225, 25, 270, 245] and video-based action recognition [201, 35, 262] have been widely explored in continual-learning settings, skeleton-based HAR has received comparatively limited attention. Although the authors of [144] took initial steps toward this problem, they rely on an expandable architecture that appends a new learnable module to the network whenever a new class appears. While this strategy helps alleviate catastrophic forgetting, the computational footprint of the model grows gradually, making the approach memory-hungry and poorly scalable. Additionally, their setting imposes constraints that diverge from realistic scenarios. Specifically, they pretrain the network on most training instances and reserve only a few classes for the incremental stage.

This chapter exploits the structure of skeletal data to store samples efficiently in an episodic memory, *i.e.*, a continuously updated buffer containing a small subset of past data. Memory efficiency is improved by compressing skeleton sequences through temporal sub-sampling, leveraging their redundancy in time [126]. During rehearsal, retained samples are reconstructed via linear interpolation, which introduces negligible overhead and requires no additional parameters.

Temporal redundancy is further leveraged through a masking strategy inspired by masked autoencoding [97, 6, 255]. Unlike prior skeleton masking works focused only on pretraining [280, 287], the reconstruction objective is optimized jointly with the recognition loss during continual learning. This design reduces training requirements and acts as an auxiliary regularizer that stabilizes the encoder representations.

At the end of each task, a lightweight linear probing phase is introduced to align the classifier with test-time conditions. Since the encoder is trained on masked sequences but evaluated on full inputs, the classifier may suffer from covariate shift [111], particularly at high masking ratios. To mitigate this effect, encoder parameters are frozen and the classifier is re-optimized using unmasked sequences; this step updates only a small number of parameters while yielding a consistent improvement in performance.

The approach is evaluated on incremental versions of NTU RGB+D 60 [233] and NTU RGB+D 120 [161], achieving state-of-the-art performance for class-incremental skeleton-based action recognition.

We remark on the following main contributions:

- We reduce the memory footprint of skeleton sequences in the episodic buffer via temporal sub-sampling and interpolation-based reconstruction.
- We introduce a masked modeling objective for skeletal data that is optimized jointly with rehearsal-based continual learning.
- We employ a lightweight linear probing phase to align the classifier with test-time inputs under masking-based training.

7.2 Related work

Skeleton-based Action Recognition. In early skeleton-based action recognition works, sequences were treated as time series and were therefore processed with Recurrent Neural Networks (RNNs) [67, 309, 103, 50] to capture temporal dynamics. These approaches struggled to integrate the spatial context of joints and proved slow and difficult to parallelize. Subsequent works exploited Convolutional Neural Networks (CNNs) [119, 115], treating skeletal data in various ways to make them compatible with CNNs; some handle coordinates as image channels [66, 141], while others reshape skeletons by combining joints in space and time [115].

However, these models faced a common limitation: they failed to effectively represent the relationships between skeletal joints moving together over time. Graph Convolutional Networks (GCNs) address these shortcomings by exploiting nodes (*i.e.*, joints) both temporally and spatially [288, 68, 69, 43, 236]. Subsequently, the emergence of ViT [64] introduced transformer-based architectures into computer vision, leading to solutions that integrate self-attention layers into convolutional architectures. One such work, STTFormer [209], divides the sequence into tuples of joints and retains some

CNN concepts (*i.e.*, pooling aggregation) for temporal feature processing. Nonetheless, this approach under-exploits the sparsity and redundancy of skeletal data. In recent years, masking approaches [280, 287] have been employed to take advantage of these characteristics for pretraining. In contrast, this chapter adopts the reconstruction objective during downstream continual learning, reducing training requirements while avoiding a separate pretraining stage.

Continual Learning. Classical CL methods employ a regularization term that penalizes the alterations of weights to avoid forgetting [123, 303, 230]. Rehearsal methods [224, 215, 25, 5], on the other hand, employ a limited memory buffer in which they store samples from past tasks and replay them. Another paradigm is represented by dynamic architectures [225, 20] in which new network components are instantiated for each incoming task; unfortunately, this often leads to a rapid increase in the number of parameters. This approach has been employed by the authors of Else-Net [144] to tackle skeleton-based HAR in Class-IL. They use the first 50 classes of NTU RGB+D 60 to pretrain their network, and perform incremental training across 10 tasks, each focusing on a different class.

7.3 Method

7.3.1 Preliminaries

Class-Incremental Learning. We adopt the standard Class-IL setting and notation introduced in Chapter 6. At each task, training is performed on the current data together with a small episodic memory buffer of past samples, following rehearsal-based learning.

Spatio-Temporal Tuples Transformer (STTFormer). We adopt STTFormer [209] as the main backbone of our architecture. It is a transformer-based model designed for skeleton-based action recognition and exploits self-attention to capture cross-joint correlations across adjacent frames. Specifically, a raw skeleton sequence $x \in \mathbb{R}^{C \times F \times V}$, where C is the number of channels (*i.e.*, spatial coordinates), F is the number of frames, and V is the number of joints, is given as input to the model. This sample is divided into tuples, *i.e.*, sequences of n adjacent frames:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{\lfloor F/n \rfloor}], \text{ where } \mathbf{x}_i \in \mathbb{R}^{C \times n \times V}. \tag{7.1}$$

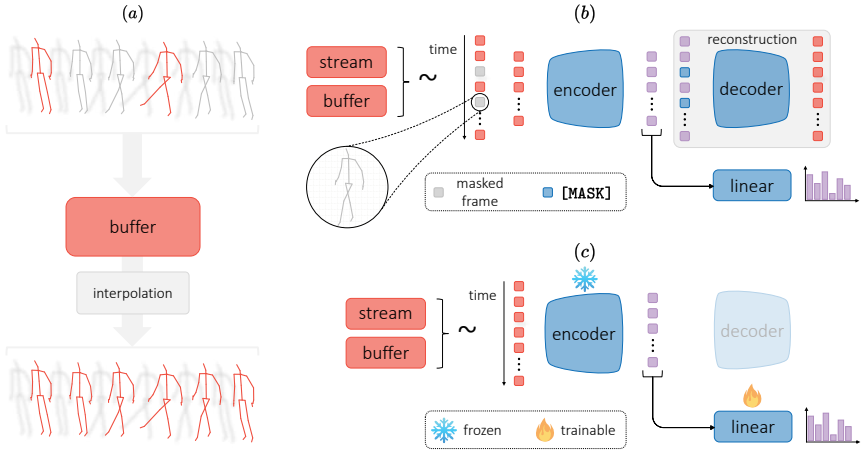


Figure 7.1: Overview of the key components of CHARON. Our efficient buffer strategy is shown on the left (a). In the upper right (b), we illustrate the training phase with reconstruction regularization, while linear probing is displayed at the bottom (c).

Each layer of STTFormer comprises two distinct modules, which target either *intra*- or *inter*-tuple relationships. Every element of $\mathbf{X}$ (*i.e.*, each tuple) is first fed to a self-attention layer, which attends the joints in $\mathbf{x}_i$. This phase aims to model the *intra*-tuple characteristics. Then, an *inter*-tuple representation is extracted via temporal pooling.

7.3.2 CHARON

In this section, we present CHARON, which encompasses three components: (i) a technique to populate the memory buffer, employing linear interpolation to decompress memory samples; (ii) an efficient training phase with masked inputs; (iii) a linear probing stage, which refines the classifier and updates the logits stored in the memory buffer. We depict these elements in Fig. 7.1.

Efficient buffer. A raw skeleton sequence $x \in \mathbb{R}^{C \times F \times V}$ collects the C coordinates (*e.g.*, xyz in Split NTU-60 and Split NTU-120) of V joints at F time instants. Unlike RGB video frames, skeletal data inherently reside in Euclidean space, where the concept of distance between points (three-dimensional joints in our case) is well defined. Additionally, skeleton sequences often exhibit temporal redundancy [90]. In light of these characteristics, skeletal data can be compressed when needed; in our case, this happens

before storing a sequence in the memory buffer. Notably, the compressed sequences can also be reconstructed with minimal loss through simple linear interpolation. In particular, even with a sampling interval of $s = 5$ frames – *i.e.*, retaining one frame every five and thus yielding a compression ratio of 80% – the reconstructions remain close to the raw samples. Based on this observation, a larger number of instances can be stored under the same memory constraints: in other words, we can accumulate up to s times as many samples in the buffer.

When a sample is extracted from the buffer for rehearsal, we reconstruct it to obtain F frames again and then treat it as a complete sample. Since linear interpolation does not require learnable parameters, reconstructing temporal skeletal sequences has low computational cost.

Training phase. As we mentioned above, a transformer-based architecture founded on [209] is adopted as our backbone. We build upon it to derive an encoder-decoder framework inspired by masked autoencoders [97]. Notably, this allows us to reduce the computational effort during training, as depicted in Fig. 7.2. Specifically, given a sample x coming from the current task or the buffer, the first step consists of a linear projection, followed by positional encoding to inject temporal dependencies. Afterward, we feed the encoder $e(\cdot;\theta_e)$ with a temporally masked sample $\tilde{x} \in \mathbb{R}^{C\times\lfloor(1-\eta)\cdot F\rfloor\times V}$ obtained by dropping a random subset of frames from the input sequence, where $\eta \in [0, 1)$ is the masking ratio.

The encoder projects the input $\tilde{x}$ into the latent space, obtaining features $\tilde{h} = e(\tilde{x};\theta_e)$. At this point, the architecture splits into two branches: the first (*recognition*) uses a fully connected layer $f(\cdot;\theta_f)$ to yield pre-softmax logits $z = f(\tilde{h};\theta_f)$. The second (*reconstruction*) implements the self-supervised regularization through a decoder module $d(\cdot;\theta_d)$. Specifically, given the latent feature vector $\tilde{h}$, which has $\lfloor(1-\eta)\cdot F\rfloor$ tokens, the decoder input is formed by filling the missing positions with learnable mask vectors denoted by `[MASK]`. We place these vectors in the same positions as the original masked ones, $h = \textsc{concat}(\tilde{h}, \texttt{[MASK]})$. The training objective is:

$$\mathcal{L}_{\text{stream}} = \mathcal{L}_{\text{CE}}(z, y) + \gamma \cdot ||d(h;\theta_d) - x||_2^2, \tag{7.2}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss and γ is a hyperparameter controlling the impact of the reconstruction loss.

To mitigate forgetting, we incorporate the objective defined in Eq. 7.2 into a rehearsal-based framework. Drawing inspiration from [25], we retrieve a mini-batch

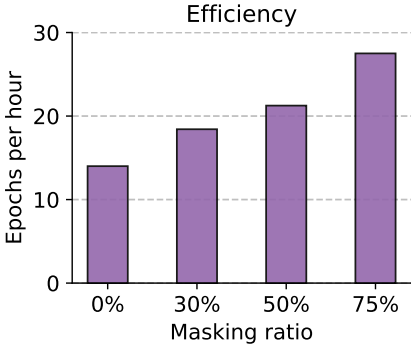


Figure 7.2: Epochs per hour at different masking ratios (GTX 1080 Ti GPU).

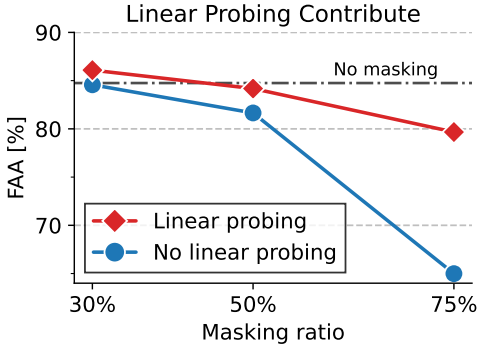


Figure 7.3: Contribution of linear probing during joint training with varying masking ratios.

of samples $x_{\mathcal{M}}$ from the memory buffer at each training step. This mini-batch includes associated predictions $z_{\mathcal{M}}$ (*i.e.*, logits) and labels $y_{\mathcal{M}}$, which are added to the episodic memory along with the corresponding samples. The loss functions for these two components are:

$$\mathcal{L}_{\text{logits}} = ||f(\tilde{h}_{\mathcal{M}};\theta_f) - z_{\mathcal{M}}||_2^2 \; + \; \gamma \cdot ||d(h_{\mathcal{M}};\theta_d) - x_{\mathcal{M}}||_2^2, \tag{7.3}$$

$$\mathcal{L}_{\text{labels}} = \mathcal{L}_{\text{CE}}(f(\tilde{h}_{\mathcal{M}};\theta_f), y_{\mathcal{M}}) + \gamma \cdot ||d(h_{\mathcal{M}};\theta_d) - x_{\mathcal{M}}||_2^2. \tag{7.4}$$

The mini-batch of samples $x_{\mathcal{M}}$ undergoes the same pipeline as the input stream x, producing the latent features $\tilde{h}_{\mathcal{M}} = e(\tilde{x}_{\mathcal{M}};\theta_e)$ and $h_{\mathcal{M}} = \text{CONCAT}(\tilde{h}_{\mathcal{M}}, [\text{MASK}])$.

The final objective of this phase is:

$$\mathcal{L} = \mathcal{L}_{\text{stream}} + \alpha \cdot \mathcal{L}_{\text{logits}} + \beta \cdot \mathcal{L}_{\text{labels}}, \tag{7.5}$$

where α and β are two balancing hyperparameters.

Linear probing. As described above, the model is trained with partial skeleton sequences. While this provides an efficient training strategy, it may hinder performance during evaluation. In particular, we argue that the classification head $f(\cdot;\theta_f)$ may become misaligned because training and test conditions differ (masking *on* during training, masking *off* at test time). To address this issue, highlighted in Fig. 7.3, we devise an auxiliary linear-probing stage at the end of each task, lasting a few epochs (*i.e.*, 10% of the number used for the main training stage). During this phase, only the

classifier parameters are updated, while the encoder remains frozen. We therefore feed each full (*i.e.*, unmasked) sample $x \in \mathbb{R}^{C \times F \times V}$ to the encoder.

In formal terms, as in the main training phase, the encoder projects the input x into the latent space, obtaining hidden features $h = e(x; \theta_e)$. The fully connected layer $f(\cdot; \theta_f)$ then produces the logits $z = f(h; \theta_f)$, to which a cross-entropy loss is applied. In this phase, we still employ the regularization from [25]. Thus, the resulting objective $\mathcal{L}_{\text{lp}}$ can be written as:

$$\mathcal{L}_{\text{lp}} = \mathcal{L}_{\text{CE}}(z, y) + \alpha \cdot ||f(h_{\mathcal{M}}; \theta_f) - z_{\mathcal{M}}||_2^2 \; + \; \beta \cdot \mathcal{L}_{\text{CE}}(f(h_{\mathcal{M}}; \theta_f), y_{\mathcal{M}}). \tag{7.6}$$

Traditional works using masked autoencoders [97, 255] typically distinguish between a pretraining phase and a linear-probing phase used to adapt to downstream tasks. However, we argue that keeping these stages separate results in a more cumbersome approach, potentially undermining the efficiency we seek. To address this, Eqs. 7.5 and 7.6 are computed sequentially during each task according to the incremental setting (*i.e.*, while retaining access only to the data of the current task).

7.4 Experimental analysis

Datasets. We evaluate on the standard class-incremental skeleton action recognition benchmarks derived from NTU RGB+D 60 and 120, using the established task splits and protocols described in Chapter 6. We report results for the cross-subject (XSub) and cross-view (XView) data modalities [233] for Split NTU-60, and cross-subject (XSub) and cross-setup (XSet) [161] for Split NTU-120.

Implementation details. Our custom STTFormer [209] reduces the width of intermediate layers to obtain a more lightweight model. We set the number of frames in each tuple $n = 6$. Following the asymmetric design proposed in [97], we employ 8 layers for the encoder and 3 for the decoder. Unless otherwise specified, architectural hyperparameters follow the original STTFormer design choices. Additionally, we set $\alpha = 0.3$ and $\beta = 0.8$ in Eqs. 7.5 and 7.6, while using $\gamma = 0.5$ in the variants employing the reconstruction regularization (Eqs. 7.2 to 7.4). We adopt a batch size of 16 and an SGD optimizer with a learning rate of 0.05. Each task in the incremental setting lasts 30 epochs, followed by 3 epochs of linear probing. Finally, for data augmentation, we follow the original STTFormer implementation.

	Split NTU-60				Split NTU-120			
Method	**XView**		**XSub**		**XSet**		**XSub**	
FT	16.05$_{\pm 0.07}$		15.64$_{\pm 0.05}$		7.19$_{\pm 0.06}$		6.97$_{\pm 0.23}$	
JT	84.75$_{\pm 0.02}$		77.32$_{\pm 0.54}$		71.18$_{\pm 1.07}$		70.15$_{\pm 0.98}$	
$\mathcal{M}_{size}$	500	2000	500	2000	500	2000	500	2000
iCaRL	51.54$_{\pm 1.3}$	53.41$_{\pm 1.1}$	47.12$_{\pm 1.4}$	50.69$_{\pm 1.2}$	32.91$_{\pm 0.9}$	34.74$_{\pm 0.7}$	33.03$_{\pm 1.3}$	36.68$_{\pm 1.0}$
Else-Net	40.81$_{\pm 0.8}$	59.10$_{\pm 0.2}$	39.72$_{\pm 0.4}$	57.00$_{\pm 1.0}$	19.37$_{\pm 0.6}$	33.52$_{\pm 0.6}$	18.43$_{\pm 0.7}$	33.95$_{\pm 0.3}$
ER	51.00$_{\pm 1.6}$	68.27$_{\pm 0.1}$	45.80$_{\pm 0.5}$	62.74$_{\pm 1.9}$	26.35$_{\pm 1.1}$	43.12$_{\pm 0.4}$	26.19$_{\pm 1.7}$	45.06$_{\pm 0.7}$
DER	51.36$_{\pm 0.9}$	66.74$_{\pm 0.1}$	49.97$_{\pm 1.9}$	63.48$_{\pm 1.3}$	27.83$_{\pm 1.7}$	40.19$_{\pm 0.9}$	30.10$_{\pm 1.5}$	36.10$_{\pm 1.8}$
DER++	60.41$_{\pm 0.5}$	73.09$_{\pm 1.3}$	57.22$_{\pm 1.0}$	67.64$_{\pm 1.6}$	34.27$_{\pm 1.4}$	50.06$_{\pm 0.6}$	36.29$_{\pm 0.3}$	49.81$_{\pm 0.8}$
CHARON	**73.60**$_{\pm 0.3}$ **+ 13.19**	**77.77**$_{\pm 0.2}$ **+ 4.68**	**68.30**$_{\pm 0.6}$ **+ 11.08**	**72.70**$_{\pm 0.2}$ **+ 5.06**	**52.19**$_{\pm 0.6}$ **+ 17.92**	**61.63**$_{\pm 0.1}$ **+ 11.57**	**48.64**$_{\pm 0.0}$ **+ 12.35**	**59.23**$_{\pm 0.4}$ **+ 9.42**

Table 7.1: FAA (%) results on Split NTU-60 and Split NTU-120. For **CHARON**, we report the results with a masking ratio equal to 30%. We highlight in green the gains achieved by our approach w.r.t. the best competing method.

7.4.1 Results

For the experimental comparison, we use Joint Training (JT) as the upper bound for our approach. It consists of training the model on the unified dataset (*i.e.*, without splitting it into tasks). As a lower bound, we adopt an incremental training approach that does not employ tailored techniques against catastrophic forgetting. We refer to it as Fine-Tuning (FT).

In Tab. 7.1 we report the results for buffer sizes $\mathcal{M}_{size} \in \{500, 2000\}$. Following prior work [25, 144, 21], we report Final Average Accuracy (FAA) as defined in Chapter 6. We repeat each experiment three times and report mean and standard deviation. As outlined by Tab. 7.1, Else-Net [144] does not reproduce the performance reported under its original protocol, which includes extensive pretraining; under a standard from-scratch class-incremental setting, its performance degrades substantially.

Furthermore, even classical replay methods such as iCaRL [215], ER [219], DER, and DER++ [25] outperform Else-Net. CHARON achieves state-of-the-art performance in class-incremental skeleton-based action recognition across both Split NTU-60 and Split NTU-120. In particular, this holds when employing a masking ratio of 30%; for higher percentages, we observe a decrease in performance, as discussed below. Significantly, the largest improvement is observed with a buffer size of 500 (surpassing the second-best method, *i.e.*, DER++, even when it uses a buffer size of 2000). This

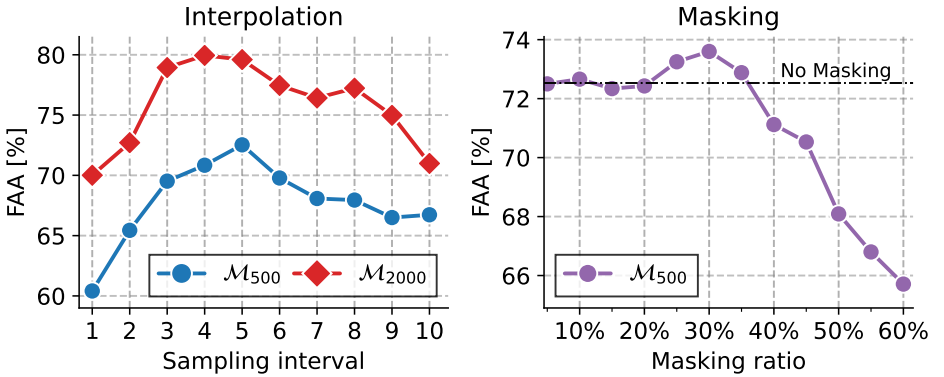


Figure 7.4: (*left*) FAA for the DER++ baseline employing different values of the sampling interval s. (*right*) FAA obtained by CHARON as the masking ratio varies.

highlights the pivotal role of sample quantity in the effectiveness of replay methods. Consequently, it underscores the importance of techniques that increase the number of stored samples under a fixed buffer budget.

On the sampling interval. To further evaluate the effectiveness of our buffer strategy, we conduct a comparative study by varying the *sampling interval* s (which we recall indicates the step length in the uniform sampling procedure). Given $s \in \mathbb{N}^+$, we obtain the *compression ratio* as:

$$\textit{compression ratio} = \frac{s-1}{s} \cdot 100. \tag{7.7}$$

We report in Fig. 7.4 (*left*) the FAA obtained at different sampling intervals s for both tested buffer sizes. For each value of s, we scale the buffer size accordingly (as documented in Sec. 7.3.2). For instance, when $s = 10$, a memory with a nominal capacity of 500 examples could contain up to $s \cdot 500 = 5000$ compressed examples. As can be appreciated, the sampling interval $s = 5$ (*i.e.*, 80% compression) yields the best final accuracy. Namely, when sampling one skeletal pose every five frames, the memory buffer attains the best compromise between sample *fidelity* (favored by lower sampling intervals) and sample *diversity* (*i.e.*, higher intervals). Moreover, we note that introducing a compression phase ($s > 1$) brings a stable and substantial gain w.r.t. the standard replay paradigm ($s = 1 \rightarrow$ no compression), suggesting that moderate compression improves replay also by broadening temporal coverage. This result highlights the central role of the trade-off between sample quality and quantity.

	Masking ratio	
	30%	60%
w/o recon. loss	70.61	61.59
CHARON	**73.60**	**65.72**

Table 7.2: Impact of the reconstruction loss at different masking ratios.

	Position	
Strategy	*pre*	*post*
Deterministic	72.08	72.43
Random	71.89	**73.60**

Table 7.3: Ablative outcomes about sampling strategy and masking position.

On the masking ratio. We next assess the impact of the *masking ratio*, which indicates the fraction of frames discarded before feeding the input sequence to the model. The results are illustrated in Fig. 7.4 (*right*) and show an increase in performance up to 30%. For higher masking ratios, performance begins to decline despite the notable efficiency gains (see Fig. 7.2). Quantitatively, even with 50% masking, CHARON achieves an acceptable final average accuracy of around 68%, which decreases to $\approx 66\%$ with a masking ratio of 60%.

7.4.2 Ablations

We report the following ablation studies; all experiments are performed on the XView modality of Split NTU-60.

On the importance of the reconstruction-based objective. Our approach not only seeks strong classification performance but also includes an auxiliary reconstruction term targeting the entire input sequence. To better understand the effect of this auxiliary objective, we provide an ablation in which we discard both the decoder module and the reconstruction loss. In doing so, we still apply random masking (testing ratios of 30% and 60%) and linear probing at the end of each task.

The results of these ablative studies are reported in Tab. 7.2: remarkably, CHARON experiences a significant performance drop when removing the decoder and the reconstruction loss, especially for the higher masking ratio of 60%. We consider such a finding as noteworthy, as it highlights the importance of auxiliary learning techniques when leveraging higher compression ratios to pursue efficiency.

Masking strategy and positioning. Our approach adopts a random masking strategy to drop frames, following most of the literature on masked autoencoders. Here we compare it with a deterministic strategy that drops one frame every k frames. We also assess different points in the pipeline at which the masking operation can be introduced. Specifically, *post* indicates that masking is applied after splitting the sequence into

tuples (see Sec. 7.3.1), as done in our approach. Results for the combinations of these two alternatives are reported in Tab. 7.3: the random strategy with post-hoc masking emerges as the best configuration.

7.5 Conclusions

Skeleton-based action recognition is a relevant task in modern human-centric learning systems, and continual learning provides a natural framework for scenarios in which the set of actions evolves over time. In this chapter, we introduced CHARON, a rehearsal-based approach for class-incremental skeleton action recognition that combines temporal sub-sampling for memory-efficient buffering, masked training with an auxiliary reconstruction objective, and a lightweight linear probing stage to align the classifier with test-time conditions.

The experimental analysis demonstrates that these components jointly reduce memory usage and training cost while achieving state-of-the-art final average accuracy on NTU RGB+D 60 and 120. In particular, the reconstruction-based regularization stabilizes learning under aggressive masking ratios, enabling an effective trade-off between efficiency and performance in memory-constrained continual settings.

More broadly, this chapter highlights how exploiting the intrinsic structure of the data stream can substantially improve replay-based continual learning. While CHARON leverages temporal redundancy in skeletal sequences to compress and reconstruct past samples, the same principle extends to higher-level representations learned by large pretrained models. In the following chapter, we shift focus from structured input data to structured embedding spaces, and study continual learning in the context of vision–language models. There, adaptation is performed directly at the representation level, with the objective of preserving both past knowledge and zero-shot generalization capabilities while incrementally learning new classes.

8

CLIP with Generative Latent Replay: A Strong Baseline for Incremental Learning

8.1 Forgetting both the past and the future

Pretrained vision–language models (VLMs) such as CLIP [210] provide strong zero-shot capabilities. This allows them to achieve strong continual-learning performance without any fine-tuning [252], thereby largely avoiding forgetting by design. However, for tasks that deviate from CLIP's pretraining (*e.g.*, satellite and medical domains), adaptation is essential. In this context, incrementally fine-tuning CLIP models presents an additional challenge: *catastrophic forgetting can break the model's zero-shot capabilities*, harming performance on other tasks and domains [314].

Motivated by this challenge, this chapter introduces a simple approach to incremental fine-tuning of CLIP. Inspired by CoOp [317], we freeze both the visual and

Publication. Emanuele Frascaroli, **Aniello Panariello**, *et al. CLIP with Generative Latent Replay: A Strong Baseline for Incremental Learning*. BMVC (Oral Presentation), 2024 [77].

Candidate contribution. Methodology, implementation, experiments, and writing.

text encoders and learn class-specific textual prompts fed to the text encoder. This approach allows the model to maintain enough plasticity to adapt to new domains while remaining stable enough to preserve its original zero-shot capabilities. Moreover, as tasks progress and the model learns new classes, we use the corresponding learned prompts for previously observed classes. For unseen ones, by contrast, we rely on hand-crafted prompts (*e.g.*, "`a photo of a <CLS>`"), resulting in a hybrid prompting approach.

This chapter considers continual learning in the context of large pretrained vision–language models, in which adaptation primarily affects representation alignment rather than low-level feature extraction. In contrast to approaches that exploit structure in the input space, forgetting is addressed through replay-based mechanisms operating directly in embedding space, with the objective of preserving zero-shot capabilities while enabling incremental adaptation to new tasks.

In addition, inspired by [305], we bridge the gap with joint training through the use of multiple lightweight generative models. Specifically, for each new class, we train a lightweight Variational Autoencoder (VAE) [121] to model the distribution of CLIP's visual embeddings.

The proposed methodology, named **Continual Generative training for Incremental prompt-Learning** (CGIL), is evaluated on various standard Class-IL benchmarks, showing strong performance even in domains in which zero-shot CLIP fails. Moreover, following [314, 297] we introduce an additional metric to assess the zero-shot capabilities on future tasks. We evaluate in this setting all competitors with zero-shot capabilities (*i.e.*, the ones relying on a VLM), showing the effectiveness of our strategy.

8.2 Preliminaries: prompt learning with CLIP

Contrastive Language-Visual Pretraining (CLIP). CLIP [210] consists of a visual encoder $E_{vis}(\cdot)$ (which can be either a ViT or a CNN) and a Transformer [260] text encoder $E_{txt}(\cdot)$. They are trained with a contrastive objective on image-text pairs to obtain aligned latent embeddings. Given an image $\mathbf{x}$, we denote by $z_{vis} = E_{vis}(\mathbf{x})$ the embedding produced by the visual encoder.

Once trained, CLIP can be used for zero-shot classification by computing the cosine similarity between the visual and textual embeddings. In particular, for each candidate class, a text prompt is created by embedding the class label into a template like "a

photo of a <CLS>". The resulting class-level prompts are fed to the text encoder, producing a textual representation z^c_{txt} for each class y^c. This formulation enables classification without task-specific training data, relying entirely on the semantic alignment learned during pretraining. However, performance is highly sensitive to the choice of textual prompts, motivating learning-based strategies to adapt prompts to downstream tasks while keeping the encoders fixed. The posterior probabilities are computed as the cosine similarity (noted as $\langle \cdot, \cdot \rangle$) between visual and class-level textual representations:

$$p(y^c|\mathbf{x}) = \frac{\exp(\langle z^c_{txt}, z_{vis} \rangle / \tau)}{\sum_{j=1}^{|\mathcal{Y}|} \exp(\langle z^j_{txt}, z_{vis} \rangle / \tau)}, \tag{8.1}$$

where τ is a temperature parameter and $\mathcal{Y}$ represents the set of classes.

Tuning CLIP with textual prompts. Prompt learning techniques allow efficient fine-tuning of large pretrained models. Among these methods, CoOp [317] stands out as particularly effective. In a nutshell, CoOp does not rely on hand-crafted prompts to generate the input for the CLIP text encoder, but rather on learnable context vectors V. These context vectors are concatenated with the label token $[CLS]$ and learned through gradient descent, using the similarity scores from Eq. 8.1 as logits for the cross-entropy loss.

8.3 CGIL: generative replay meets prompt learning

Fig. 8.1 depicts our approach, termed **Continual Generative training for Incremental prompt-Learning** (CGIL), which comprises two main phases:

Phase 1: *generative modeling.* The proposed approach performs replay directly in representation space. Using all images from the current task, the corresponding visual embeddings are extracted through the CLIP visual encoder and grouped by class. For each class, a lightweight Variational Autoencoder (VAE) [121] is trained to model the class-conditional distribution of visual embeddings. The VAE's encoder and decoder are lightweight, consisting of only three fully connected layers interleaved with LeakyReLU activations. Once the VAEs are trained, we discard the encoders and retain the decoders in a memory buffer. The decoders are then used to sample new data points from the respective priors during the subsequent alignment phases.

Phase 2: *prompt alignment.* We learn the context vectors for the text encoder. However, instead of computing image embeddings from real images, we sample synthetic

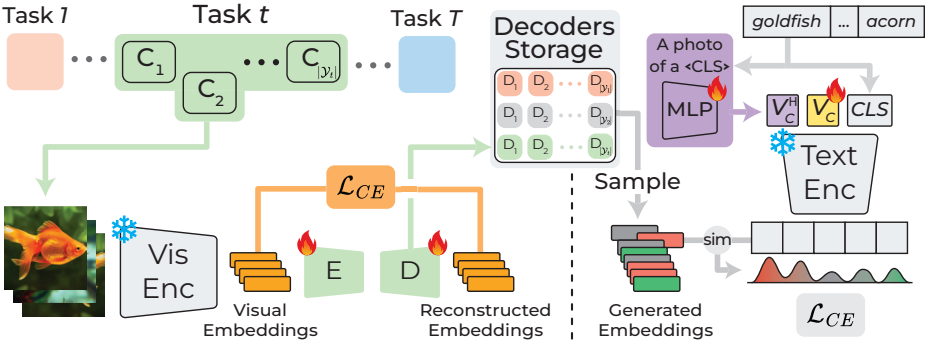


Figure 8.1: Training of the generative models (**left**) and prompt alignment (**right**). For each class C_i of task t, we train a class-specific generative model (*i.e.*, a VAE). Afterwards, only the decoders are retained for later generative replay, while the encoders can be discarded. In the second phase, we perform prompt alignment by matching the features sampled from all stored decoders up to task t with the text features generated using the learnable prompts.

embeddings from all VAEs, thus including data from both past and current classes. In detail, the prompt construction involves generating two distinct tokens: (i) a learnable class-specific token V_c to capture fine-grained details, and (ii) a hyper-token V_c^H that models cross-domain knowledge. The hyper-token is generated by a shared, learnable Multi-Layer Perceptron (MLP), fed with the textual embedding z_{txt} obtained through the standard hand-crafted prompt. Overall, the prompt $\mathbf{t}_c$ for class c fed to the text encoder E_{txt} is:

$$\mathbf{t}_c = [V_c^H]\ [V_c]\ [CLS], \tag{8.2}$$

$$\text{where} \quad V_c^H = \text{MLP}(E_{txt}(\text{“a photo of a <CLS>”})). \tag{8.3}$$

The posterior probability of class c is obtained as in Eq. 8.1, using the text embeddings obtained with our prompts $z_{txt}^c = E_{txt}(\mathbf{t}_c)$.

The two phases are repeated at each task to refine previously learned prompts with knowledge from subsequent tasks. It is worth noting that training becomes considerably faster because visual embeddings are sampled from the VAEs rather than computed through the CLIP visual encoder, allowing us to avoid costly forward passes through the deep visual backbone.

Zero-shot inference. During inference, we adopt a **hybrid** approach to deal with both seen and unseen classes. For classes the model has encountered in previous tasks,

we employ the corresponding learned prompts; for those it has not yet encountered, we feed the text encoder with the original hand-crafted prompts (*e.g.*, "`a photo of a <CLS>`"). Such a straightforward approach allows us to preserve the zero-shot capabilities of CLIP while adapting to novel classes that arrive sequentially.

8.4 Experiments

Datasets. We evaluate our approach across a wide range of datasets with varying degrees of similarity to ImageNet pretraining [54, 191] and varying compatibility with CLIP's zero-shot capabilities [210] (see Tab. 8.1). In particular, we test on Seq. ImageNet-R, Seq. Cars-196, Seq. CUB-200, Seq. EuroSAT, and Seq. ISIC.

Metrics. When evaluating in the Class-IL scenario, performance is summarized using the Final Average Accuracy (FAA), defined in Chapter 6. Additionally, we assess zero-shot performance on future (unseen) tasks by adapting the **Transfer** metric from [314, 297], originally introduced to evaluate zero-shot capabilities across different domains. Specifically, let A_t^i be the accuracy on the i-th task after being trained until task t, the **Class-Incremental Transfer** is defined as:

$$\text{CI-Transfer} = \frac{1}{T-1}\sum_{t=1}^{T-1}\left(\frac{1}{T-t}\sum_{i=t+1}^{T} A_t^i\right). \tag{8.4}$$

Comparison methods. We benchmark our model against several prompt-tuning approaches, including L2P [270], DualPrompt [269], CODA-Prompt [245], and AttriCLIP [266]. Additionally, we assess models that fine-tune the entire architecture, namely LwF [148], GDumb [206], DER++ [25], and SLCA [305]. We also integrate MoE-Adapters [297] into our framework, which is designed to prevent zero-shot accuracy degradation across datasets. AttriCLIP, MoE-Adapters, and our CGIL are also evaluated on future tasks, measuring how their zero-shot capabilities are affected by incremental training. We employ the same ViT-L/14 backbone for each model that uses CLIP.

8.4.1 Results

Tab. 8.1 reports the Class-IL performance for all evaluated competitors and benchmarks. The last column shows the average accuracy of each method across all benchmarks. Despite the impressive results of zero-shot CLIP on Seq. ImageNet-R and Seq. Cars-196,

Method	Img-R	Cars-196	CUB-200	EuroSAT	ISIC	*Avg.*
Zero-shot CLIP [210]	81.95	64.99	50.52	53.32	26.59	55.47
LwF † [148]	19.09	23.24	16.73	25.13	33.06	23.45
GDumb † [206]	44.28	28.74	61.34	90.99	61.64	57.40
DER++ † [25]	56.66	53.66	74.62	93.08	65.68	68.74
L2P [270]	66.49	38.18	62.21	46.34	47.13	52.07
DualPrompt [269]	68.50	40.14	66.00	71.39	49.99	59.20
CODA-Prompt [245]	75.45	31.99	67.30	63.12	44.87	56.55
AttriCLIP [266]	87.39	75.63	58.28	72.33	28.26	64.38
SLCA † [305]	77.00	67.73	**84.71**	88.69	59.19	75.46
MoE Adapters [297]	**90.67**	77.76	64.98	80.56	34.52	69.70
CGIL	89.42	**89.27**	83.12	**96.17**	**73.03**	**86.20**

Table 8.1: The FAA on the tested benchmarks. † denotes methods that fine-tune the whole model, while other methods apply parameter-efficient techniques.

CI-Transfer	Img-R	Cars-196	CUB-200	EuroSAT	ISIC	*Avg.*
Zero-shot CLIP [210]	82.14	66.16	50.86	55.00	22.42	55.32
AttriCLIP [266]	85.75	73.98	54.07	59.69	24.14	59.53
MoE Adapters [297]	**88.25**	75.82	61.73	55.77	21.06	60.53
CGIL	86.71	**78.80**	**66.34**	**71.52**	**48.18**	**70.31**

Table 8.2: The Class-Incremental Transfer on the tested benchmarks. Only methods based on CLIP as a backbone could be tested.

it fails in other domains, particularly in the medical field. Consequently, competitors that rely on CLIP are heavily affected by this limitation and exhibit a similar drop. On the other hand, CGIL effectively addresses CLIP-related limitations, delivering consistently strong performance across all scenarios. Considering average performance, the proposed method achieves a substantial improvement (+ **11**) over SLCA [305], while other prompt-based techniques fall behind.

Zero-shot performance. Tab. 8.2 reports performance on unseen tasks through the Class-Incremental Transfer metric. Here, a similar trend emerges, with CLIP and the other competitors excelling on Seq. ImageNet-R and Seq. Cars-196 while struggling on the remaining datasets. CGIL shows a strong ability to leverage both CLIP's zero-shot expertise and knowledge acquired from previously learned tasks, achieving superior performance in nearly all benchmarks.

	Img-R	Cars-196	CUB-200	EuroSAT	ISIC	*Avg.*
CoOp (*Joint*)	89.24	89.89	82.52	96.25	73.57	86.29
CoOp (*Finetune*)	84.94	68.61	59.84	79.27	37.08	65.95
CGIL	**89.42**	**89.27**	**83.12**	**96.17**	**73.03**	**86.20**
Different generative approaches						
Multinomial Gaussian	83.89	82.59	80.06	85.70	51.89	76.83
Mixture of Gaussians	88.54	88.82	82.10	93.04	62.42	82.98
Diffusion Models	89.28	90.14	83.48	95.73	68.99	85.52
VAEs (CGIL)	89.42	89.27	83.12	96.17	73.03	86.20
Different techniques to create the context prompt						
Class-specific token	89.09	88.96	83.06	95.59	72.21	85.78
MLP-generated token	89.41	88.91	82.82	95.69	70.79	85.52
Multiple shared tokens	89.02	88.08	81.50	95.47	70.18	84.85
CGIL	89.42	89.27	83.12	96.17	73.03	86.20

Table 8.3: Ablative studies on CGIL. Results are expressed as Final Average Accuracy.

8.5 Model analysis

To further validate the effectiveness of CGIL and its architectural design, we report additional experiments in Tab. 8.3.

Detailed comparison with CoOp. Vanilla CoOp is evaluated under two distinct benchmarks: one trained jointly, *i.e.*, without partitioning the dataset into tasks (*Joint*), and the other trained in the conventional Class-IL scenario (*Fine-tune*). For the latter, we make two minor adjustments to accommodate the incremental scenario: (i) we allocate a class-specific learnable prompt to each class, rather than relying on a single global prompt, and (ii) during subsequent tasks, the previously learned prompts are kept frozen. This strategy, also employed in [245], helps prevent forgetting; conversely, training all contexts in subsequent tasks could overwrite previous knowledge.

The insights derived from the results of these two approaches (first rows of Tab. 8.3) highlight the effectiveness of CGIL in bridging the gap between fine-tuning and joint training when prompt learning is used. Indeed, the proposed method nearly matches the performance of the *Joint* setting.

Different generative models. Along with Variational Autoencoders, we evaluate various families of generative models to determine which best complements our method.

The most straightforward approach involves fitting a multivariate Gaussian distribution for each class [305]. As indicated in Tab. 8.3, this approach alone achieves strong results, with an average of 76.83 compared to 75.46 for SLCA. However, exploiting more powerful generative models such as VAEs considerably improves the effectiveness of the alignment procedure. This suggests that the quality and variety of the generated data are crucial.

We also evaluate Mixture of Gaussians (MoGs) and Denoising Diffusion Probabilistic Models (DDPMs) [102]. While DDPMs achieve results comparable to VAEs, we prefer VAEs because they train faster and require fewer parameters.

Different prompting techniques. As introduced in Sec. 8.3, our context consists of a class-specific token and a generated token. Hence, at the bottom of Tab. 8.3, we present the results with different choices: (i) using a single class-specific context, as in *CoOp (Finetune)*; (ii) utilizing only the hyper-token generated by the MLP; and (iii) adopting a method similar to the original CoOp, in which multiple tokens are learned and shared across classes. For the first two ablative variants, we increase the number of contextual tokens in our preliminary experiments. However, while the third variant, which uses a shared context across classes, benefited from this modification, the first two strategies showed no improvement. We thus report only the results with a single context token.

The results of these alternatives fall slightly behind CGIL, indicating that the main contributors are the generative rehearsal and the alignment phase. This becomes evident when considering the performance gap between CoOp *(Finetune)* and *Class-specific Context*: they share the same prompting mechanism, but the latter is enhanced with generative replay.

8.6 Conclusions

This chapter introduced CGIL, a simple yet effective baseline for incremental fine-tuning of vision–language models that preserves zero-shot capabilities while adapting to new classes. By operating directly in embedding space and combining prompt learning with generative replay, the proposed approach mitigates forgetting without modifying the pretrained encoders. In addition, the introduction of the Class-Incremental Transfer metric enables the evaluation of zero-shot generalization on future tasks, providing an overall assessment of continual-learning performance in VLMs.

A limitation of CGIL is its reliance on class-specific generative models, which must be stored for all observed classes. While the memory footprint of each decoder is modest and these models are only required during training, scalability remains a consideration as the number of classes grows. As we will explore in Chapter 11, the memory requirements can be significantly reduced by using a task-specific generative model that captures the distribution of all classes within a task, rather than individual models for each class. Nevertheless, even with class-specific generative models, inference remains efficient, requiring only a forward pass through the frozen CLIP visual encoder and a similarity computation with the learned textual embeddings.

More broadly, this chapter reinforces a central theme of this dissertation: effective continual learning can be achieved by carefully selecting the level at which adaptation occurs. While CHARON exploits structure in the input space of skeletal sequences, CGIL demonstrates that replay and adaptation can be successfully performed at the representation level of large pretrained models. The following part of the dissertation builds upon this insight by investigating how knowledge can be transferred across tasks and models through direct composition and merging, without relying on sequential retraining.

Summary of Part II: Continual learning through data and representations

This part investigated continual learning under the assumption of sequential task exposure, where models must adapt over time while preserving previously acquired knowledge. The focus was on rehearsal-based strategies, explored at different levels of abstraction and across heterogeneous modalities, highlighting how the choice of representation fundamentally shapes the design and effectiveness of continual learning algorithms.

The first contribution, CHARON, addressed class-incremental learning for skeleton-based action recognition. By exploiting the structured and temporally redundant nature of skeletal data, CHARON demonstrated that memory efficiency can be significantly improved through temporal sub-sampling and interpolation-based reconstruction. In addition, masked modeling was integrated directly into the continual learning process, acting as an auxiliary regularizer that stabilizes representations under replay. This chapter showed that, when input structure is explicitly leveraged, rehearsal-based continual learning can achieve strong performance even under tight memory and computational constraints.

The second contribution, CGIL, shifted the focus from input-space structure to representation-level adaptation in large pretrained vision–language models. Rather than modifying the backbone architecture, CGIL operated entirely in embedding space, combining prompt learning with generative replay to preserve zero-shot capabilities while enabling incremental adaptation. This chapter highlighted a complementary perspective on continual learning: when powerful pretrained representations are available, forgetting can be mitigated by aligning and replaying representations, without altering low-level feature extractors.

Taken together, the chapters in this part illustrate two complementary strategies for continual learning. CHARON shows how structured inputs can be compressed, reconstructed, and replayed efficiently, while CGIL demonstrates how adaptation can be confined to a lightweight representational interface built on top of frozen pretrained models. In both cases, continual learning is achieved through explicit rehearsal and sequential optimization across tasks.

While effective, these approaches still rely on task-by-task training and explicit replay mechanisms. As the number of tasks grows, this paradigm raises questions about scalability, efficiency, and long-term knowledge consolidation. The next part of the dissertation moves beyond sequential retraining by exploring an alternative direction: model merging. Instead of learning incrementally from data streams, Part III investigates how independently trained models, task vectors, or parameter-efficient updates can be composed and merged to transfer knowledge across tasks and domains. In particular, this perspective enables extensions of the ideas introduced in CGIL, as exemplified by MoDER, where generative replay and representation alignment are combined with model composition techniques to obtain continual adaptation without explicit sequential training.

Part III

Learning through Models

9

Background: Model Composition and Knowledge Transfer

The previous parts of this dissertation addressed continual learning primarily through sequential adaptation, in which models are updated over time using rehearsal, regularization, or representation alignment. While effective, these approaches rely on explicit retraining across tasks and assume continued access to data streams or replay mechanisms. As the number of tasks grows, such assumptions raise concerns regarding scalability, efficiency, and long-term knowledge consolidation.

This part explores a complementary paradigm: *model composition*. Rather than learning new tasks through additional sequential optimization, model composition aims to reuse, combine, or transport knowledge already encoded in trained models or parameter-efficient updates. In this setting, adaptation emerges from the structured combination of existing components, enabling knowledge transfer without direct retraining on the target task data.

Model composition provides a unifying framework for a diverse set of problems, ranging from domain adaptation and zero-shot generalization to cross-task transfer and modular learning. Across these settings, the central question is how to represent task-specific knowledge in a form that can be reliably reused and combined. In this

part, adaptation is no longer framed as an optimization process, but as an algebraic operation over learned task representations.

9.1 An informal overview

The easiest way to read this part is to think of learning as *reusing pieces of models* rather than retraining a system from scratch. Each chapter instantiates this idea in a different way, but the common intuition is that useful knowledge can be stored in a modular form and later recombined to solve a new problem.

PASTA (Chapter 10) offers the most concrete example: a tracker may behave differently in daylight, rain, or crowded scenes, and instead of retraining the full model for every condition, one can compose condition-specific modules to obtain the desired behavior. MoDER pushes the same idea toward zero-shot class synthesis (Chapter 11), in which knowledge learned on previous tasks is recombined to recognize classes that were never observed jointly. In both cases, adaptation comes from composition rather than further end-to-end optimization.

Core Space and GradFix (Chapters 12 and 13) address two additional questions that arise once this compositional view is adopted. If multiple experts are available, how can they be merged reliably without destructive interference? And if a useful update was learned on one pretrained model, how can it be transported to another one? These examples provide an informal map of the part before the formal treatment below, in which task vectors, low-rank updates, and shared representation spaces make these operations precise.

9.2 From continual adaptation to compositional learning

Traditional continual learning focuses on preserving performance on past tasks while acquiring new ones through incremental optimization. In contrast, compositional learning assumes that knowledge can be decomposed into reusable units that remain stable once learned. New capabilities are obtained by selecting, combining, or transforming these units, rather than by modifying the underlying model parameters through further gradient-based training.

This shift is motivated by two observations. First, modern models often encode rich

and transferable representations that generalize across tasks, domains, and modalities. Second, parameter-efficient adaptation methods provide compact representations of task-specific knowledge that can be manipulated independently of the backbone model.

Within this perspective, tasks are no longer treated as sequential optimization problems, but as compositional objects whose solutions can be assembled from existing components. This enables adaptation in settings in which task data may be unavailable, impractical to store, or expensive to retrain on.

9.3 Representing task knowledge

A key requirement for model composition is an explicit representation of task-specific knowledge. Several forms of representation are considered throughout this part, with a primary focus on parameter-space representations.

One approach represents tasks through *parameter deltas*, such as task vectors or low-rank adaptation modules, which capture how a pretrained model must change to solve a given task. These representations can be added, merged, or transported across models, enabling the reuse of learned behaviors.

Another complementary approach represents tasks implicitly through learned prototypes or embeddings, which encode task-relevant information in feature space rather than parameter space. Composition in this case operates by combining representations to infer new concepts or conditions without retraining the model.

Finally, task knowledge may be distributed across multiple specialized models, each trained under different conditions. Composition then consists of selecting and combining these models or their outputs to adapt to unseen scenarios.

Across all cases, the challenge lies in ensuring that composed knowledge remains coherent, stable, and faithful to the original tasks.

9.4 Composition under external conditions and unseen tasks

Model composition is particularly relevant when adaptation must occur under external conditions that were not explicitly observed during training. These conditions may correspond to environmental factors, domain shifts, or semantic variations that alter the model's behavior without changing the underlying task. Examples include adapting

a tracker to changing environmental conditions or synthesizing classifiers for unseen semantic categories by recombining known concepts.

In such settings, composition enables conditional adaptation by activating or combining task components associated with different conditions. This allows models to generalize to unseen configurations by leveraging previously learned knowledge, rather than requiring new training data for every possible variation.

Similarly, composition enables generalization to unseen classes or tasks by combining known components in novel ways. Instead of learning new concepts from scratch, the model infers them through structured combinations of existing representations.

9.5 Efficient and reliable model merging

As compositional approaches increasingly rely on parameter-efficient updates, an important challenge is how to merge multiple adaptations reliably. Naive combination strategies may lead to interference, degradation, or instability, particularly when merging a large number of updates.

This motivates the need for structured representation spaces and principled criteria for merging task representations without leaving the parameter-efficient regime. Such spaces must preserve task-specific information while enabling efficient combination and minimizing destructive interference. Reliability and accuracy of merging are essential, especially when composition replaces retraining entirely.

9.6 Compositional learning and task representations

The works presented in this part are grounded in a view of learning in which adaptation is represented explicitly in parameter space and manipulated through algebraic operations. Rather than training a model end-to-end for each new task, knowledge is encoded as structured modifications of a shared pretrained model and reused through composition, interpolation, and transport.

Let $f(\cdot;\theta)$ denote a neural network with parameters $\theta \in \mathbb{R}^d$, initialized from a pretrained solution θ_0. Adapting the model to a task $\mathcal{T}_i$ produces a new parameter vector θ_i, obtained either through full fine-tuning or parameter-efficient updates. Following prior work on task arithmetic and model composition [109, 2, 177], we define the *task vector* associated with $\mathcal{T}_i$ as

$$\tau_i \triangleq \theta_i - \theta_0. \tag{9.1}$$

The task vector τ_i captures how the pretrained model must be modified to solve task $\mathcal{T}_i$. Under the assumption that adaptations remain within the same basin of the loss landscape, task-specific solutions can be approximately recovered through linear superposition:

$$\theta_0 + \tau_i \approx \theta_i. \tag{9.2}$$

This representation enables a form of *task arithmetic*, in which multiple task vectors can be combined through linear operations to construct new behaviors without retraining:

$$\theta_{\text{comp}} = \theta_0 + \sum_i \alpha_i \tau_i, \tag{9.3}$$

with $\alpha_i \in \mathbb{R}$ controlling the contribution of each task. Such linear combinations have been shown to support skill composition, attribute manipulation, and domain adaptation in large models [109, 298].

A key empirical property enabling these operations is *linear mode connectivity* [82, 65]. Given two solutions θ_a and θ_b obtained from the same initialization, the linear interpolation

$$\theta(\lambda) = (1-\lambda)\theta_a + \lambda\theta_b, \quad \lambda \in [0,1], \tag{9.4}$$

often exhibits low loss throughout the entire path. Rewriting this interpolation in terms of task vectors yields

$$\theta(\lambda) = \theta_0 + (1-\lambda)\tau_a + \lambda\tau_b, \tag{9.5}$$

which highlights interpolation as a special case of task composition. This observation underlies conditional adaptation mechanisms, in which external factors (*e.g.*, environment conditions or domains) modulate the contribution of different task vectors.

At the same time, linear mode connectivity should be viewed as an empirical approximation rather than a universal guarantee. It is most reliable when solutions remain in a nearby basin and admit a compatible parameterization, and it may weaken under large domain shifts, unresolved permutation symmetries, or regimes in which the low-loss connection is curved rather than well approximated by a single straight segment. The chapters in this part build on this linear perspective, while also introducing additional structure or local geometric information when linear composition alone is insufficient.

In practice, storing full task vectors $\tau_i \in \mathbb{R}^d$ is often infeasible. Instead, task knowledge is commonly represented through parameter-efficient structures, such as low-rank updates or adapters, which constrain task vectors to low-dimensional

subspaces [106, 127]. These representations preserve the linear structure required for task arithmetic while enabling efficient storage and manipulation. In this view, parameter-efficient modules such as low-rank adapters are treated as structured task vectors.

However, naively combining multiple task vectors can lead to interference when updates overlap in parameter space. This motivates the development of structured composition and merging strategies that account for alignment, redundancy, and conflict between task representations [177, 2, 298]. Designing such strategies is central to scaling composition beyond a small number of tasks.

An additional challenge arises when task vectors are learned relative to different pretrained models. Given two initializations θ_0 and $\tilde{\theta}_0$, a task vector τ_i learned from θ_0 may not transfer directly to $\tilde{\theta}_0$ due to differences in representation geometry. Transporting task knowledge across models requires constructing a transformed vector $\tilde{\tau}_i$ such that

$$\theta_0 + \tau_i \quad \text{and} \quad \tilde{\theta}_0 + \tilde{\tau}_i \tag{9.6}$$

exhibit equivalent behavior. Addressing this problem enables cross-model reuse of task knowledge and decouples task learning from the choice of pretrained backbone.

Overall, this part adopts a unified perspective in which learning is reframed as the construction, composition, and transport of task representations. Within this framework, adaptation can be achieved without retraining, conditioned on external signals, or transferred across models. The following chapters instantiate this perspective across different domains and problem settings, ranging from condition-aware tracking and zero-shot classification to efficient merging and cross-model task transport.

10

Is Multiple Object Tracking a Matter of Specialization?

10.1 Scenario-specialized tracking with parameter-efficient modules

Video surveillance systems rely on Multiple Object Tracking (MOT) to localize and maintain identities over time, often under variations in viewpoint, illumination, crowd density, and camera motion. Modern MOT methods are commonly grouped into tracking-by-detection (TbD) [18, 275, 312, 172, 232, 207, 174] and end-to-end query-based trackers [301, 313, 295, 79]. As discussed in Chapter 4, TbD approaches remain strong and reliable, but their association stage is non-differentiable, making joint optimization of detection and tracking difficult. Query-based trackers address this limitation by learning detection and association within a unified transformer architecture.

Despite their appeal, end-to-end transformer trackers are costly to train and sens-

Publication. Gianluca Mancusi, Mattia Bernardi, **Aniello Panariello**, *et al. Is Multiple Object Tracking a Matter of Specialization?*. NeurIPS, 2024 [175].

Candidate contribution. Research question, experimental design, and writing.

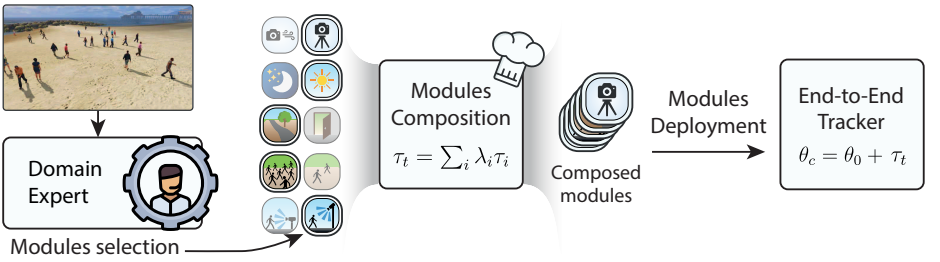


Figure 10.1: Given a scene, we select the modules corresponding to its attributes, such as lighting and indoor/outdoor. These modules are composed and then deployed, yielding a specialized model.

itive to changes in scenario conditions [208, 294]. In particular, models trained on a specific mixture of scenes can overfit to spurious regularities (*e.g.*, camera placement or lighting), and performance may degrade when deployed in environments whose attributes differ from those seen during training. Collecting sufficient data for each operational setting is often impractical, making efficient adaptation necessary.

This chapter introduces Parameter-efficient Scenario-specific Tracking Architecture (PASTA), a modular framework that adapts query-based trackers through lightweight parameter-efficient modules [106, 202]. Instead of fine-tuning a single model on all conditions, PASTA learns small experts associated with discrete scene attributes (*e.g.*, indoor/outdoor, lighting, camera motion) and composes the corresponding experts at deployment time to obtain a tracker specialized to the current scenario (Fig. 10.1). Modules are trained independently, which reduces negative interference across incompatible conditions [271, 272, 208, 294], while composition enables reuse of learned expertise under new attribute combinations. We evaluate PASTA on MOTSynth [72] and on real-world benchmarks (MOT17 [57] and PersonPath22 [242]) under both in-domain and zero-shot transfer settings. Results show that scenario-specialized modules improve tracking performance on the source domain and provide consistent gains under synth-to-real and real-to-real shifts, without requiring test-time optimization. We summarize the main contributions as follows:

- We introduce PASTA, a modular framework for query-based MOT that adapts a pretrained tracker through parameter-efficient, attribute-specific experts.
- We show that independent expert training improves robustness under scenario variation and reduces interference relative to monolithic fine-tuning.
- We validate the approach on synthetic and real benchmarks, including zero-shot transfer, demonstrating practical gains under domain shifts.

10.2 Related work

Multiple Object Tracking. We refer the reader to the background provided in Part I for a detailed discussion of Multiple Object Tracking paradigms and benchmarks. Here, we briefly recall only the aspects most relevant to this chapter.

Modern MOT methods are commonly divided into tracking-by-detection (TbD) approaches [18, 275, 312] and end-to-end transformer-based trackers [181, 301, 313]. While the latter unify detection and association through attention mechanisms, they are particularly sensitive to data scarcity and domain shifts [301, 208], often overfitting to the scenarios seen during training. PASTA operates within this second family and specifically targets robustness under scenario variation without retraining.

Modular Deep Learning and parameter-efficient adaptation. Modular Deep Learning (MDL) [203] aims at decomposing adaptation into lightweight, reusable components attached to a shared backbone. This paradigm has gained traction as model sizes increase and full fine-tuning becomes impractical. Parameter-efficient techniques such as LoRA [106], $(IA)^3$ [159], and scale-and-shift layers [149] enable learning task- or domain-specific updates while freezing the base model.

A key aspect of MDL is that multiple modules can be trained independently and later combined at inference time through routing and aggregation mechanisms. When aggregation is performed directly in parameter space, the process can be interpreted as model merging [290], yielding a single set of weights and preserving inference efficiency. PASTA builds on this principle by associating modules with interpretable scene attributes and composing them to specialize a tracker for the current scenario.

Domain adaptation and robustness in MOT. Domain adaptation for MOT has primarily focused on tracking-by-detection pipelines. Methods such as GHOST [232] and DARTH [231] adapt appearance models or detectors using test-time adaptation or distillation. While effective, these approaches often require multiple passes over video sequences or additional optimization at deployment time.

In contrast, PASTA targets end-to-end tracking-by-attention models and performs adaptation entirely through pretrained, attribute-specific modules. Once trained, specialization is achieved by composition alone, without test-time learning. Unlike open-vocabulary tracking approaches based on CLIP [143, 256], our method does not aim to generalize across object categories, but rather across *scenarios*, focusing on robustness to domain shifts in surveillance settings.

10.3 Preliminaries

Query-based Multiple Object Tracking. The underlying backbone of our tracker follows the structure of [301]. In a nutshell, such a query-based model forces each query to recall the same instance across different frames. Specifically, we leverage an end-to-end trainable tracker built upon the Deformable DETR [31] framework conditioned by the image features extracted with a convolutional backbone (*i.e.*, ResNet [95]). Following [313], we further condition the DETR decoder with a set of detections from an external detector network and a shared learnable query.

At time $t = 0$, new proposals are generated from the objects detected in the scene. These proposals are then updated through self-attention and interact with image features via the deformable attention layer. The final prediction output is the sum of the initial anchors and the predicted offsets. For subsequent frames ($t > 0$), track queries generated from the previous frame are concatenated with learnable proposal queries of the current frame. Moreover, previous predictions are integrated with current proposals to establish new anchors for the incoming frame. Additional architectural details can be found in [313]. Notably, the flexibility of this architecture allows seamless integration of modular adaptation techniques.

10.4 Method

This section presents PASTA (Fig. 10.2), a modular approach to Multiple Object Tracking that leverages PEFT modules to enable attribute-specific specialization and reuse. This approach allows for the dynamic configuration of an end-to-end tracker by selecting the appropriate modules for each input scene, fully leveraging heterogeneous pretraining while avoiding negative transfer.

Attribute-based modularity. We devise a set of learnable modules to fine-tune each layer of our query-based tracker. Each module is related to an **attribute**: as shown in Fig. 10.3, we define $N = 5$ attributes, namely *lighting*, *viewpoint*, *occupancy*, *location*, and *camera motion*, and provide a tailored module for each discrete value these attributes take. For instance, the *location* attribute has indoor and outdoor modules. At inference time, prior knowledge about the input scene is used to determine the appropriate value for each attribute, which in turn selects the corresponding modules from the "inventory", denoted as M.

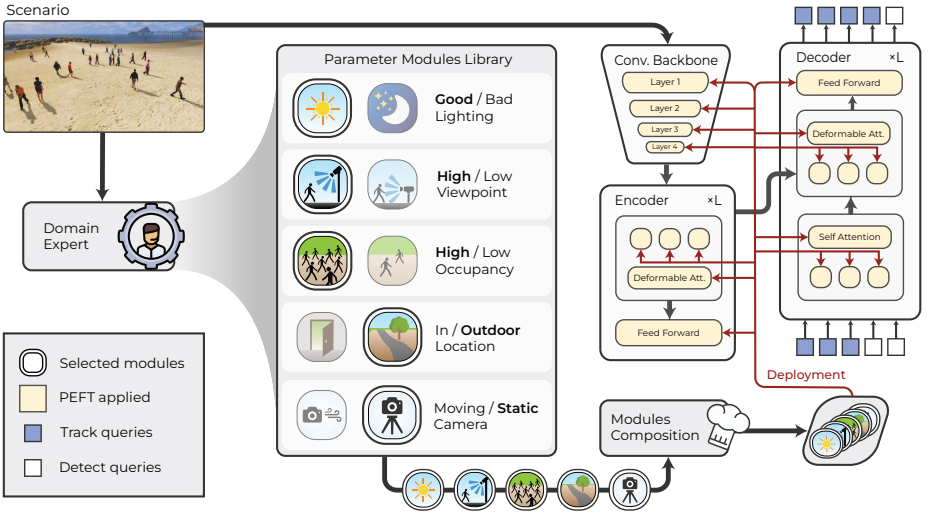


Figure 10.2: Overview of our modular architecture. A domain expert selects PEFT modules based on sequence attributes such as lighting and camera movement. These selected modules are then composed and applied to each model layer, adapting the backbone and encoder-decoder architecture.

Since the base model [313] relies on heterogeneous layers – namely, convolutional (*e.g.*, ResNet) and attention-based blocks (*e.g.*, Deformable DETR) – we employ two different strategies to fine-tune the modules. Specifically, after each convolutional layer of the ResNet backbone, we apply a strategy that learns channel-wise scale-and-shift parameters; for each layer of Deformable DETR, instead, we employ LoRA-based fine-tuning [106] at each linear layer. In formal terms, considering each convolutional layer of the ResNet backbone, we deploy $|M|$ pairs $\{\gamma_m, \beta_m\}_{m=1}^{|M|}$ of learnable vectors $\gamma, \beta \in \mathbb{R}^C$, where C is the number of output channels. For each linear layer l of the encoder-decoder structure underlying Deformable DETR, we devise $|M|$ pairs $\{A_m, B_m\}_{m=1}^{|M|}$ of learnable LoRA matrices.

During training, we start with the pretrained weights and integrate all the modules while keeping the original parameters frozen. To reduce negative interference, each module is optimized independently by sampling one attribute at a time and updating only the corresponding module at each training iteration. By the end of the training process, we obtain a set of specialized parameters (*experts*), which can be seamlessly merged during inference to improve overall tracking performance.

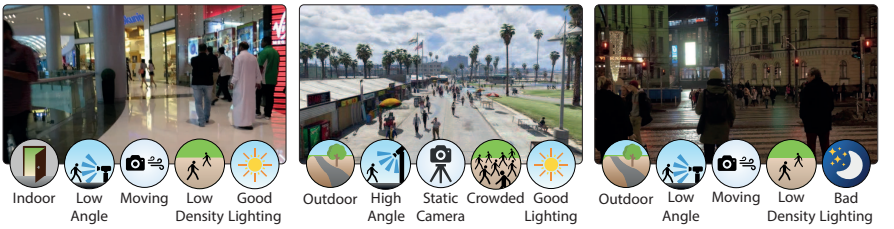


Figure 10.3: Examples of surveillance scenes and their corresponding attributes used by PASTA.

Routing via a Domain Expert. During inference, two essential steps are required to exploit the learned modules: *routing* and *aggregation*. With multiple modules available from the inventory M, a routing strategy is required to determine which ones should be active. To make this selection, we draw on what is known in the literature as *expert knowledge* [281, 203] (or "**Domain Expert**" in Fig. 10.2). In real-world applications such as video analytics, the expertise guiding the selection can come from a video-surveillance operator or human analyst, who configures the appropriate modules to reflect domain- and scene-specific settings, such as camera perspective, lighting conditions, and other critical details. This approach allows users to optimize the tracking model for their specific contexts without extensive retraining. Additionally, the modular nature of the system enables easy integration of new modules to address emerging attributes or scenarios.

Module composition. In the final step, we aggregate the selected modules ("Modules Composition" in Fig. 10.2) and incorporate the result into the pretrained tracker to create an expert model. Since these modules have been obtained by fine-tuning from $\boldsymbol{\theta}_0$, each module $\boldsymbol{\theta}^\star$ corresponds to a specific displacement $\tau^\star = \boldsymbol{\theta}^\star - \boldsymbol{\theta}_0$ in parameter space relative to the initial pretraining parameters $\boldsymbol{\theta}_0$. This displacement is known as the *task vector* [109]. The final composed model $f(\cdot;\boldsymbol{\theta}_c)$ is defined as:

$$f(\cdot;\boldsymbol{\theta}_c) \quad \text{where} \quad \boldsymbol{\theta}_c = \boldsymbol{\theta}_0 + \textstyle\sum_{i=1}^{N} \lambda_i \tau_i, \quad \sum_i \lambda_i = 1 \text{ and } \tau_i \in M. \tag{10.1}$$

When $\lambda_i = \frac{1}{N}$, the formula is simply the average of the task vectors corresponding to each attribute. We employ this straightforward strategy for λ_i, giving equal weight to all attributes. However, for the task vector τ_i associated with the i-th attribute, we adopt a more refined strategy. If there are no domain shifts during inference (*i.e.*, both training and testing occur on the same dataset, such as MOTSynth), the task vector τ_i

is set to the displacement $\tau^\star$ produced by the expert module selected by the Domain Expert. In contrast, when domain shifts are present (*e.g.*, training on MOTSynth and testing on MOT17), we adopt a soft strategy that considers *all* the modules in the inventory associated with the relevant attribute. In doing so, we follow the insights from [306], where the authors showed that scenarios with shifting tasks benefit from richer representations than those derived from a single optimization episode.

Specifically, given the i-th attribute, let $R(i)$ be the set of its modules. Each attribute admits multiple discrete values (*e.g.*, $R(\text{occupancy}) = \{\text{“low”}, \text{“medium”}, \text{“high”}\}$), and different attributes may have different cardinalities (*e.g.*, $|R(\text{occupancy})| = 3$ and $|R(\text{lighting})| = 2$). Building on this, we employ soft routing to create the corresponding task vector, assigning most of the weight, *e.g.* $\rho = 0.80$, to the module selected by the Domain Expert. The remaining modules are weighted by $(1-\rho)/(|R(i)|-1)$, ensuring that the coefficients sum to 1. For example, for the layers fine-tuned with LoRA, the corresponding task vector is computed as:

$$\tau_i = \textstyle\sum_{m \in R(i)} \bar{\lambda}_m B_m A_m, \quad \text{where} \quad \bar{\lambda}_m = \begin{cases} \rho & \text{if } m \text{ is selected,} \\ \frac{1-\rho}{|R(i)|-1} & \text{otherwise.} \end{cases} \tag{10.2}$$

Note that when $\rho = 1$, the soft strategy becomes hard, meaning that only the module selected by Domain Expert is utilized. By applying the formula above to all attributes, we obtain N task vectors, which we aggregate following Eq. 10.1.

Similarly, we apply channel-wise scale and shift [149] operations to adapt each backbone layer. Formally, given the output F of a convolutional layer, the i-th module applies a scale & shift operation to obtain the edited $\hat{F}_i$, such that $\hat{F}_i = \gamma_i \odot F + \beta_i$ with $\odot$ denoting the Hadamard product. At inference time, we combine the output of different scale & shift modules by noting that

$$\hat{F} = \textstyle\sum_{i=1}^{N} \lambda_i(\gamma_i \odot F + \beta_i) = \sum_{i=1}^{N} \lambda_i(\gamma_i \odot F) + \lambda_i\beta_i = (\sum_{i=1}^{N} \lambda_i\gamma_i) \odot F + \sum_{i=1}^{N} \lambda_i\beta_i, \tag{10.3}$$

which means that parametrizing the scale & shift layer with a simple weighted average effectively results in averaging the outputs of the corresponding individual layers. The formula above applies to the in-domain setting but can be easily generalized to the soft routing scheme outlined by Eq. 10.2. Eventually, as discussed in [149], the scale & shift layer can be absorbed into the previous projection layer, thus ensuring that the inference process incurs no additional computational costs.

For convolutional layers adapted via scale & shift, the weighted module composition can be absorbed into the convolution parameters, yielding an explicit task vector in parameter space. Let a convolution produce $F = W_0 * h + b_0$, and let the composed module apply $\hat{F} = \gamma^\star \odot F + \beta^\star$ with $\gamma^\star = \sum_{m \in R(i)} \bar{\lambda}_m \gamma_m$ and $\beta^\star = \sum_{m \in R(i)} \bar{\lambda}_m \beta_m$ (hard routing is recovered when $\bar{\lambda}$ is one-hot). By re-parameterization, $\hat{F} = W^\star * h + b^\star$ where $W^\star = (\gamma^\star \odot W_0)$ and $b^\star = \gamma^\star \odot b_0 + \beta^\star$, which allows defining the corresponding task-vector increments as $\tau_W = W^\star - W_0$ and $\tau_b = b^\star - b_0$; therefore, scale & shift modules admit the same task-vector view used for LoRA modules and can be merged without inference-time overhead.

10.5 Experiments

Datasets. We evaluate PASTA on a combination of synthetic and real-world pedestrian tracking benchmarks. The **MOTSynth** [72] and **MOT17** [57] datasets are described in detail in Chapter 2 and are reused here under the same protocols and preprocessing settings. Additionally, we evaluate on **PersonPath22** [242], a large-scale real-world dataset for long-term pedestrian tracking, characterized by extended temporal horizons, severe occlusions, and crowded scenes. It consists of 236 videos, split into 138 training and 98 test sequences. Compared to MOT17, PersonPath22 features significantly longer trajectories and more challenging identity preservation conditions, making it particularly suitable for evaluating cross-domain and zero-shot generalization.

Experimental setting. PASTA is evaluated in both **in-domain** and **out-of-domain** scenarios. For the in-domain evaluation, we train and test PASTA on the MOTSynth synthetic dataset (Sec. 10.5.1) using expert modules in a domain-specific context. As a baseline, we train MOTRv2 [313] on MOTSynth without using modules, referring to this model as MOTRv2-MS. For the out-of-domain evaluation, we conduct a synth-to-real zero-shot experiment on MOT17 and PersonPath22 (Sec. 10.5.2). Starting from training on MOTSynth, we test PASTA on these datasets without additional training, showcasing its ability to generalize under distribution shift. Finally, we present ablation studies in Sec. 10.6 to take a closer look at the effectiveness of our method.

Competing trackers and metrics. We report the performance of notable methods, including strong tracking-by-detection baselines such as ByteTrack [312] and OC-Sort [30]. We also include evaluations of query-based trackers, such as TrackFormer [181] and MOTRv2 [313] (see MOTRv2-MS). To compare their performance, we

employ five metrics, ordered from detection to association, as recommended by [231]. These metrics are DetA [170], MOTA [15], HOTA [170], IDF1 [223], and AssA [170]. For the PersonPath22 dataset, we use their official metrics, MOTA and IDF1, supplemented by FP (false positives), FN (false negatives), and IDSW (identity switches).

Implementation details. We initialize our models using the pretrained weights from DanceTrack [249], as provided by the authors of [313]. We employ YOLOX [83] as the auxiliary detector, exploiting weights from ByteTrack [312]. To provide a shared initialization for both PASTA and MOTRv2-MS training, we train a bootstrap model starting from the DanceTrack initialization for 28k iterations on the MOTSynth training set. This bootstrap stage uses half of the original training sequences from MOTSynth to align our model with the scenarios represented in the dataset. The learning rates are set to 5×10^{-5} for the transformer and 1×10^{-6} for the visual backbone.

In the second phase, we deploy the PEFT modules to fine-tune the bootstrap model. By excluding half of the sequences during the bootstrap stage, we leave room for the modules to learn complementary features. To ensure a fair comparison, we train each module for a similar number of iterations as MOTRv2-MS, with approximately 17k iterations. Regarding the encoder-decoder model, we apply our modularization strategy to every linear layer except those with output dimension smaller than 128. For the LoRA hyperparameters, we use $r = 16$, a weight decay of 0.1, and a learning rate of 3×10^{-4}. The scale & shift layers employ a learning rate of 1×10^{-5} and a weight decay of 1×10^{-4}. Training is performed on a single RTX 4080 GPU with a batch size of 1 for both phases. Due to the small batch size, we accumulate gradients over four backward steps before performing an optimizer step. Each module is trained independently on the entire MOTSynth training set. With 12 modules, our model has approximately 15 million trainable parameters.

Attributes. We employ five key attributes to realize our modular architecture: lighting, camera viewpoint, people occupancy, location, and camera motion. For **lighting**, we specialize modules for *good* and *bad* lighting conditions. To do so, we threshold the brightness value V of the HSV representation at 70. The **viewpoint** attribute includes modules for *high*, *medium*, and *low* camera angles. We manually annotate this attribute as follows: (i) scenes where the camera is parallel to the ground at or below pedestrian head level are labeled as "low-level"; (ii) "high-level" viewpoints include vertical perspectives or scenes where the camera is positioned very high or far from people; and (iii) "medium-level" includes all other camera angles. For **occupancy**,

	$\|\Theta\|$	HOTA↑	IDF1↑	MOTA↑	DetA↑	AssA↑
SORT [18]	-	46.0	55.7	50.9	49.9	42.8
ByteTrack [312]	-	45.7	56.4	61.8	50.1	41.9
OCSort [30]	-	46.9	56.8	59.1	48.7	45.6
TrackFormer [181]	44M	41.3	49.9	47.7	44.4	40.6
MOTRv2-MS	42M	52.4	56.6	61.9	**56.4**	49.0
PASTA	15M	**53.0**	**57.6**	**62.0**	56.2	**50.4**

Table 10.1: Evaluation on MOTSynth test set. $|\Theta|$ is the number of trainable parameters.

we design modules that reflect the crowd density within the scene: *low* (up to 10 people), *medium* (10 to 40 people), and *high* (more than 40 people), based on the count of detections with a confidence score above 0.2. The **location** attribute differentiates between *indoor* and *outdoor* settings. Lastly, the **motion** attribute comprises modules for both *moving* and *static* cameras, enabling the model to adapt to different camera movement scenarios.

Computational costs. PEFT modules reduce training memory by decreasing the number of parameters updated by the optimizer (13GB for full fine-tuning versus 8.25GB for PASTA on MOTSynth), while inference incurs no measurable overhead beyond a negligible weight composition step for stationary attributes.

On the Domain Expert. In our experiments, we rely on a human Domain Expert to select the appropriate modules based on the attributes of the input scenes, *i.e.*, we assume access to the attribute values for each sequence in the test set. This is a reasonable assumption in real-world applications, in which such information is often available or can be easily inferred. For instance, the camera's mounting perspective and whether the scene is indoors or outdoors are typically known factors in fixed-camera scenarios. Additionally, automatic approaches can be envisioned to minimize human intervention further. For example, lighting conditions can be inferred by analyzing brightness levels, and a detector can count objects of interest in the scene to estimate crowd density.

10.5.1 Performance in the in-domain setting

To assess the impact of negative interference, we conduct several experiments on MOTSynth (see Tab. 10.1). Given the wide variety of scenarios in this synthetic dataset, the advantages of using specialized modules become evident. Indeed, integrating

	HOTA↑	IDF1↑	MOTA↑	DetA↑	AssA↑
fully-trained					
SORT [18]	64.3	73.1	70.9	63.3	66.1
OC-SORT [30]	66.4	77.8	74.5	64.1	69.1
TrackFormer [181]	–	74.4	71.3	–	–
ByteTrack [312]	67.9	79.3	76.6	66.6	69.7
MOTRv2 [313]	66.8	78.9	73.2	62.5	71.4
zero-shot					
TrackFormer [181]	51.0	63.9	58.7	51.8	61.2
MOTRv2-MS	62.6	73.0	67.6	60.3	65.5
PASTA ($\rho = 1$)	63.7	74.1	67.9	60.3	67.9
PASTA ($\rho = 0.8$)	**64.0**	**74.9**	**68.1**	**60.4**	**68.3**

Table 10.2: Zero-shot evaluation on MOT17. PASTA is evaluated in zero-shot by selecting the best attributes on the source dataset.

our modules yields an overall improvement w.r.t. the fully fine-tuned counterpart (MOTRv2-MS). Specifically, we observe gains on the association metrics (AssA, IDF1) as well as on HOTA and MOTA. These enhancements suggest that our approach reduces negative interference during training. By assigning each module a specific role tailored to particular scenario settings, we achieve improved training stability through a deterministic selection process guided by a domain expert.

10.5.2 Performance in the out-of-domain setting

By designing distinct modules for various input conditions, we can effectively select the appropriate modules to handle distribution shifts, such as transitions to a new domain. We assess the benefits of this ability using synthetic data for training and then evaluate on new, unseen datasets without any additional retraining (*zero-shot*). To do this, we start from the model trained on MOTSynth as described in Sec. 10.5.1 and evaluate it on MOT17 (Tab. 10.2) and PersonPath22 (Tab. 10.3). While these datasets share similarities in the attributes we employ, the source dataset is synthetic whereas the targets are real-world, resulting in a substantial shift.

The results reported in Tabs. 10.2 and 10.3 show an improvement over the baseline (*i.e.*, MOTRv2-MS), with + **1.4** in HOTA and + **1.9** in IDF1 on zero-shot MOT17, and + **1.7** in MOTA and + **0.7** in IDF1 on PersonPath22. Our approach demonstrates better gener-

	MOTA↑	IDF1↑	FP↓	FN↓	IDSW↓
fully-trained					
CenterTrack [318]	59.3	46.4	24 340	71 550	10 319
SiamMOT [241]	67.5	53.7	13 217	62 543	8942
FairMOT [311]	61.8	61.1	14 540	80 034	5095
IDFree [243]	68.6	63.1	9218	66 573	6148
TrackFormer [181]	69.7	57.1	23 138	47 303	8633
ByteTrack [312]	75.4	66.8	17 214	40 902	5931
zero-shot					
TrackFormer [181]	39.2	43.3	21 402	126 082	10023
MOTRv2-MS	48.3	53.1	28 483	**98 007**	7154
PASTA ($\rho = 1$)	49.7	53.7	18 211	105 611	6321
PASTA ($\rho = 0.8$)	**50.0**	**53.8**	**18 038**	105 454	**6037**

Table 10.3: Evaluation on PersonPath22 test set. PASTA is evaluated in zero-shot by selecting the best attributes on the source dataset.

alization capabilities, helping close the gap with fully trained methods while requiring less computation. These results indicate that modularity enhances performance on the source dataset and improves domain generalization, leading to a more reliable and versatile tracking approach. Furthermore, in addition to reporting the results with the standard module selection (considering only the modules present in the scene, $\rho = 1$), we also experiment with the weighted aggregation of all modules ($\rho = 0.8$), as detailed in Sec. 10.4. Interestingly, while the standard strategy already shows improvements, the weighted aggregation strategy yields even better performance. This suggests that richer representations, obtained by including multiple modules per attribute, are more effective for zero-shot scenarios than a single-module approach [306].

Evaluating zero-shot real-to-real transfer. In Tab. 10.4, we present an additional experiment to evaluate the performance of PASTA in a zero-shot setting, this time using a real-world dataset as the source rather than a synthetic one. For comparison, we train MOTRv2 on MOT17 and assess its performance on PersonPath22. Our approach achieves superior results with respect to the fine-tuned MOTRv2, confirming that modular specialization improves generalization to new real-world domains.

Source retention after adaptation. To quantify how modular fine-tuning affects source-domain retention, a further experiment evaluates the MOTSynth performance after adapting the tracker to a target real dataset. Starting from PASTA and MOTRv2-

	MOTA↑	IDF1↑	FP↓	FN↓	IDSW↓
fully-trained					
TrackFormer [181]	69.7	57.1	23 138	47 303	8633
ByteTrack [312]	75.4	66.8	17 214	40 902	5931
zero-shot					
MOTRv2-MS	43.9	51.5	8304	119 391	5342
PASTA	**46.1**	**54.6**	**7895**	**114 620**	**4702**

Table 10.4: Zero-shot evaluation of PASTA trained on MOT17 and tested on PersonPath22. PASTA is evaluated in zero-shot by selecting the best attributes on the source dataset.

MOTSynth	HOTA↑	IDF1↑	MOTA↑	DetA↑	AssA↑
Trained on MOTSynth (Tab. 10.1)					
MOTRv2-MS	52.4	56.6	61.9	**56.4**	49.0
PASTA	**53.0**	**57.6**	**62.0**	56.2	**50.4**
Subsequently trained on MOT17					
MOTRv2-MS	48.1 **(-4.3)**	56.3 **(-0.3)**	60.8 **(-1.1)**	50.7 **(-5.7)**	46.2 **(-2.8)**
PASTA	**49.8** **(-3.2)**	**57.4** **(-0.2)**	**61.8** **(-0.2)**	**52.3** **(-3.9)**	**48.0** **(-2.4)**

Table 10.5: Source-domain (MOTSynth) results before and after fine-tuning on target-domain (MOT17). We report the difference in performance in brackets.

MS trained on MOTSynth, both models are further fine-tuned on MOT17 and then re-evaluated on the MOTSynth test split; the results in Tab. 10.5 show that modular adaptation is less prone to degrading source performance than full fine-tuning, consistent with reduced interference across scenario-specific updates.

10.6 Ablation studies

In Tab. 10.6, we evaluate the effect of various routing and aggregation strategies in both the in-domain setting (MOTSynth, left side of Tab. 10.6) and the zero-shot setting (MOT17, right side of Tab. 10.6). In the in-domain scenario, the results show that averaging the modules selected by the Domain Expert, specifically using Mean avg. ($\rho = 1.0$), is the most effective strategy. Summation, as proposed by [307], yields worse results, plausibly due to altered weight magnitudes when combining multiple modules by addition. Another noteworthy approach is the *weighted avg.*, described in Sec. 10.4, which incorporates all modules, including those not selected.

MOTSynth (*val*)	HOTA↑	IDF1↑	MOTA↑
aggregation			
Sum (*only selected*)	0.65	0.44	-0.69
Weighted avg. ($\rho = 0.8$)	59.9	66.8	59.6
Mean avg. ($\rho = 1.0$)	**60.1**	**67.2**	**59.9**
selection			
Opposite modules	59.2	66.5	58.7
All modules	59.8	67.0	59.4
Domain Expert	**60.1**	**67.2**	**59.9**

MOT17 (*val*)	HOTA↑	IDF1↑	MOTA↑
aggregation			
Sum (only selected)	0.58	0.41	-0.03
Weighted avg. ($\rho = 0.8$)	**64.0**	**74.9**	**68.1**
Mean avg. ($\rho = 1.0$)	63.7	74.1	67.9
selection			
Opposite modules	62.9	73.9	67.1
All modules	63.1	**74.1**	67.7
Domain Expert	**63.7**	**74.1**	**67.9**

Table 10.6: Ablation study on different module aggregation and selection strategies. (**Left**) MOTSynth validation, (**Right**) Zero-shot on MOT17 validation.

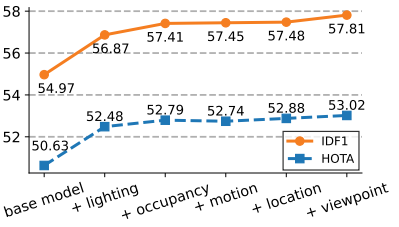


Figure 10.4: IDF1 and MOTA when adding new attributes on MOTSynth.

	HOTA↑	IDF1↑	MOTA↑
No modules	58.2	64.9	55.6
Opposite: only lighting	58.9	66.2	57.8
Opposite: only viewpoint	58.8	65.7	57.0
Opposite: only occupancy	58.6	66.4	59.2
Opposite: only location	58.9	66.0	58.4
Opposite: only camera	58.5	65.6	58.4
Average opposite	58.7	66.0	58.2
Correct modules	**60.1**	**67.2**	**59.9**

Table 10.7: Opposite modules selection.

While using only the *selected modules* is the optimal strategy in the in-domain scenario, for the zero-shot case (MOT17), incorporating knowledge from the non-selected modules, specifically using Weighted avg. ($\rho = 0.8$), enhances tracking performance. This pattern is also consistent on the PersonPath22 dataset.

Module selection. Should we select only the modules representing the current scenario, as determined by the Domain Expert approach, or would performance improve by incorporating all available modules? In Tab. 10.6, we investigate this matter by comparing these two approaches. To provide a more comprehensive perspective, we also evaluate a strategy that, in sharp contrast to the Domain Expert, selects the *opposite modules* (*e.g.*, selecting the outdoor and poor lighting modules when presented with an indoor, well-lit scene). The lowest performance is observed when using opposite modules, indicating that using the proper modules provides valuable information about the current scene. Interestingly, the model still performs relatively well despite using opposite attributes, likely due to contributions from other modules whose general knowledge of the domain sustains overall performance. This suggests that modules can assist one another in solving tasks. Moreover, reduced negative interference – achieved

Fine-tuning applies on	HOTA↑	IDF1↑	MOTA↑	DetA↑	AssA↑
none	52.4	56.6	61.9	**56.4**	49.0
all except the decoder	51.5	56.0	58.9	53.6	49.8
all except the encoder	52.4	56.9	61.2	55.7	49.7
all except the backbone	52.5	57.0	61.5	55.6	49.9
PASTA (all)	**53.0**	**57.6**	**62.0**	56.2	**50.4**

Table 10.8: Performance comparison of our approach when modules are not applied or are applied only to specific parts of the architecture (*i.e.*, decoder, encoder, visual backbone).

PersonPath22	MOTA↑	IDF1↑	FP↓	FN↓	IDSW↓
Sum (only selected)	0.91	0.64	–	–	–
Avg. (only selected)	49.6	53.6	18 211	105 611	6321
Weighted avg. (all)	**50.0**	**53.8**	**17 786**	**105 454**	**6037**

Table 10.9: Ablation study on different module aggregation strategies on PersonPath22 test set in zero-shot.

by training each module separately – prevents the modules from relying on each other and allows them to make unique contributions independently.

Furthermore, in Fig. 10.4, we illustrate how the incremental addition of specialized modules improves IDF1 and HOTA metrics, showing that greater specialization gradually enhances overall performance. For a more detailed analysis, in Tab. 10.7, we select the opposite modules instead of the correct one for each attribute. Although the metrics are further reduced, the model still performs well due to its robust pretraining, as indicated by the *no modules* baseline shown in the table.

Block-wise analysis. In our approach, attribute-related modules are applied to the entire network. However, users may opt to edit only specific parts of the architecture, thereby identifying which components are most critical. In Tab. 10.8, we conduct an ablation study by excluding our modules from different components of the architecture. The results indicate that not applying task vectors to the decoder significantly degrades both detection and association metrics. We believe this degradation reflects the crucial role of the decoder, which must gather information from detection, tracking, and proposal queries while simultaneously integrating visual information from the encoder. Consequently, not adapting the decoder prevents the architecture from effectively leveraging both queries and visual cues. The encoder also contributes substantially, though to a lesser extent than the decoder, as it primarily refines and contextualizes visual features from the backbone. Finally, the backbone shows the smallest contribution.

Weighted aggregation under domain shift. An additional ablation on the PersonPath22 test split further supports the benefit of incorporating non-selected modules under domain shifts. As reported in Tab. 10.9, assigning $\rho = 0.8$ to the selected modules and distributing the remaining weight across the others improves performance over unweighted averaging, reinforcing the observation that richer attribute mixtures can be advantageous in zero-shot transfer.

10.7 Conclusions

This chapter introduced PASTA, a modular framework that improves domain robustness in query-based Multiple Object Tracking through attribute-specific parameter-efficient modules. The approach trains experts independently to reduce negative interference and composes them in parameter space to obtain a specialized tracker at inference time. The experimental evaluation shows that modular composition improves tracking performance in-domain and strengthens zero-shot transfer under synth-to-real and real-to-real shifts. The resulting framework also supports practical deployment, since scenario attributes can be selected by a domain expert or inferred automatically.

Beyond tracking, PASTA also provides a first concrete instance of a broader theme in this part of the dissertation: acquiring new capabilities by *composing* lightweight adaptations on top of a shared pretrained backbone. In PASTA, composition is guided by scene attributes and performed in parameter space to obtain a scenario-specialized tracker without additional optimization at deployment time. The next chapters generalize this idea from scenario-conditioned specialization to more challenging forms of knowledge reuse: *MoDER* composes modular prototypes to recognize novel classes, *Core Space* formalizes how LoRA-style modules can be merged reliably within a shared representation space, and *GradFix* studies how to transport a learned update between different pretrained models. Together, they shift the focus from *adapting a single model sequentially* to *building new behavior by combining and transferring learned increments.*

11

Modular Embedding Recomposition for Incremental Learning

11.1 From preserving zero-shot to improving it

Pretrained Vision–Language Models (VLMs) such as CLIP [210] offer strong zero-shot recognition and a convenient interface for open-vocabulary classification. Yet, when CLIP is adapted incrementally, two failure modes appear simultaneously: (i) accuracy on previously learned classes deteriorates (*catastrophic forgetting*) [178], and (ii) zero-shot performance on future (unseen) tasks degrades as incremental updates overwrite alignment inherited from pretraining [314, 297]. This second aspect is particularly limiting in realistic deployments, where the set of future domains and classes is unknown and retraining is costly.

Part II introduced replay in embedding space as an effective way to stabilize incremental adaptation of CLIP, culminating in CGIL [77]. MoDER builds on this perspective and asks a different question: can the knowledge accumulated across tasks be *reused*

Publication. Aniello Panariello, *et al. Modular Embedding Recomposition for Incremental Learning.* BMVC, 2025 [195].

Candidate contribution. Idea, methodology, implementation, experiments, and writing.

compositionally to *improve* recognition of classes that have never been observed yet? **MoDular Embedding Recomposition** (**MoDER**) answers positively by turning incremental learning into a problem of *expert acquisition and expert composition*. Instead of maintaining a single evolving prompt (or relying on conservative hand-crafted templates for future classes), we incrementally learn lightweight *textual experts* that specialize the CLIP text encoder and store them in a **foundational hub**. At inference time, the hub enables two behaviors: (i) robust classification of *seen* classes using their dedicated experts, and (ii) *expert forging* for *unseen* classes by composing the most relevant experts in parameter space, producing textual prototypes on the fly.

MoDER is evaluated on both Class-Incremental Learning and Multi-Domain Task Incremental Learning [314, 297]. In addition to standard accuracy on seen classes (Final Average Accuracy), we quantify generalization to future tasks using Class Incremental Transfer [77]. Overall, MoDER complements the Part II narrative by moving from *retaining* CLIP's zero-shot capabilities to *actively reusing* incremental knowledge to strengthen them. We summarize our contributions as follows:

- We introduce MoDER, a modular framework for incremental adaptation of CLIP that stores class- or task-specialized textual experts in a foundational hub.
- We propose a training strategy (Textual Alignment) that supports replay in embedding space while encouraging experts to be linearly composable in parameter space.
- We introduce Mixture of Textual Experts (MoTE), a composition rule that forges textual prototypes for unseen classes by merging relevant experts from the hub, improving transfer without additional image-level training.

11.2 Related work

Prompt-based continual learning with CLIP. Prompting is a common strategy to adapt CLIP in Class-IL while keeping the encoders frozen. L2P [270], DualPrompt [269], and CODA-Prompt [245] maintain pools of prompts and select or compose them across tasks. AttriCLIP [266] adapts CLIP through learnable textual prompts, while CGIL [77], as seen in Chapter 8, performs replay directly in CLIP's embedding space to stabilize prompt learning. MoDER differs in its objective: rather than preserving a conservative zero-shot interface for unseen classes, it leverages incrementally learned *textual experts* and composes them to forge prototypes for unseen classes.

Zero-shot continual learning. ZSCL [314] formalizes the problem of preserving zero-shot performance under continual adaptation, using teacher-student constraints to retain pretraining knowledge. MoE-Adapters [297] introduce multiple experts to mitigate degradation across domains, but require task identity and routing at inference. MoDER instead stores experts in a hub and uses parameter-space composition to form class-level prototypes, supporting transfer to unseen classes without requiring a task router at test time.

Model compositionality and task vectors. Recent work has shown that fine-tuned models can often be combined through linear operations in parameter space, enabling editing and composition [109, 193, 175]. MoDER adopts this view for CLIP's text encoder: each expert is treated as a task vector, and unseen-class prototypes are obtained by merging a small set of relevant vectors.

11.3 Preliminaries

Setting and notation. We consider a sequence of tasks indexed by $t = 1, \dots, T$. In task t, the learner observes a dataset $\mathcal{D}_t = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N_t}$ with labels drawn from a disjoint class set $\mathcal{Y}_t$ (Class-IL) or a dataset-specific label space (MTIL). We denote by $\mathcal{Y}^{[1:t_c]} = \bigcup_{t=1}^{t_c} \mathcal{Y}_t$ the set of **seen** classes up to the current task t_c, and by $\mathcal{Y}^{[t_c+1:T]}$ the set of **unseen** classes belonging to future tasks.

CLIP interface. We adopt CLIP [210] with a frozen visual encoder $E^{\text{vis}}(\cdot)$ and a text encoder $E^{\text{txt}}(\cdot)$. Given an image x, the visual embedding is $z^{\text{vis}} = E^{\text{vis}}(x)$. Given a textual prompt p, the text embedding is $z^{\text{txt}} = E^{\text{txt}}(p; \theta)$, where θ denotes the parameters of the text encoder.

Experts as task vectors. MoDER adapts the text encoder through lightweight experts. For an expert associated with class (or task) i, we denote its parameter displacement by τ_i relative to the pretrained text-encoder weights θ_0. The adapted text encoder is parameterized as $\theta_0 + \tau_i$, and the corresponding class textual embedding is obtained by prompting the adapted encoder. In our implementation, τ_i is instantiated through parameter-efficient modules (*e.g.*, LoRA [106]), but MoDER only requires the task-vector view: experts are stored, retrieved, and composed as displacements τ in parameter space.

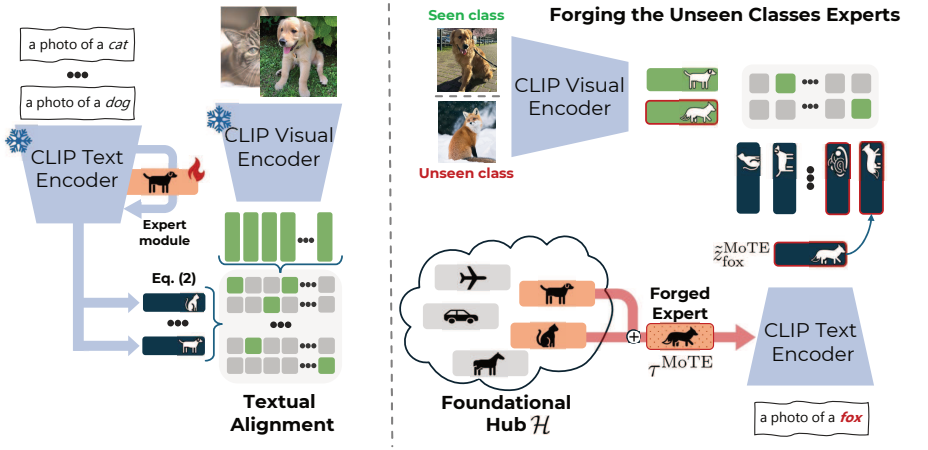


Figure 11.1: Overview of MoDER. The left side depicts the generative modeling and Textual Alignment (TA) phases. The right side represents the forging of embeddings for unseen classes.

11.4 MoDular Embedding Recomposition

The name of our method – **MoDular Embedding Recomposition** (**MoDER**) – builds on the concept of enhancing capabilities for unseen classes by recomposing fragments of knowledge accumulated across previous tasks. Our training framework consists of specialized experts in the form of lightweight PEFT modules, which fine-tune the CLIP text encoder. We denote by τ_i the task vector associated with class i, implemented as a low-rank LoRA update. We then introduce **Textual Alignment** (**TA**, see Sec. 11.4.1), namely a training strategy that promotes the composition of different learnable modules.

During inference, we deploy a dual strategy to compute textual embeddings. Specifically, for classes from the seen set $\mathcal{Y}^{[1:t_c]}$, we use the output of the respective LoRA experts. For the unseen set, we present an approach called **Mixture of Textual Experts** (**MoTE**, see Sec. 11.4.3) that leverages compositionality to create experts on the fly. Specifically, given a **foundational hub** $\mathcal{H}$, comprising all trained experts, we merge the K experts most relevant to the target unseen class.

11.4.1 Textual parameter-efficient specialized experts

Given all images x in the current dataset $\mathcal{D}_{t_c}$, we **freeze** the CLIP visual encoder and extract the relative visual embeddings z^{vis}. These embeddings are used to learn **aligned** textual prototypes – one per class in the current task – ensuring that image-to-text cosine similarity allows for accurate classification. To do this, while existing methods mainly rely on prompt-tuning [317, 316], we explore a different approach and devise a distinct LoRA [106] expert module τ_i for each class. The textual embedding for the class i fine-tuned through the expert is obtained as:

$$\tilde{z}_i^{\text{txt}} = E^{\text{txt}}\left(p_i; \theta_0 + \tau_i\right) \quad \text{s.t. } \tau_i = B_i A_i \ , \ i \in \mathcal{Y}^{[1:t_c]} \tag{11.1}$$

where $p_i =$ "`a photo of a [CLS]`" is a hand-crafted textual prompt for the i-th class within the seen set. Hence, an expert specializes the CLIP text encoder on textual prompts related to the same reference class. The displacement τ_i can be understood as a **task vector** [109], that is, the direction in parameter space along which the capabilities of the pretrained model rapidly improve for the target class i. In a sense, we aim to capture unique aspects of class i with the corresponding task vector τ_i.

Notably, to achieve specialization, each expert in our method is associated with its own set of weights. While this may raise concerns in memory-constrained settings, our formulation is flexible. It can be adapted to define task-specific experts (as shown in Tab. 11.2), requiring only one additional set of parameters per task. Furthermore, we can leverage highly efficient adaptation techniques, such as VeRA [127]. Results for both LoRA- and VeRA-based expert modules are reported in Tab. 11.3.

11.4.2 Textual alignment (TA)

While training the textual experts for the newly introduced task, we must consider two aspects: (i) the new textual prototypes should not interfere with the existing ones, which were trained for previous classes, in order to prevent *catastrophic forgetting* during prediction; (ii) the experts should be trained to be composable, such that their linear combination yields meaningful outputs. In the following, we discuss how our training strategy, called Textual Alignment (TA), achieves both objectives.

To avoid interference between experts and the resulting forgetting issues, we augment the training data for the current task by including samples from the past, akin to generative replay [77]. Namely, we train the experts not only on examples

from the current task's data but also on **synthetic visual embeddings** generated by lightweight generative models previously trained on past tasks. Specifically, at the onset of the current task, we train a lightweight **class-conditioned diffusion model** [102], denoted by $g_{t_c}(\cdot)$, on the features produced by the visual encoder. Notably, due to the low-dimensional nature of these features, the generator can be implemented as a lightweight Multi-Layer Perceptron (MLP), with negligible computational cost. In particular, each diffusion model $g_{t_c}(\cdot)$ consists of an eight-layer MLP with 256 hidden channels per layer and SELU [124] as the activation function.

Thanks to the generators trained on previous tasks (one per task), we can compensate for the absence of corresponding data in later stages and create balanced training sets for each expert. In practice, after learning the generator for the current task, we construct a synthetic dataset $\mathcal{D}_{\text{SYN}}$ by combining the features generated from all the diffusion models. We then train the experts using only data from the $\mathcal{D}_{\text{SYN}}$, as discussed in the following.

Loss function. To train the PEFT experts, we aim to align the text embeddings in Eq. 11.1 with the visual embeddings in the synthetic dataset $\mathcal{D}_{\text{SYN}}$. While a standard cross-entropy loss could be used for multi-modal alignment (as in [317, 210]), we treat each class independently by framing the problem as multiple binary classification problems. Specifically, we train the experts with the **sigmoid loss** $\mathcal{L}_\sigma$, similarly to [304]. Formally, given a data-point (z^{vis}, j) of class j, the per-sample loss becomes:

$$
\begin{aligned}
&s_i = \langle z^{\text{vis}}, \tilde{z}_i^{\text{txt}} \rangle, \text{ where } \tilde{z}_i^{\text{txt}} \text{ is from Eq. 11.1,} \\
&\mathcal{L}_\sigma(z^{\text{vis}}, j; \tau_i) = \sum\nolimits_{i \in \mathcal{Y}^{[1:t_c]}} \log(1 + e^{-s_i \cdot \mathbf{1}_{\{i=j\}}}),
\end{aligned} \tag{11.2}
$$

where $\mathbf{1}_{\{i=j\}}$ is an indicator function that returns 1 if the candidate class i equals the ground truth class j of the synthetic example z^{vis}, and -1 otherwise.

We employ the sigmoid loss over cross-entropy primarily to enhance training efficiency. While cross-entropy requires a joint forward/backward pass through all experts, with the sigmoid loss, we can reformulate the problem into multiple independent binary tasks. This allows us to split each update step into manageable batches of experts, which can be distributed across distinct nodes in a multi-GPU or multi-node setup, since the sigmoid objective decomposes over classes. This reduces peak GPU memory as the number of classes grows, while preserving the same objective as the full update. Finally, as further discussed in the experimental section, we observe that the sigmoid loss yields a consistent beneficial effect in terms of transfer to unseen categories.

After their training, the experts are stored in the **foundational hub** $\mathcal{H}$, which grows incrementally with each task to accommodate new knowledge. Such a hub serves as a deep module library, enabling reuse and composition of expert modules for unseen tasks.

11.4.3 Expert forging via mixture of experts

During evaluation, we compare the visual embeddings extracted from the test images with the textual embeddings from the experts. For a class i of the seen set, we follow Eq. 11.1 and prompt the associated expert $E^{\text{txt}}(p_i; \theta_0 + \tau_i)$.

By contrast, for a class j from the *unseen set* (*e.g.*, one appearing in future tasks), we forge a new textual prototype by leveraging the experience accumulated up to the current task and stored in the foundational hub $\mathcal{H}$. Specifically, we first identify the K experts in $\mathcal{H}$ most relevant to the unseen class j, then combine their **weights** to form a new expert. We term this procedure the **Mixture of Textual Experts** (**MoTE**); it proceeds as follows:

$$\tilde{z}_j^{\text{MoTE}} = E^{\text{txt}}\left(p_j; \theta_0 + \boldsymbol{\tau}_j^{\text{MoTE}}\right), \qquad \boldsymbol{\tau}_j^{\text{MoTE}} = \sum\nolimits_{i \in \text{top}_K(p_j)} w_{i,j} \tau_i. \tag{11.3}$$

Here, for an unseen class j, p_j denotes its synthesized textual prompt, while $w_{i,j}$ represents an affinity score between classes i and j. In addition, $\text{top}_K(p_j)$ returns the K experts from the seen set $\mathcal{Y}^{[1:t_c]}$ that maximize the similarity $\texttt{sim}(i, j)$. We resort to text-to-text similarity in the original CLIP space, such that $\texttt{sim}(i, j) = \langle z_i^{\text{txt}}, z_j^{\text{txt}} \rangle$. We normalize these scores across the K experts with a softmax, thus obtaining the $w_{i,j}$ in Eq. 11.3.

Improving expert capabilities. Following Eq. 11.1, each expert is fed with the same hand-crafted prompt during training. However, this approach is likely to result in overfitting, with poor generalization when the input prompt varies. This would be particularly detrimental when forging the textual embedding of an unseen class j. Indeed, following Eq. 11.3, the corresponding prompt p_j provided to each expert entails a substantial domain shift for expert i, which was trained solely on p_i. Therefore, we need a tailored strategy to enhance the robustness of the experts to domain shifts and improve their out-of-distribution capabilities, which was also shown in [205] to benefit model compositionality.

In this respect, we combine two simple yet effective strategies. The first one is

template augmentation: in analogy with the concept of data augmentation, we modify the training process by randomly sampling a textual template to construct the input prompt from the 80 templates commonly used for ImageNet zero-shot tests [210].

Secondly, inspired by [276], we enhance OOD robustness by ensembling the weights $\theta_0 + \tau_i$ of each expert with those of the original zero-shot model θ_0:

$$\tilde{\theta}_i = (1-\alpha)\theta_0 + \alpha(\theta_0 + \tau_i) = \theta_0 + \alpha\tau_i, \tag{11.4}$$

where $\alpha \in [0, 1]$. We refer to this step as α-**smoothing** and denote by $\tilde{z}_j^{\alpha\text{-MoTE}}$ the embedding obtained from the *smoothed* experts. With this, the final textual embedding $\tilde{z}_j^{\alpha\text{-MoTE}}$ can be generated with a single forward pass with weights $\theta_0 + \alpha\boldsymbol{\tau}_j^{\text{MoTE}}$, as follows:

$$\tilde{z}_j^{\alpha\text{-MoTE}} = E^{\text{txt}}(p_j; \theta_0 + \alpha\boldsymbol{\tau}_j^{\text{MoTE}}). \tag{11.5}$$

11.5 Experiments

MoDER is evaluated on Class-Incremental Learning (Class-IL) [258] and Multi-domain Task Incremental Learning (MTIL) [314, 297]. Both involve a sequence of image classification tasks, assessing forgetting on the old tasks and generalization to unseen ones. MTIL evaluates transfer capabilities across distinct domains; in Class-IL, the unseen set comes from the same image domain observed during training (*e.g.*, satellite imagery).

Class-Incremental Learning. We evaluate on five datasets with varying degrees of alignment with CLIP pretraining. These are *Seq. ImageNet-R* [101], following [270, 245, 305]; two fine-grained datasets, *Seq. Cars-196* [129] and *Seq. CUB-200* [263], which contain only a few samples per class; and two out-of-distribution datasets, *Seq. EuroSAT* [98, 99] and *Seq. ISIC* [53].

Multi-domain Task Incremental Learning. MTIL comprises 11 consecutive tasks, each learning on a distinct dataset: *i.e.*, Aircraft, Caltech101, CIFAR100, DTD, EuroSAT, Flowers, Food, MNIST, OxfordPet, StanfordCars, and SUN397. Unlike Class-IL, MTIL relaxes the constraint of unknown task identities at test time. This adjustment simplifies the challenge posed by the 1201 classes spread across diverse domains, making evaluation more manageable.

Implementation details.

For a fair comparison, all methods use the same backbone. Specifically, in the Class-IL and CI-Transfer setting, we follow [77] and employ OpenAI's CLIP with ViT-L/14.

Method	IN-R	Cars	CUB	ESAT	ISIC	Avg.
CLIP [210]	82.1	66.2	50.9	55.0	22.4	55.3
AttriCLIP [266]	85.7	74.0	54.1	59.7	24.1	61.0
MoE Adapters [297]	88.2	75.8	61.7	55.8	21.1	60.5
ZSCL [314]	85.3	72.5	62.8	69.7	25.3	63.1
CGIL [77]	86.7	78.8	66.3	71.5	48.2	70.3
MoDER (LoRA)	**89.7**	**87.0**	**73.2**	74.0	**52.8**	**75.3**
MoDER (VeRA)	89.6	86.9	72.9	**74.7**	51.7	75.2

Table 11.1: The Class-Incremental Transfer on the tested benchmarks.

Method	Transf.	Δ	Avg.	Δ	Last	Δ
CLIP [210]	69.4	0.0	65.3	0.0	65.3	0.0
Continual-FT	44.6	-24.8	55.9	-9.4	77.3	+12.0
LwF [148]	58.9	-10.5	64.7	-0.6	74.6	+9.3
Wise-FT [276]	52.3	-17.1	60.7	-4.6	77.7	+12.4
ZSCL [314]	68.1	-1.3	75.4	+10.1	83.6	+18.3
MoE Adapters [297]	68.9	-0.5	76.7	+11.4	85.0	+19.7
MoDER	**69.7**	**+0.3**	**76.9**	**+11.6**	**85.8**	**+20.5**

Table 11.2: Accuracy of various approaches in the MTIL setting (Order I).

In contrast, in the MTIL setting, we follow [314, 297] and use OpenAI's CLIP with ViT-B/16.

Each diffusion model is trained from scratch for 30K iterations using the AdamW optimizer [168] with a learning rate of 1×10^{-3} and weight decay of 1×10^{-2}. To create the synthetic dataset $\mathcal{D}_{\text{SYN}}$, for each class, we sample approximately 15K embeddings in batches of size 512. During the TA phase, we train LoRA with Adam using a learning rate of 1×10^{-4} for Seq. ImageNet-R, Seq. Cars-196, and Seq. CUB-200, and 1×10^{-3} for Seq. EuroSAT and Seq. ISIC. The LoRA rank is fixed at 16. The experiments with VeRA adopt the same hyperparameters, except for the learning rates: 1×10^{-3} for Seq. ImageNet-R, Seq. Cars-196, and Seq. CUB-200, and 1×10^{-2} for Seq. EuroSAT and Seq. ISIC.

Evaluation metrics. In the Class-IL setting, the task identities remain unknown during inference. To assess performance on unseen classes, we use the *Class Incremental Transfer* (CI-Transfer) metric [77], which measures accuracy on tasks not yet

Method	IN-R	Cars	CUB	ESAT	ISIC	Avg.
CLIP [210]	81.9	65.0	50.5	53.3	26.6	55.5
CODA-Prompt [245]	78.9	45.2	72.2	63.7	47.4	61.5
AttriCLIP [266]	87.4	75.6	58.3	72.3	28.3	64.4
SLCA [305]	85.5	73.5	**87.8**	93.6	63.8	80.8
TMC [164]	63.2	39.9	63.4	65.0	48.9	56.1
Inf-LoRA [150]	84.4	58.0	80.4	79.7	56.1	71.7
MoE Adapters [297]	**90.7**	77.8	65.0	80.6	34.5	69.7
STAR-Prompt [182]	89.2	86.5	85.2	94.2	67.4	84.5
ZSCL [314]	89.1	77.7	62.4	79.1	34.1	68.5
CGIL [77]	89.4	89.3	83.1	96.2	73.0	86.2
MoDER (LoRA)	89.7	**90.1**	83.7	**96.4**	76.3	**87.2**
MoDER (VeRA)	89.4	89.6	82.7	96.3	**76.4**	86.9

Table 11.3: The Final Avg. Accuracy on the tested benchmarks.

encountered by the model (see Tab. 11.1). Additionally, we report results on the MTIL benchmark [314, 297] in Tab. 11.2. For this benchmark, we report the three standard evaluation metrics: *Transfer*, which captures changes in zero-shot accuracy; *Average*, which tracks average accuracy throughout incremental training; and *Last*, representing accuracy on the final task. Finally, in Tab. 11.3, we present results in terms of *Final Average Accuracy* to assess the impact of the different methods on seen classes. Results are averaged over three runs.

11.5.1 Comparison with the state of the art

Competing Methods. We compare MoDER against several established CL methods from recent literature. We always include a baseline showing the zero-shot performance of CLIP. Following the previous literature, in the MTIL setting we include the Continual FT baseline, which fine-tunes CLIP without any mechanisms for preventing forgetting.

Results. MoDER demonstrates strong and consistent performance across benchmarks. In particular, we achieve a clear and substantial lead in terms of CI-Transfer, outperforming other CLIP-based methods (Tab. 11.1). Furthermore, Fig. 11.2 illustrates the *CI-Transfer* progress throughout training, revealing a steeper increase in zero-shot accuracy compared to other methods, highlighting the effectiveness of our transfer technique.

	IN-R	Cars	CUB	ESAT	ISIC	Avg.
MoDER	**89.7**	**87.0**	**73.2**	**74.0**	**52.8**	**75.3**
CE loss	80.9	71.2	66.8	65.7	50.8	67.1
No Template Aug.	89.7	87.0	72.0	73.6	49.1	74.3
$\alpha = 0$	88.0	86.0	72.3	55.0	34.7	67.2

Table 11.4: Ablation analysis on CI-Transfer using LoRA-based experts.

Method	Trainable Params	GPU (MiB)
LWF [148]	149.6 M (×7.1)	32 172 (×6.7)
ZSCL [314]	149.6 M (×7.1)	26 290 (×5.5)
MoE Adapters [297]	59.8 M (×2.9)	22 358 (×4.7)
Textual Alignment	16.2 M (×0.8)	4748 (×0.99)
generator	4.7 M (×0.2)	45 (×0.01)
MoDER	20.9 M (×1)	4793 (×1)

Table 11.5: Trainable parameters and GPU memory usage for methods in MTIL.

In the MTIL benchmark, MoDER also excels, surpassing existing approaches across the *Average* and *Last* metrics (Tab. 11.2). Notably, it not only prevents zero-shot degradation but improves zero-shot performance, outperforming even CLIP's original zero-shot accuracy, as shown by the *Transfer* metric. Finally, in the standard Class-IL setting (Tab. 11.3), MoDER achieves the best average performance, albeit by a narrower margin, consistently enhancing both zero-shot generalization to new classes and retention on previously seen ones.

11.5.2 Ablative studies

We present in Tab. 11.4 the ablation studies on the Class-Incremental Transfer metric.

Sigmoid loss *vs* cross entropy loss. Besides reducing computational and memory overhead, the sigmoid loss contributes to the performance of MoDER. This gain is specific to the CI-Transfer metric, whereas its contribution to Final Average Accuracy is marginal. We argue that the sigmoid loss, which models each class label independently, provides a more suitable learning objective for experts to be composed. Since each class is learned as an independent function, the sigmoid loss allows experts to specialize in distinct concepts and be reused modularly in novel combinations. Similarly, Template Augmentation improves CI-Transfer, albeit to a lesser extent than the sigmoid loss.

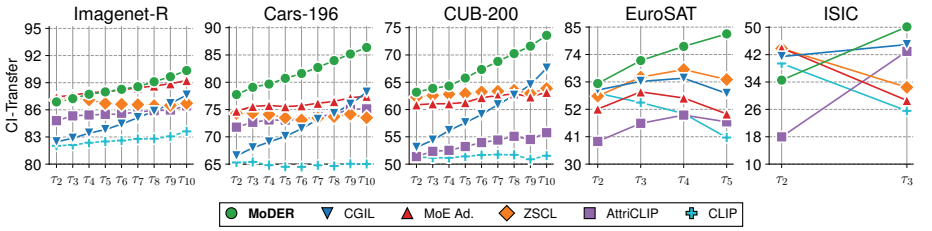


Figure 11.2: For various benchmarks, the accuracy trend in Class-Incremental transfer indicates the model's effectiveness in transferring to unseen classes in future tasks. A higher trend reflects greater effectiveness in adapting to unseen classes.

Preservation of OOD performance. The results indicate that α-smoothing significantly contributes to the performance on unseen classes. Indeed, when no smoothing is applied ($\alpha = 0$), the model achieves inferior performance, especially on out-of-domain datasets.

On the computational cost of MoDER. As shown in Tab. 11.5, MoDER achieves strong performance with substantially fewer trainable parameters and lower GPU memory usage than other baselines. This efficiency highlights its scalability and suitability for large-scale class-incremental scenarios. All results are measured on a single NVIDIA 3060 GPU with 12GB of memory. During **inference**, MoDER incurs the same overhead as the base CLIP model. Unlike other models such as STAR-Prompt [182], MoE-Adapters [297], and AttriCLIP [266], which require two forward passes, MoDER requires only a single forward pass on the visual encoder, as the text embeddings can be computed once and cached for future reuse.

11.6 Conclusions

MoDER frames incremental adaptation of CLIP as the accumulation of reusable textual experts and shows that these experts can be composed to form prototypes for unseen classes. By storing experts as task vectors τ in a foundational hub and enabling parameter-space composition at inference time, MoDER complements embedding-space replay baselines such as CGIL [77] with a mechanism that *actively* improves transfer to future tasks. This perspective anticipates the broader theme of Part III: obtaining new capabilities through model composition and merging, rather than through repeated sequential fine-tuning.

12

Accurate and Efficient Low-Rank Model Merging in Core Space

12.1 Merging low-rank experts at scale

Model composition has become a practical alternative to repeated fine-tuning: instead of training a single monolithic model for every new use case, practitioners increasingly rely on collections of task-specialized experts and combine them when needed [177, 109, 285, 298]. This trend is reinforced by widespread model distribution through public hubs [108], and by the fact that modern foundation models are expensive to fine-tune in full, which makes parameter-efficient adaptations (*e.g.*, low-rank modules) the dominant format for releasing experts [105, 106, 127, 163, 308].

However, most merging methods were developed for fully fine-tuned weights [48, 81, 109, 176, 193, 264, 285, 220]. When experts are stored as low-rank updates, a naive strategy is to reconstruct full update matrices and apply existing merge operators. Unfortunately, this is often suboptimal and can be computationally wasteful: reconstruc-

Publication. Aniello Panariello, Daniel Marczak, *et al. Accurate and Efficient Low-Rank Model Merging in Core Space.* NeurIPS, 2025 [197].

Candidate contribution. Idea, methodology, implementation, experiments, and writing.

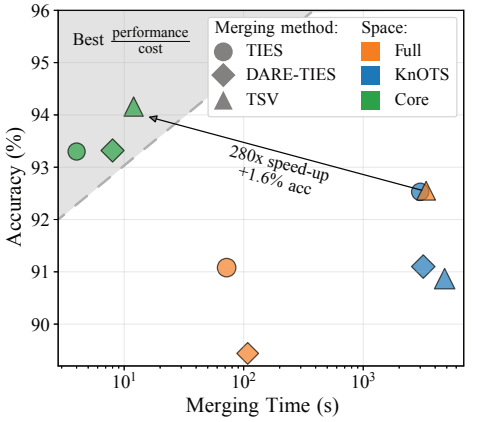


Figure 12.1: Merging in full space is fast but suboptimal (bottom center). Merging in KnOTS space or using strong merging methods (*e.g.*, TSV) improves performance but increases cost by orders of magnitude (right). **Core Space merging is effective and efficient (top left).** Results on Llama 3 8B.

tion increases memory traffic, while stronger merging methods that rely on repeated singular value decompositions (SVDs) become prohibitive on large layers. KnOTS [247] mitigates the accuracy gap by introducing an alignment space for LoRA updates, but it does so by performing SVD on concatenated *full-size* matrices, effectively abandoning the low-rank representation and scaling poorly with model size.

This chapter introduces **Core Space**, a merging framework that keeps LoRA experts low-rank throughout the pipeline while providing a shared coordinate system in which advanced merging operators become tractable. Core Space builds a *common reference basis* from the collection of low-rank factors across tasks and represents each task update with a compact *core matrix*. Importantly, the dimension of this representation depends on the number of experts and their ranks (roughly Tr), not on the base layer size. We prove that the transformation into Core Space is **lossless** for each individual task update and show that it enables both efficient merging and improved compatibility between updates.

Across both vision and language backbones (ViT-B/32, ViT-L/14, and Llama 3 8B), Core Space consistently improves the accuracy of existing merging strategies while reducing merging cost by orders of magnitude compared to prior alignment-space approaches (see Fig. 12.1). In the broader narrative of Part III, Core Space provides the technical foundation that makes composition of many parameter-efficient experts

practical at scale. The main contributions of this chapter are:

- We introduce *Core Space Merging*, a framework to merge LoRA-adapted experts in a shared low-rank basis, avoiding expensive full-space operations while improving accuracy. The framework is plug-and-play with existing merging operators.
- We prove that projection into Core Space is **lossless** for each task update and provide an explicit complexity analysis showing the resulting **efficiency gains**.
- We present extensive experiments on vision and language settings demonstrating state-of-the-art merging performance at a fraction of the computational cost of competing approaches.

12.2 Related work

Parameter-efficient experts. Large pretrained models are commonly adapted via lightweight modules that update only a small parameter subset [105, 145, 140]. Low-rank updates, in particular LoRA and variants, are widely used due to their simplicity and strong performance [106, 127, 163, 308]. In this chapter we treat each LoRA module as an expert update and focus on how to compose many such experts efficiently.

Model merging and task arithmetic. A common view of merging constructs task vectors as parameter differences from a shared base model and combines them via arithmetic or more structured operators [177, 109]. Many methods improve over naive averaging by reducing interference, *e.g.*, by pruning, resolving sign conflicts, or using low-rank structure extracted with SVD [285, 56, 264, 81, 176, 48]. Most of this literature targets fully fine-tuned weights, for which computing and storing full task vectors is feasible.

Merging LoRA-adapted models. Directly transferring full-model merging recipes to LoRA experts is non-trivial and can degrade performance [251, 247]. KnOTS [247] introduces an alignment space for LoRA merging and substantially improves accuracy, but relies on SVD of concatenated full updates, which scales poorly with large layers. Core Space addresses the same alignment goal while remaining entirely in a low-rank regime, enabling strong merging operators to be used efficiently.

12.3 Preliminaries

We adopt the *task-vector* view introduced in Chapter 9: experts are represented as parameter displacements from a shared base model and are combined through a merging operator [177, 109].

LoRA experts. Consider a weight matrix $W_0 \in \mathbb{R}^{m\times n}$ from a shared base model. A LoRA expert for task t is represented by two low-rank factors $B^{(t)} \in \mathbb{R}^{m\times r}$ and $A^{(t)} \in \mathbb{R}^{r\times n}$, defining an update

$$\Delta W^{(t)} = B^{(t)}A^{(t)}, \qquad W^{(t)} = W_0 + \Delta W^{(t)}. \tag{12.1}$$

We refer to $\Delta W^{(t)}$ as the task vector for task t.

Merging operators. Given T experts, a generic merging method constructs a merged update

$$\Delta W_{\text{merged}} = \mathcal{M}\big(\{\Delta W^{(t)}\}_{t=1}^{T}\big), \qquad W_{\text{merged}} = W_0 + \Delta W_{\text{merged}}, \tag{12.2}$$

where $\mathcal{M}$ can be linear (*e.g.*, Task Arithmetic [109]) or non-linear and structure-aware (*e.g.*, conflict reduction or SVD-based operators) [285, 81, 176]. For LoRA experts, a common baseline reconstructs each $\Delta W^{(t)}$ in full space and applies $\mathcal{M}$ there. Core Space will instead provide a compact, lossless representation that enables applying $\mathcal{M}$ efficiently without operating on the full matrices.

12.4 The Core Space merging framework

In this section, we introduce *Core Space Merging* (see Fig. 12.2), a framework designed to identify an effective and efficient subspace – referred to as the *Core Space* – in which model merging for LoRA-adapted models can be performed while remaining in the low-rank regime. Core Space is designed to be reversible – it ensures no loss of information when projecting into Core Space and back to the original space – while being as compact as possible. Compactness allows for the use of state-of-the-art merging methods relying on Singular Value Decomposition (SVD) of weight matrices, which are highly costly to perform in the original space for large models.

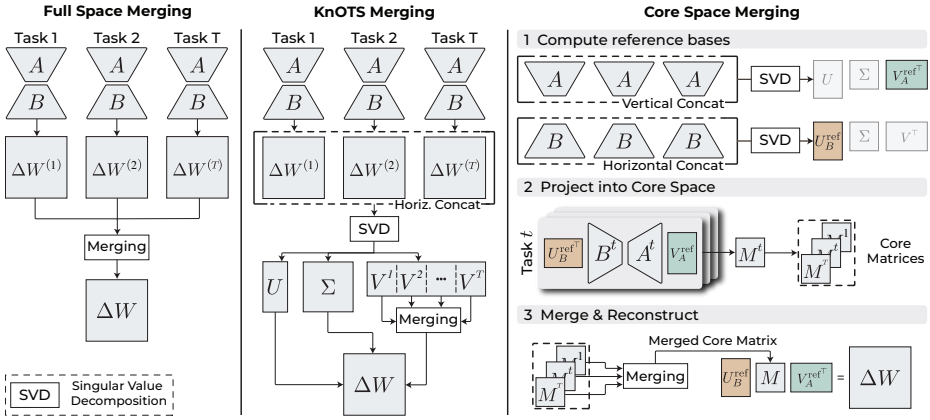


Figure 12.2: Full Space Merging (left) first reconstructs the full-space matrices $\Delta W^{(t)} = B^{(t)}A^{(t)}$, and then performs merging in the full space to obtain ΔW. **KnOTS Merging** concatenates the $\Delta W^{(t)}$ matrices and performs a costly SVD on this high-dimensional matrix. Then, the $V^{(t)}$ matrices are merged and used to obtain the final ΔW. The proposed **Core Space Merging** (right) performs SVD on a concatenation of the low-dimensional $A^{(t)}$ and $B^{(t)}$ matrices to obtain reference bases $(V_A^{\text{ref}}, U_B^{\text{ref}})$. Afterwards, it projects each update into the Core Space to obtain the core matrices $\{M^{(t)}\}_{t=1}^{T}$. It then performs merging in the Core Space and reconstructs to obtain the final ΔW.

12.4.1 Model merging in Core Space

Let $A^{(t)} \in \mathbb{R}^{r\times n}$ and $B^{(t)} \in \mathbb{R}^{m\times r}$ denote the low-rank matrices for task t, derived from a shared pretrained base model W_0. Each task update $\Delta W^{(t)} = B^{(t)}A^{(t)}$ can be reconstructed from the SVD of the matrices:

$$\begin{aligned} A^{(t)} = U_A^{(t)}\Sigma_A^{(t)}V_A^{(t)\top}, \quad B^{(t)} = U_B^{(t)}\Sigma_B^{(t)}V_B^{(t)\top}, \\ \Delta W^{(t)} = U_B^{(t)}\Sigma_B^{(t)}V_B^{(t)\top}U_A^{(t)}\Sigma_A^{(t)}V_A^{(t)\top}, \end{aligned} \tag{12.3}$$

where the shapes of the matrices in the decomposition are: $U_A^{(t)} \in \mathbb{R}^{r\times r}$, $\Sigma_A^{(t)} \in \mathbb{R}^{r\times r}$, $V_A^{(t)} \in \mathbb{R}^{n\times r}$, $U_B^{(t)} \in \mathbb{R}^{m\times r}$, $\Sigma_B^{(t)} \in \mathbb{R}^{r\times r}$, and $V_B^{(t)} \in \mathbb{R}^{r\times r}$.

Motivation. Under the hypothesis that all tasks share approximately the same common bases (U_B, V_A) such that $\forall t \in \{1,\dots,T\},\ U_B \approx U_B^{(t)}$ and $V_A \approx V_A^{(t)}$, we have:

$$\Delta W = \sum_{t=1}^{T}\Delta W^{(t)} = \sum_{t=1}^{T}B^{(t)}A^{(t)} \approx U_B\left(\sum_{t=1}^{T}\Sigma_B^{(t)}V_B^{(t)\top}U_A^{(t)}\Sigma_A^{(t)}\right)V_A^{\top}, \tag{12.4}$$

where $\Sigma_B^{(t)} V_B^{(t)\top} U_A^{(t)} \Sigma_A^{(t)} \in \mathbb{R}^{r \times r}$ encodes the directional transformation applied by the low-rank update of task t. This suggests that, under aligned bases, the sum of low-rank updates (*i.e.*, Task Arithmetic [109]) can be reduced to merging operations in a much smaller $r \times r$ space.

Projecting into the Core Space. In practice, task-specific bases are not aligned, making direct merging of core matrices as in Eq. 12.4 infeasible. Therefore, we aim to find a shared basis that can represent all tasks without loss of information and that enables merging to be performed in a reduced space compared to the full space $\Delta W^{(t)}$. Intuitively, such a shared basis should span the subspace formed by the union of the individual task subspaces.

Definition 1 (Reference Bases)**.** Given a set of low-rank matrices $\{A^{(t)}, B^{(t)}\}_{t=1}^{T}$, we define as *reference bases* the orthonormal matrices $U_B^{\text{ref}} \in \mathbb{R}^{m \times Tr}$ and $V_A^{\text{ref}} \in \mathbb{R}^{n \times Tr}$, obtained by performing SVD over the horizontally stacked $B^{(t)}$ and vertically stacked $A^{(t)}$ matrices across tasks:

$$\begin{bmatrix} B^{(1)} & \dots & B^{(T)} \end{bmatrix} = U_B^{\text{ref}} \Sigma_B V_B^{\top}; \qquad \begin{bmatrix} A^{(1)} \\ \vdots \\ A^{(T)} \end{bmatrix} = U_A \Sigma_A \left(V_A^{\text{ref}}\right)^{\top}. \tag{12.5}$$

These bases span a shared latent subspace into which all task-specific updates are projected.

Although the reference bases $\left(U_B^{\text{ref}},\ V_A^{\text{ref}}\right)$ span all task-specific directions, each task t is originally expressed in its own local bases $(U_B^{(t)}, V_A^{(t)})$. To express each update in the common coordinate system for merging, we solve the following least-squares problems:

$$R_B^{(t)} = \underset{R \in \mathbb{R}^{T \cdot r \times r}}{\operatorname{argmin}} \left\| U_B^{\text{ref}} R - U_B^{(t)} \right\|_F^2, \quad Q_A^{(t)} = \underset{Q \in \mathbb{R}^{T \cdot r \times r}}{\operatorname{argmin}} \left\| V_A^{\text{ref}} Q - V_A^{(t)} \right\|_F^2, \tag{12.6}$$

where $V_A^{(t)} \in \mathbb{R}^{n \times r}$ and $V_A^{\text{ref}} \in \mathbb{R}^{n \times T \cdot r}$ (and similarly for $U_B^{(t)}$ and U_B^{ref}). These problems are convex, and since U_B^{ref} and V_A^{ref} are orthonormal, setting the gradients to zero yields the global minimizers (see Sec. 12.6.1 for the full derivation):

$$R_B^{(t)} = {U_B^{\text{ref}}}^{\top} U_B^{(t)}, \quad Q_A^{(t)} = {V_A^{\text{ref}}}^{\top} V_A^{(t)}. \tag{12.7}$$

As we will show in Sec. 12.4.2, $\|U_B^{\text{ref}} R_B^{(t)} - U_B^{(t)}\|_F^2 = 0$, which allows us to substitute $U_B^{(t)}$ with $U_B^{\text{ref}} R_B^{(t)}$, and similarly $V_A^{(t)}$ with $V_A^{\text{ref}} Q_A^{(t)}$, in Eq. 12.3, yielding:

$$\Delta W^{(t)} = U_B^{\text{ref}} \underbrace{R_B^{(t)} \Sigma_B^{(t)} V_B^{(t)\top} U_A^{(t)} \Sigma_A^{(t)} Q_A^{(t)\top}}_{\text{task-}t\text{ update in reference coordinates}} V_A^{\text{ref}\top}. \tag{12.8}$$

By substituting the least-squares solutions from Eq. 12.7 and using the definitions of $B^{(t)}$ and $A^{(t)}$ from Eq. 12.3, we can equivalently write:

$$\Delta W^{(t)} = U_B^{\text{ref}} \left(U_B^{\text{ref}\top} B^{(t)} A^{(t)} V_A^{\text{ref}} \right) V_A^{\text{ref}\top}. \tag{12.9}$$

This reformulation expresses each $\Delta W^{(t)}$ in the reference basis, enabling all updates to be compared or merged within a shared coordinate system.

Definition 2 (Core Matrix)**.** We define the *core matrix* $M^{(t)}$ as:

$$M^{(t)} = \left(U_B^{\text{ref}\top} B^{(t)} \right) \left(A^{(t)} V_A^{\text{ref}} \right) \in \mathbb{R}^{Tr \times Tr}. \tag{12.10}$$

This formulation generalizes the middle expression in Eq. 12.4, where aligned task-specific bases were implicitly assumed. In contrast, the core matrix $M^{(t)}$ is expressed in the reference bases $\left(U_B^{\text{ref}},\ V_A^{\text{ref}}\right)$ and thus does not rely on any alignment assumption. It encodes the directional transformation applied by the low-rank update of task t, providing a compact and lossless representation of each task update in the shared reference space and enabling efficient merging in a reduced $Tr \times Tr$ space.

Reparametrized Model Merging in Core Space. Once task-specific updates $\Delta W^{(t)}$ have been reparameterized into their corresponding core matrices $M^{(t)}$ in the shared reference bases $\left(U_B^{\text{ref}},\ V_A^{\text{ref}}\right)$, Core Space Merging enables model merging to be performed entirely within a compact, aligned, low-rank subspace. Specifically, the merged update is computed by applying a merging operator $\mathcal{M}$ over the set of core matrices:

$$M_{\text{merged}} = \mathcal{M}\left(\{M^{(t)}\}_{t=1}^{T}\right), \tag{12.11}$$

where $\mathcal{M}$ may be *any merging function*, ranging from simple arithmetic averaging [109] to more advanced, non-linear or geometry-aware techniques [285, 298]. The final update in the original model space is recovered by projecting M_{merged} back through the reference bases:

$$\Delta W = U_B^{\text{ref}} M_{\text{merged}} V_A^{\text{ref}\top}. \tag{12.12}$$

Because Core Space is a lossless representation of the original updates for each individual task (see Eq. 12.9), merging in this space preserves all relevant task information. Furthermore, when $\mathcal{M}$ is *linear*, such as Task Arithmetic, the merge operation in Core Space is *exactly equivalent* to applying the same merge in the full model space:

$$\mathcal{M}(\{\Delta W^{(t)}\}_{t=1}^{T}) = \mathcal{M}(\{U_B^{\text{ref}} M^{(t)} {V_A^{\text{ref}}}^{\top}\}_{t=1}^{T}) = U_B^{\text{ref}} \mathcal{M}(\{M^{(t)}\}_{t=1}^{T}) {V_A^{\text{ref}}}^{\top}. \tag{12.13}$$

Core Space merging offers key benefits over full space merging:

- **Efficiency.** Core matrices $M^{(t)} \in \mathbb{R}^{Tr \times Tr}$ are significantly smaller than their full-space counterparts $\Delta W^{(t)} \in \mathbb{R}^{m \times n}$. This reduction enables high-cost merging algorithms to run at a fraction of the time and memory footprint (see Sec. 12.4.3).
- **Efficacy.** As shown in Sec. 12.5.1, Core Space merging improves performance over full-space merging when *non-linear* methods are used. In Sec. 12.5.2, we show that this improvement stems from better alignment and more compact representation of task-specific directions.

12.4.2 Lossless Core Space representation

Replacing $U_B^{(t)}$ and $V_A^{(t)}$ with $U_B^{\text{ref}} R_B^{(t)}$ and $V_A^{\text{ref}} Q_A^{(t)}$ to obtain Eqs. 12.8 and 12.9, which define the final form of the core matrix, requires that the solutions to the least-squares problems in Eq. 12.6 incur zero alignment error. That is,

$$\left\| U_B^{\text{ref}} R_B^{(t)} - U_B^{(t)} \right\|_F^2 = 0, \qquad \left\| V_A^{\text{ref}} Q_A^{(t)} - V_A^{(t)} \right\|_F^2 = 0. \tag{12.14}$$

In this section, we show that the reference bases U_B^{ref} and V_A^{ref}, obtained via the SVD of the stacked matrices $B^{(t)}$ and $A^{(t)}$ (see Eq. 12.5), *minimize* the total alignment error across all T tasks, achieving an error of *exactly zero*. To illustrate this, we first analyze the alignment error for a single task t, focusing on U_B^{ref}. Analogous results hold symmetrically for V_A^{ref}. For clarity, we assume in the following derivations that $T \cdot r \leq m$ and $T \cdot r \leq n$, so that the total LoRA rank does not exceed the maximum possible rank of the target weight matrix. In Sec. 12.6.3, we provide a more general analysis that removes this assumption and demonstrate that the zero alignment error result continues to hold.

Lemma. *Let $U_B^{(t)} \in \mathbb{R}^{m \times r}$ and $U_B^{\text{ref}} \in \mathbb{R}^{m \times T \cdot r}$ be matrices with orthonormal columns, and let $R_B^{(t)} = {U_B^{\text{ref}}}^{\top} U_B^{(t)} \in \mathbb{R}^{T \cdot r \times r}$ be the optimal solution minimizing the error of the*

least-square problem. Then, the optimal alignment error is given by:

$$\varepsilon_U = \left\| U_B^{\text{ref}} R_B^{(t)} - U_B^{(t)} \right\|_F^2 = r - \left\| {U_B^{(t)}}^\top U_B^{\text{ref}} \right\|_F^2. \tag{12.15}$$

The proof, provided in Sec. 12.6.2, leverages the properties of Frobenius norm and the orthonormality of $U_B^{(t)}$ and U_B^{ref}. To formally demonstrate that our chosen reference basis U_B^{ref} minimizes the alignment error across all T tasks (or equivalently maximize $||{U_B^{(t)}}^\top U_B^{\text{ref}}||_F^2$ for each task t), we first formulate the following constrained optimization problem for a single task, and then extend it to the multi-task scenario:

$$\begin{gathered} \max_{U \in \mathcal{S}} \left\| {U_B^{(t)}}^\top U \right\|_F^2 = \max_{U \in \mathcal{S}} \operatorname{tr}\left(U^\top U_B^{(t)} {U_B^{(t)}}^\top U \right), \\ \text{where} \quad \mathcal{S} = \left\{ U \in \mathbb{R}^{m \times Tr} \;\middle|\; U^\top U = I_{T \cdot r} \right\} \end{gathered} \tag{12.16}$$

and $\operatorname{tr}(\cdot)$ denotes the trace operator. The optimization domain is restricted to the Stiefel manifold $\mathcal{S}$ (*i.e.*, the set of matrices with orthonormal columns). The following lemma characterizes the solution to this optimization problem:

Lemma. *A solution U^* to the quadratic program in Eq. 12.16 is given by a basis whose columns include the r eigenvectors corresponding to nonzero eigenvalues of $B^{(t)} {B^{(t)}}^\top \in \mathbb{R}^{m \times m}$ or, equivalently, by the r left singular vectors of the matrix $B^{(t)}$. Moreover, at the optimum, the objective attains its maximum value r, resulting in zero alignment error in Eq. 12.15.*

This follows from standard constrained quadratic optimization on the Stiefel manifold.

Extension to multiple tasks. Achieving zero reconstruction error for a single model t does not guarantee optimality for any other model $t' \neq t$. Therefore, we aim to identify a reference basis U^* that jointly optimizes Eq. 12.16 across all T models. We formulate this global problem as:

$$\max_{U \in\, \mathcal{S}} \sum_{t=1}^{T} \operatorname{tr}(U^\top U_B^{(t)} {U_B^{(t)}}^\top U) = \max_{U \in\, \mathcal{S}} \operatorname{tr}(U^\top \mathbf{U}_B \mathbf{U}_B^\top U), \tag{12.17}$$

where $\mathbf{U}_B = \left[U_B^{(1)},\ U_B^{(2)},\ \ldots,\ U_B^{(T)} \right]$ denotes the horizontal concatenation of all $U_B^{(t)}$ matrices. The equality in Eq. 12.17 follows directly from the linearity of the trace

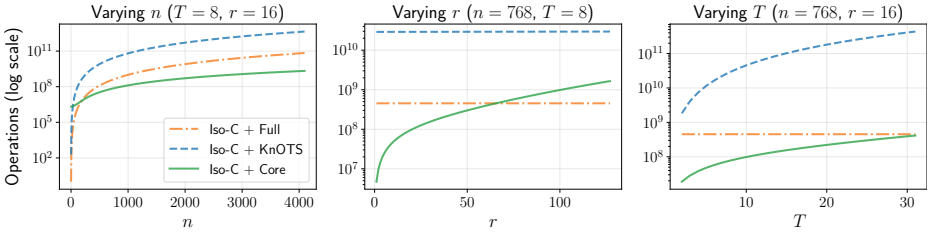


Figure 12.3: Core Space merging is more efficient than the previous state-of-the-art KnOTS. The cost is similar to full space merging, which results in much lower performance. We visualize the number of operations performed to merge T rank r LoRA modules of final shape $n \times n$.

Space	TA	Iso-C	TSV
Full	$\boldsymbol{n^2Tr}$	n^3	n^3T
KnOTS	n^3T^2	$n^3T^2 + n^2Tr$	$n^3T^2 + T^3r^2n$
Core	$\boldsymbol{n^2Tr}$	$\boldsymbol{n^2Tr + T^3r^3}$	$\boldsymbol{n^2Tr + T^4r^3}$

Table 12.1: $\mathcal{O}(\cdot)$ time complexities. The cheapest method is highlighted in **bold** ($T, r \ll n$).

operator and the distributivity of matrix multiplication concerning matrix addition: $M^\top A_1 M + M^\top A_2 M = M^\top (A_1 + A_2) M$.

By considerations analogous to the single-task case, a global solution U^* is given by the top $T \cdot r$ left singular vectors of the matrix $\mathbf{B}$, obtained by horizontally stacking each matrix $B^{(t)}$, *i.e.*, $U^* = U_B^{\text{ref}}$. This choice ensures zero alignment error simultaneously across all T tasks, consistent with the procedure described in Sec. 12.4.1.

12.4.3 Computational complexity analysis

We summarize the time complexities of TA, Iso-C, and TSV merged in all three spaces (Full, KnOTS, and our Core) in Tab. 12.1 and Fig. 12.3. Our approach exhibits a time complexity comparable to that of Task Arithmetic in full space. Our method's additional terms are negligible unless the product $T \cdot r$ becomes significantly large. A key advantage of our method lies in its scalability compared to KnOTS, whose time complexity is super-cubic, driven by a factor that scales cubically with the weight matrix size n. Finally, we emphasize the minimal additional overhead incurred when combining our method with Iso-C or TSV in the core space; it introduces a cost substantially lower than its counterpart in full space or KnOTS space.

12.5 Experimental results

Experimental setup. We follow the experimental setup of KnOTS and use the LoRA checkpoints provided by the authors [247]. For the vision experiments, we use two variants of CLIP [210] with ViT-B/32 and ViT-L/14 [64] as vision encoders fine-tuned on a standard set of 8 tasks. We employ Llama 3 8B [92] fine-tuned on 6 NLI tasks for the language experiments. All models are fine-tuned with LoRA [106] with rank 16 applied on all matrices (keys, queries, values, and outputs) across all attention layers. Following [247], we report *normalized accuracy* as a ratio of the accuracy of the merged model on a given task to the accuracy of the model fine-tuned on this task. We also report *absolute accuracy* for the joint-task setting (additional experimental details in Sec. 12.6.5).

Baseline merging spaces. We compare our proposed Core Space with two alternative merging spaces. **Full Space** operates in space of full reconstructed weight matrices $\Delta W^{(t)} = B^{(t)}A^{(t)} \in \mathbb{R}^{m\times n}$. **KnOTS Space** [247] operates in the space of the right singular vectors of the concatenated reconstructed weight matrices $\{\Delta W^{(t)}\}_{t=1}^{T} \in \mathbb{R}^{m\times nT}$. Unless otherwise stated, we adopt the same data splits, training schedules, and evaluation protocols as KnOTS to isolate the effect of the merging space and merging strategy.

Baseline merging methods. We evaluate each merging space using the following merging methods. **Task Arithmetic (TA)** [109] performs a scaled summation of each task matrix $W_{\text{merged}} = W_0 + \alpha \sum_{i=1}^{T} \Delta W_i$. As this is a linear operation, the results of merging in each space are the same (see Eq. 12.13 for Core and [247] for KnOTS). **TIES** [285] trims low-magnitude parameters and averages parameters with dominating sign, while **DARE** [298] preprocesses task vectors by randomly dropping a fraction of parameters and rescaling the remaining ones. **TSV** [81] concatenates low-rank approximations of task matrices and orthogonalizes them across tasks. **CART** [48] calculates centered task vectors as a difference of fine-tuned weights from the average of all fine-tuned weights and performs task arithmetic on the low-rank approximation of these centered task vectors. **Iso-C** [176] flattens the spectrum of singular values for a model merged with task arithmetic. As the spectrum flattening can be performed on weights merged with any merging technique, we combine Iso with other merging techniques, denoting it with +Iso-C.

Method	Space	SNLI	MNLI	SICK	QNLI	RTE	SCITAIL	Avg (ΔAcc)	Time [s]	Rel. Time
Abs. Accurcay		92.50	90.31	91.58	94.49	89.86	96.52	-	-	-
TA	Full	93.57	95.28	87.96	68.71	100.0	96.73	90.38 (+0.00)	9	-
TIES	Full	95.17	96.19	84.18	74.18	100.0	96.78	91.08 (+0.00)	72	9
	KnOTS	91.82	94.19	92.97	78.57	100.0	97.61	92.53 (+1.45)	3000	375
	Core	92.07	93.51	93.63	83.72	99.19	97.66	**93.30** (+2.22)	**8**	1
DARE-TIES	Full	94.76	96.8	78.39	72.08	98.39	96.20	89.44 (+0.00)	108	13
	KnOTS	91.62	96.72	74.90	84.75	99.48	99.13	91.10 (+1.66)	3180	397
	Core	92.10	93.58	93.70	83.68	99.19	97.66	**93.32** (+3.88)	**8**	1
TSV	Full	95.38	95.12	88.83	76.80	101.61	97.56	92.55 (+0.00)	3360	280
	KnOTS	92.53	95.83	82.77	77.01	100.0	97.08	90.87 (-1.68)	4800	400
	Core	95.86	95.70	89.25	83.89	102.42	97.86	**94.16** (+1.61)	**12**	1
Iso-C	Full	55.00	39.04	76.54	55.90	46.77	69.25	57.08 (+0.00)	540	67
	KnOTS	85.28	52.86	89.43	54.90	75.00	77.73	72.53 (+15.45)	4860	607
	Core	91.54	90.10	87.87	75.85	99.19	97.42	**90.33** (+33.25)	**8**	1

Table 12.2: Normalized accuracies of merged models on NLI tasks for Llama 3 8B.

12.5.1 Results

LLMs merging. We present Llama 3 8B results on natural language inference in Tab. 12.2. In line with our complexity analysis, merging in Core Space is much more efficient than merging in Full or KnOTS space, yielding up to a $600\times$ speedup. Moreover, merging in Core Space improves the performance of all tested merging methods. In particular, it elevates TSV to 94.16% average normalized accuracy, achieving state-of-the-art results.

Per-task evaluation in vision setting. We present per-task vision results for ViT-B/32 in Tab. 12.3. We observe that 8 out of 9 merging methods achieve their highest average accuracy when performed in our proposed Core Space. The best combination – TSV + Iso-C merged in Core Space – achieves state-of-the-art average normalized accuracy of 76.3%. It significantly outperforms the previously reported SoTA of TIES in KnOTS space, achieving 68.0% [247]. Similar conclusions hold for experiments on ViT-L/14.

Heterogeneous ranks. While handling LoRA modules with heterogeneous ranks might seem non-trivial, our method supports it seamlessly without modification. Even with different ranks, the modules can be concatenated across tasks to form an aggregate basis spanning the combined subspaces, after which projection and alignment are applied to each local task core matrix. Since SVD makes no assumptions about input ranks, it yields valid orthonormal bases in all cases, enabling our framework to merge variable-rank LoRA modules naturally. We evaluate this setting by assigning rank 16

Method	Space	Cars	DTD	ESAT	GTSRB	MNIST	RESISC	SUN397	SVHN	Avg (Δ Acc)
Abs. accuracies		74.00	58.30	99.00	92.70	99.30	88.40	64.50	96.20	-
TA	Full	81.97	73.72	48.97	42.24	53.12	71.50	97.46	41.25	63.78 (+0.00)
TIES	Full	82.37	72.72	49.91	36.62	57.16	69.38	96.92	44.56	63.70 (+0.00)
	KnOTS	83.75	74.45	50.36	47.31	67.01	71.79	96.51	50.64	67.73 (+4.03)
	Core	84.74	76.46	52.19	50.41	67.36	71.21	96.45	50.18	**68.63** (+4.93)
DARE-TIES	Full	82.14	73.72	49.35	37.78	56.63	70.14	97.35	42.12	63.65 (+0.00)
	KnOTS	82.01	72.90	44.15	45.54	60.59	70.89	95.56	47.64	64.91 (+1.26)
	Core	84.57	76.09	57.09	51.01	66.64	71.39	96.16	52.14	**69.39** (+5.74)
TSV	Full	83.44	75.55	50.99	45.03	59.31	73.33	96.40	49.23	**66.66** (+0.00)
	KnOTS	81.86	74.91	51.25	41.64	53.93	71.64	97.95	40.36	64.19 (-2.47)
	Core	83.86	75.09	52.64	45.39	58.53	72.95	97.63	45.21	66.41 (-0.25)
CART	Full	83.04	81.93	50.39	70.17	59.14	79.11	99.26	49.11	71.52 (+0.00)
	KnOTS	83.94	75.18	52.23	54.48	64.78	74.48	95.88	55.73	69.59 (-1.93)
	Core	80.83	83.94	54.99	73.28	66.25	80.95	98.69	48.57	**73.44** (+1.92)
TIES +Iso-C	Full	78.86	74.45	60.01	39.02	66.65	70.30	98.39	48.59	67.03 (+0.00)
	KnOTS	78.46	80.38	58.81	64.97	72.10	76.89	98.33	49.78	72.47 (+5.44)
	Core	82.91	84.76	52.41	78.79	71.56	81.43	99.48	52.14	**75.44** (+8.41)
DARE-TIES +Iso-C	Full	78.71	75.54	50.84	42.86	65.03	71.88	98.92	48.08	66.48 (+0.00)
	KnOTS	82.93	74.18	49.31	46.73	66.64	71.82	96.72	50.57	67.36 (+0.88)
	Core	83.27	83.12	54.55	79.04	71.83	82.08	99.36	52.37	**75.70** (+9.22)
TSV +Iso-C	Full	79.38	80.38	57.99	65.64	64.22	79.74	98.59	46.49	71.55 (+0.00)
	KnOTS	80.81	83.03	58.25	74.34	67.66	79.69	98.54	49.86	74.02 (+2.47)
	Core	82.98	85.12	50.95	84.25	71.14	84.39	99.06	53.53	**76.43** (+4.88)
CART +Iso-C	Full	80.33	82.11	57.31	77.38	71.17	81.57	98.72	51.91	75.06 (+0.00)
	KnOTS	82.05	80.47	56.12	64.58	62.40	78.81	99.22	45.05	71.09 (-3.97)
	Core	82.93	84.21	51.14	81.32	72.12	82.83	99.33	55.32	**76.15** (+1.09)
Iso-C	Full	80.16	83.03	51.44	74.76	70.72	79.89	98.66	50.20	73.60 (+0.00)
	KnOTS	80.33	79.29	57.50	67.60	65.63	79.54	99.26	46.62	71.97 (-1.63)
	Core	83.35	84.30	50.13	81.97	71.07	83.46	99.17	53.90	**75.92** (+2.32)

Table 12.3: Normalized accuracies of merged models on the vision tasks with ViT-B/32.

to half the tasks and rank 64 to the rest; we observe that our method still outperforms other approaches.

Additional PEFT methods. Our method can also be applied to other PEFT methods, such as VeRA [127]. In VeRA, $\Delta W = \Lambda_b B \Lambda_d A$, where $A \in \mathbb{R}^{r \times n}$, $B \in \mathbb{R}^{m \times r}$, $\Lambda_b \in \mathbb{R}^{1 \times m}$, and $\Lambda_d \in \mathbb{R}^{r \times 1}$. Unlike LoRA, in VeRA the A and B matrices are randomly chosen, frozen, and shared across the network, while only the two scaling vectors Λ are learned for each layer. To adapt VeRA to our Core Space merging, we absorb the scaling vectors into the matrices, *i.e.*, $\tilde{B} = \Lambda_b B$ and $\tilde{A} = \Lambda_d A$, and then treat $\tilde{A}$ and $\tilde{B}$ as the LoRA A and B matrices. We also note that the same construction extends to VeRA by absorbing the learned scaling vectors into the effective low-rank factors before stacking, projection, and reconstruction.

Joint-task evaluation in vision setting. We also evaluate vision models in the

Space	TA	TIES	DARE-TIES	TSV	CART	TIES +Iso-C	DARE-TIES +Iso-C	TSV +Iso-C	CART +Iso-C	Iso-C
Full	43.5	43.6	44.0	**45.4**	44.8	43.5	44.3	48.3	44.8	52.1
KnOTS	43.5	46.8	45.2	44.6	44.7	40.5	44.8	51.4	52.6	52.9
Core	43.5	**47.4**	**47.6**	44.5	**49.6**	**54.1**	**54.0**	**55.7**	**55.6**	**55.9**

Table 12.4: Joint-task setting absolute accuracy on the vision tasks with ViT-B/32.

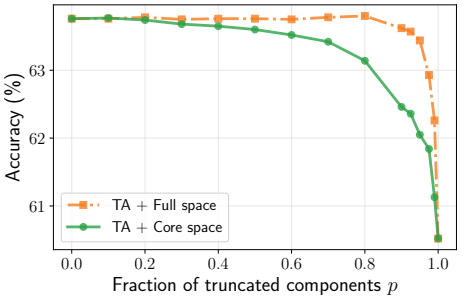


Figure 12.4: Most components in full space are irrelevant when doing Task-Arithmetic (TA). Removing any components from the core space results in a performance drop, showing that it is an information-dense space. We report the results on vision tasks with ViT-B/32.

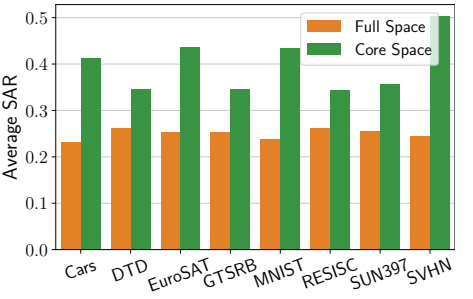


Figure 12.5: Subspace Alignment Ratio (SAR) [176]. Each bar shows the average SAR between LoRA task matrices, Full, and Core Space. In Core Space, task matrices show higher SAR. The performance gains suggest that better alignment facilitates effective merging.

challenging joint-task setting introduced in [247], in which the task ID is unknown during inference. Instead of a per-task evaluation, this protocol evaluates the merged model on the union of all classes, requiring it to distinguish between classes from all tasks. We present the results in Tab. 12.4. Core Space facilitates merging with almost all methods, achieving state-of-the-art results when combined with Iso-C.

12.5.2 Analysis

Truncation. We compare the utilization of Full Space and Core Space for models merged with TA. First, we compute the SVD of the merged matrices: ΔW_{merged} for Full Space and M_{merged} for Core Space. Then, we truncate a fraction p of the least significant values, *i.e.*, $\sigma_i = 0$ for $i > (1-p) * \dim(\Sigma)$, and observe the drop in accuracy of the merged model after truncation. As shown in Fig. 12.4, in Full Space we can truncate up to $p = 0.8$ of the values without performance loss, while in Core Space truncating any component results in a performance drop. This shows that Core Space is dense,

	Reference Basis U_B^{ref}	Shape	Avg. Acc.	Avg. ε_U
(1)	$U_B^{(1)}$ (first task)	$m \times r$	60.4	13.4
(2)	Random orthonormal	$m \times Tr$	61.6	13.3
(3)	Concatenation (Eq. 12.5)	$m \times Tr$	**68.6**	**0.0**

Table 12.5: Ablations on the choice of reference basis. Our basis (3) achieves higher results than the single-task basis (1) and the random orthonormal basis of the same shape (2). We proceed with V_A^{ref} analogously to U_B^{ref}. We report the TIES-Core results on vision tasks with ViT-B/32.

whereas Full Space contains many unused or redundant components. We hypothesize that the compactness of Core Space facilitates model merging because it extracts only the relevant components. This behavior indicates that information in Core Space is more uniformly distributed across components, making it inherently sensitive to rank reduction.

Core Space improves subspace alignment. In this section, we evaluate the Subspace Alignment Ratio (SAR) [176] between each pair of LoRA updates fine-tuned on different tasks. SAR measures how much of the subspace of one task is contained in another and correlates with post-merge performance. We compute SAR in Full Space and Core Space. Fig. 12.5 shows that Core Space yields consistently higher alignment. We argue that this result arises because Core Space enforces a shared basis across tasks, which filters out task-specific noise and promotes alignment.

Choice of the reference basis. To evaluate the choice of reference bases, we assess performance under different alternatives and compute the alignment error ε_U defined in Eq. 12.15 (averaged over all layers and tasks). We present the results in Tab. 12.5. In row (1), we evaluate using the basis of the first task as a reference basis. In row (2), we set the reference basis to a random orthonormal basis of the same dimensionality. These two bases perform much less than our reference basis in row (3). Moreover, we confirm that the optimal reference basis from row (3) achieves zero alignment error. Additionally, we verified experimentally that even in the extreme case where $T \cdot r > \min(m, n)$ (*e.g.*, $Tr = 2048 > 768$ for merging 8 ViT-B/32 LoRA models), the reconstruction error defined in Eq. 12.15 remains exactly zero, consistent with the generalized theoretical result in Sec. 12.6.3.

12.6 Proofs and additional details

12.6.1 Least-squares alignment in reference bases

We recall the least-squares problems from Eq. 12.6:

$$R_B^{(t)} = \underset{R\in\mathbb{R}^{Tr\times r}}{\operatorname{argmin}} \left\| U_B^{\text{ref}} R - U_B^{(t)} \right\|_F^2, \quad Q_A^{(t)} = \underset{Q\in\mathbb{R}^{Tr\times r}}{\operatorname{argmin}} \left\| V_A^{\text{ref}} Q - V_A^{(t)} \right\|_F^2. \tag{12.18}$$

Since U_B^{ref} and V_A^{ref} have orthonormal columns, the objectives are convex and their global minimizers are obtained by setting gradients to zero:

$$R_B^{(t)} = {U_B^{\text{ref}}}^\top U_B^{(t)}, \qquad Q_A^{(t)} = {V_A^{\text{ref}}}^\top V_A^{(t)}. \tag{12.19}$$

Substituting Eq. 12.19 into Eq. 12.8 yields the equivalent reparameterization in Eq. 12.9, which motivates the definition of the core matrices in Eq. 12.10.

12.6.2 Quantifying alignment error and proving exact reconstruction

Let $U_B^{(t)} \in \mathbb{R}^{m\times r}$ and $U_B^{\text{ref}} \in \mathbb{R}^{m\times d_U}$ be matrices with orthonormal columns, where d_U denotes the intrinsic rank of the stacked matrix in Eq. 12.5. With the least-squares solution $R_B^{(t)} = {U_B^{\text{ref}}}^\top U_B^{(t)}$, the residual is

$$\varepsilon_U \;=\; \left\| U_B^{\text{ref}} R_B^{(t)} - U_B^{(t)} \right\|_F^2 \;=\; r - \left\| {U_B^{(t)}}^\top U_B^{\text{ref}} \right\|_F^2. \tag{12.20}$$

By construction, U_B^{ref} spans the column space of the stacked matrix $\left[B^{(1)} \;\dots\; B^{(T)}\right]$ in Eq. 12.5. Therefore, for every task t, the columns of $U_B^{(t)}$ lie in $\operatorname{span}(U_B^{\text{ref}})$, which implies $\left\| {U_B^{(t)}}^\top U_B^{\text{ref}} \right\|_F^2 = r$ and hence $\varepsilon_U = 0$. The same argument applies symmetrically to V_A^{ref}, yielding zero alignment error for both factors and proving that Core Space projection and reconstruction are lossless for all tasks.

12.6.3 Overcomplete case: when $Tr > m$ or $Tr > n$

The lossless reconstruction result holds even when $Tr > m$ or $Tr > n$. In this case, the stacked matrices in Eq. 12.5 have intrinsic ranks

$$d_U = \operatorname{rank}\Big(\Big[B^{(1)} \ \dots \ B^{(T)}\Big]\Big) \le m, \qquad d_V = \operatorname{rank}([A^{(1)}; \dots; A^{(T)}]^\top) \le n,$$

and the SVD produces truncated orthonormal bases $U_B^{\text{ref}} \in \mathbb{R}^{m \times d_U}$ and $V_A^{\text{ref}} \in \mathbb{R}^{n \times d_V}$. Repeating the least-squares derivation with these truncated bases yields the same closed-form solutions as Eq. 12.19 and the same error formula as Eq. 12.20. Since U_B^{ref} and V_A^{ref} still span the union of task subspaces, the alignment errors remain zero.

12.6.4 Rank preservation of merged updates

When merging T LoRA updates of rank r, Core Space operates in a Tr-dimensional representation and reconstructs updates with effective rank at most Tr. This preserves the intended low-rank structure across merging methods that operate on core matrices. In contrast, certain full-space preprocessing steps (*e.g.*, trimming in TIES applied after reconstructing BA) can destroy low-rank structure and lead to much higher effective ranks, increasing interference.

12.6.5 Experimental environment and hyperparameter selection

The language experiments with Llama 3 8B were performed on a single 48G NVIDIA L40S. The vision experiments were executed on a single 16G NVIDIA RTX 4080. We build directly on the KnOTS codebase and use the LoRA checkpoints released by the authors.

We tune method-specific hyperparameters using a validation holdout strategy. For Task Arithmetic, the scaling factor α is searched starting at 0.1 with increments of 0.1. For TIES and DARE-TIES, top-K is searched starting at 10 with increments of 10. For DARE-TIES, the pruning factor p is searched starting at 0.1 with increments of 0.1. For CART, the pruning rank is searched over $\{0.04, 0.08, 0.16, 0.32\}$, following the original methodology.

12.7 Conclusions

This chapter introduced **Core Space**, a lossless and compact representation that enables accurate and efficient merging of LoRA experts. By projecting low-rank updates into shared reference bases, Core Space improves subspace alignment and makes strong merging operators practical on large models, delivering consistent gains across vision and language backbones while drastically reducing merging cost. In the broader dissertation narrative, Core Space provides the computational substrate for model composition: it turns collections of parameter-efficient experts into modular building blocks that can be merged reliably, supporting the transition from sequential adaptation to compositional reuse.

A limitation of Core Space is that it still operates in a single shared linear coordinate system. When experts are related only through curved low-loss paths or more complex symmetry transformations, a fixed reference basis may not fully capture their compatibility, even though each individual update is represented losslessly. Extending Core Space toward piecewise-linear, hierarchical, or kernelized alignment spaces is therefore a natural direction for future work.

13

Gradient-Sign Masking for Task Vector Transport Across Pre-Trained Models

13.1 Transporting task vectors across pre-trained models

Recent practice in deep learning increasingly relies on fine-tuning large pretrained models rather than training from scratch. This paradigm has proven effective across domains such as vision and language, where models like BERT [61], CLIP [210], and instruction-tuned successors [160, 192] serve as reusable foundations. As pretraining pipelines evolve and new checkpoints are released, however, practitioners are often forced to repeat fine-tuning on the same downstream tasks, even when the changes between pretrained models are incremental. This redundancy motivates the question

Publication. Filippo Rinaldi, **Aniello Panariello**, *et al. Gradient-Sign Masking for Task Vector Transport Across Pre-Trained Models.* ICLR, 2026 [221].

Candidate contribution. Methodology, experimental design, and writing.

of whether adaptation knowledge can be reused or transferred across pretraining runs.

Several recent lines of work suggest that such reuse is possible. Task arithmetic interprets fine-tuning updates as *task vectors* in parameter space and shows that they can be added, subtracted, or merged to induce new behaviors [109, 193, 285, 176]. In parallel, the literature on linear mode connectivity demonstrates that independently fine-tuned solutions are often connected by low-loss paths, revealing strong geometric structure in parameter space [82, 76]. More recently, model rebasin methods explicitly align independently trained models to enable parameter or task-vector transfer across pretraining runs [2, 220, 222].

Despite these advances, directly transporting a task vector from one pretrained model to another remains challenging. Even after architectural alignment, transferred updates may contain components that are misaligned with the loss geometry of the target model, leading to degraded or even harmful performance. This gap highlights a central limitation: *while task vectors encode valuable adaptation information, not all of their directions are transferable across pretrained models.* Identifying which components should be preserved and which should be discarded is therefore crucial for reliable transfer.

In this chapter, we introduce **GradFix**, a simple and principled framework for task-vector transport based on *gradient-sign masking*. Our key insight is that the sign of the gradient at the target model provides a robust proxy for locally beneficial descent directions. By retaining only those components of a source task vector whose signs agree with the target gradient, GradFix filters out harmful directions while preserving transferable structure. We provide a first-order theoretical guarantee that the resulting update reduces the target loss, and show empirically that this strategy enables effective knowledge transfer even in the low-data regime, in which only a handful of labeled samples are available. This approach is particularly appealing in settings where direct fine-tuning of the target model is expensive or when data is scarce, as it leverages the geometric information encoded in the target model's gradients to guide safe transfer of adaptation knowledge.

Within the broader narrative of Part III, GradFix complements model merging by addressing a different axis of reuse: instead of composing multiple experts trained from the same base model, it enables *transport* of task-specific knowledge across distinct pretraining runs, further expanding the space of reusable adaptations.

The main contributions of this chapter are:

- We establish a theoretical connection between the *oracle task vector*, the ideal fine-

tuning update on the target model, and quantities we can actually compute, namely the source task vector and the gradient at the zero-shot target model. We show that the sign of the zero-shot gradient provides a reliable proxy for the descent directions encoded in the target model.

- Building on this insight, we propose **GradFix**, a simple mechanism that filters the source task vector using the target model's local loss geometry. We formally prove that, to first order, the transported update reduces the target loss.
- We empirically show that our method enables effective transport of fine-tuning knowledge across pretrained models in both vision and text domains, even in the *low-data regime* where gradients must be estimated from only a handful of samples. We further validate that GradFix improves model-merging performance in both multi-task and multi-source settings, showing that the transported updates remain useful beyond single-task transfer.

13.2 Related work

Task vectors and model merging. Task arithmetic interprets fine-tuning updates as vectors in parameter space that can be combined to induce new capabilities [109]. Subsequent work improves robustness by resolving sign conflicts, pruning harmful components, or exploiting low-rank structure [285, 205, 195]. These methods typically assume a shared pretrained initialization and focus on composing multiple task updates within the same parameter space.

Model rebasin and cross-pretrain transfer. Rebasin methods aim to align independently trained models into a shared basin so that parameters or task vectors become comparable [2]. For transformers, alignment can be achieved via permutation matching or spectral techniques [110, 220, 187]. While effective, these approaches require explicit parameter alignment and often incur substantial computational overhead.

Gradient-based signals. A complementary body of work highlights the robustness of gradient sign information. SignSGD and related methods show that gradient signs alone can support convergence in noisy or distributed settings [16, 4]. More recent approaches exploit gradient-based masking or statistics for efficient adaptation and robustness [113, 142, 200]. GradFix builds on these insights, using gradient signs not for optimization itself, but as a geometric filter to enable safe task-vector transport.

13.3 Preliminaries

Let θ_A and θ_B denote the parameters of the same architecture, pretrained on different datasets (or with different hyperparameters). The fine-tuned θ_A on a downstream task is denoted by θ_A^{ft}.

Task vectors. We follow the definition of task vectors introduced earlier in the dissertation (Chapter 9) and denote by τ_A the task-specific parameter update obtained from fine-tuning θ_A^{ft}.

Cross-pretrain setting. In this chapter, the shared-initialization assumption does not hold. We consider a source task vector

$$\tau_A = \theta_A^{ft} - \theta_A$$

and aim to apply it to a different pretrained model θ_B. For reference, if the same downstream task were used to fine-tune model B, the corresponding target task vector would be $\tau_B = \theta_B^{ft} - \theta_B$. We use τ_B only as an oracle object to describe ideal transfer directions, and do not assume it is available in practice. Since θ_A and θ_B lie in different basins, directly adding τ_A to θ_B can introduce directions that increase the target loss. Rather than explicitly aligning parameterizations, we seek to identify which components of τ_A are compatible with the loss geometry of θ_B, motivating the gradient-based filtering strategy introduced next.

13.4 Method

GradFix is a framework for transferring task vectors across different pretrained models by filtering them with gradient information from the target model. As a conceptual starting point, we consider an *oracle* ideal setting where the target task vector, obtained from full fine-tuning, defines the ideal transferable directions (Sec. 13.4.1). We then approximate the oracle with a *single gradient step* on the target model, using gradient signs to capture an approximate direction of the full fine-tuning trajectory. This yields a *gradient-sign mask* that selectively filters the source task vector into a compatible update (Sec. 13.4.2). Finally, we extend the approach to the limited-data regime, where gradients are estimated from only a handful of labeled samples (Sec. 13.4.3).

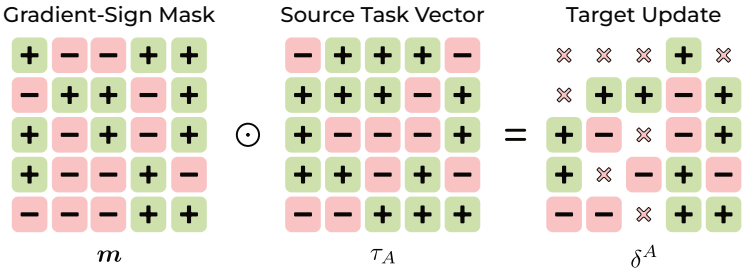


Figure 13.1: Illustration of our masking procedure. The gradient mask $\boldsymbol{m}$ suppresses harmful directions in the task vector τ_A while preserving those aligned with the target model.

13.4.1 GradFix (gradient-sign masking)

We begin by considering an ideal scenario where the true fine-tuned target task vector τ_B of model B and the whole target dataset $\mathcal{D}$ are available. Such a vector represents the optimal parameter change to adapt B on the target dataset $\mathcal{D}$. In this ideal setting, it is possible to construct a mask that perfectly retains only the components of a candidate update (*e.g.*, τ_A) that are aligned with τ_B, ensuring that every retained coordinate contributes to decreasing the loss. In other words, τ_B (or its sign structure) defines the "gold standard" for locally beneficial directions. Formally, we define as $\boldsymbol{m}^\star \in \{0,1\}^d$ the mask induced by τ_B, where d is the total number of model parameters and $i \in \{1,\dots,d\}$ indexes each coordinate:

$$m_i^\star = \mathbb{1}\{\mathrm{sign}(\tau_{A,i}) = \mathrm{sign}(\tau_{B,i})\}. \tag{13.1}$$

As shown in Fig. 13.1, applying this mask to τ_A produces the oracle-masked update $\delta^\star$, which preserves only the components consistent with τ_B:

$$\delta^\star := \boldsymbol{m}^\star \odot \tau_A, \tag{13.2}$$

where $\odot$ denotes element-wise multiplication. This vector $\delta^\star$ represents a reliable transfer of τ_A onto θ_B, since it filters out all components of τ_A that are misaligned with the true adaptation directions of B. In practice, however, τ_B (and thus $\delta^\star$) is unavailable because it requires access to the fine-tuned target model θ_B^{ft}, which defeats the purpose of transporting the solution from A to B. To approximate this ideal mask, we use the gradient of the zero-shot target model as a surrogate for τ_B, since it captures

locally beneficial directions:

$$\boldsymbol{g} := \nabla_\theta \mathcal{L}(\theta_B), \quad \mathcal{L}(\theta) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f_\theta(x), y)], \tag{13.3}$$

where ℓ is the training objective (*e.g.*, cross-entropy) and (x, y) is a labeled example from $\mathcal{D}$. Based on this gradient, we define the gradient-sign mask $\boldsymbol{m}$, which retains only the components of τ_A whose sign matches that of the corresponding anti-gradient coordinate:

$$m_i := \mathbb{1}\{\text{sign}(\tau_{A,i}) = \text{sign}(-g_i)\}. \tag{13.4}$$

Intuitively, $-\boldsymbol{g}$ acts as a signal for local alignment with the loss geometry of B. Notably, in the idealized setting where B is fine-tuned using full-batch gradient descent for a single epoch, the resulting task vector τ_B is proportional to $-\boldsymbol{g}$, so the gradient-sign mask coincides with the oracle. This observation justifies using the gradient-sign mask as an approximation of the ideal update, even when only a few labeled examples are available. The mask retains only components of τ_A whose sign matches the anti-gradient of $\mathcal{L}(\theta_B)$, pruning coordinates that would increase the loss for target model B. In this way, the gradient-sign mask serves as a practical surrogate for the trajectory-informed directions encoded in the unavailable τ_B.

13.4.2 Transporting the update

Given the gradient-sign mask $\boldsymbol{m}$ from Eq. 13.4, we define the updated target parameters by directly applying the masked task vector with a scaling factor $\alpha > 0$:

$$\theta_B^{\text{trans}} = \theta_B + \delta^A, \quad \delta^A := \alpha\,(\boldsymbol{m} \odot \tau_A). \tag{13.5}$$

We explicitly use the convention $\boldsymbol{g} = \nabla_\theta \mathcal{L}(\theta_B)$ (ascent direction), so useful transport directions should align with $-\boldsymbol{g}$. It is important to note that τ_A points in a descent direction for model A, whereas the gradient $\boldsymbol{g}$ of the target model points in the ascent direction of its loss. By selecting the coordinates aligned with $-\boldsymbol{g}$ and then adding δ^A, each retained component moves along a descent-aligned direction for B. In contrast, $\delta^\star$ is oracle-aligned with the target task direction since its mask is constructed from τ_B.

This construction induces a coordinate-wise filtering effect: we do not alter the direction of retained entries, but only suppress coordinates that are sign-incompatible

with the target anti-gradient. As a result, the transported update preserves task information from τ_A while reducing the risk of injecting locally harmful directions into θ_B. We now formalize this intuition with a first-order loss analysis.

Descent guarantee. To understand why this gradient masking provides effective transfer, we analyze its effect on the loss of the target model B. Consider the transported update from Eq. 13.5. By expanding the target loss $\mathcal{L}$ around θ_B via a first-order Taylor approximation, we obtain:

$$\mathcal{L}(\theta_B + \delta^A) \approx \mathcal{L}(\theta_B) + \boldsymbol{g}^\top \delta^A, \quad \text{where } \boldsymbol{g} = \nabla_\theta \mathcal{L}(\theta_B). \tag{13.6}$$

The sign of the inner product $\boldsymbol{g}^\top \delta^A$ determines whether the update increases or decreases the loss to first order. By construction, the gradient-sign mask $\boldsymbol{m}$ retains only components of τ_A that are aligned with $-\boldsymbol{g}$. Concretely, for each coordinate i, we have:

$$g_i \cdot (m_i \tau_{A,i}) = \begin{cases} -|g_i|\,|\tau_{A,i}|, & \text{if } \operatorname{sign}(\tau_{A,i}) = \operatorname{sign}(-g_i), \\ 0, & \text{otherwise}, \end{cases} \tag{13.7}$$

which is always nonpositive. Coordinates with $g_i = 0$ or $\tau_{A,i} = 0$ are naturally covered by this expression and contribute zero. Therefore, the overall inner product satisfies the following:

$$\boldsymbol{g}^\top \delta^A = -\alpha \sum_i m_i |g_i|\,|\tau_{A,i}| \;\leq\; 0. \tag{13.8}$$

Thus, for sufficiently small α, the update δ^A is guaranteed to be a descent direction for $\mathcal{L}$. Practically, the mask removes all sign-mismatched components of τ_A, so that every retained entry contributes to reducing the loss. Without masking, τ_A could contain harmful directions that increase the loss for B; with masking, the transported update is locally aligned with the descent geometry of the target model.

13.4.3 Limited data regime

In Sec. 13.4.1, we have assumed access to the full target dataset $\mathcal{D}$ to compute the gradient $\boldsymbol{g}$ at the zero-shot target model θ_B. In practice, one of the main motivations for task vector transport is the *few-shot* or limited data regime. If we had access to the entire dataset, we could directly fine-tune θ_B to obtain θ_B^{ft}, making task vector transfer unnecessary in that setting.

When only a small number of samples is available, we estimate the anti-gradient

signs using a subset of labeled examples. Let $\mathcal{D}_s \subset \mathcal{D}$ denote a small subset of N samples. For each parameter coordinate i, we compute the sign of the anti-gradient via **majority voting** across these samples:

$$\hat{s}_i = \operatorname{sign}\left(-\sum\nolimits_{(x_n,y_n)\in\mathcal{D}_s} \operatorname{sign}\left(\nabla_\theta \ell\left(f_{\theta_B}(x_n), y_n\right)\right)\right). \tag{13.9}$$

Lemma (Concentration of Majority Vote Sign Estimator)**.** *Let the probability that a single-sample gradient sign matches the true gradient sign at coordinate i be*

$$p_i = \Pr[\operatorname{sign}(\nabla_\theta \ell(f_{\theta_B}(x), y)) = \operatorname{sign}(g_i)].$$

Then, under mild independence assumptions and for $p_i > 1/2$, the majority-vote estimator satisfies:

$$\Pr\left[\hat{s}_i = \operatorname{sign}(-g_i)\right] \geq 1 - \exp\left(-2N(p_i - 1/2)^2\right), \tag{13.10}$$

which shows that the estimated sign concentrates around the true anti-gradient direction as the number of samples N grows.

The proof of this lemma uses Hoeffding's inequality [104]. In practice, even a few samples provide a robust estimate of the true descent direction. Each gradient acts as a vote for the correct sign, and majority voting filters out noisy or conflicting directions. This implies that, with high probability, the masked task vector δ^A points in a descent direction, preserving the first-order loss reduction behavior of the full-data update. As shown in Sec. 13.5.3, this approach is robust to small sample sizes, making it particularly attractive when direct fine-tuning of θ_B is expensive or prone to overfitting. Compared to mean-based aggregation, majority voting is less sensitive to outlier magnitudes because it depends only on sign frequency and provides a more stable transfer (Sec. 13.5.4).

13.5 Experimental results

Implementation details. For the vision settings, we consider CLIP ViT-B/16 and ViT-L/14 Vision Transformers [210], implemented in Open-CLIP [45]. The original pretrained weights are denoted θ_A and the target model weights θ_B. For ViT-B/16, θ_A was pretrained on Datacomp XL (`s13b`, `b90k`) and θ_B on LAION-2B (`s34b`, `b88k`). For ViT-L/14, θ_A was pretrained on Datacomp XL (`s13b`, `b90k`) and θ_B on LAION-

2B (`s32b`, `b82k`). For the language settings, we investigated different Text-To-Text Transfer Transformer (T5) [211] models in the base configuration. As θ_A, we used `T5v1.1`, pretrained on the C4 [211] dataset without any supervised training. For θ_B, we used `FLAN-T5` [49], pretrained and instruction-tuned on several datasets, including GSM8K [52], AQUA-RAT [157], and LAMBADA [199]. Task vectors were obtained following the fine-tuning protocol of [109]: 2000 iterations, batch size 128, learning rate 1×10^{-5}, cosine annealing with 200 warm-up steps, AdamW optimizer [168], weight decay 0.1. The text encoder backbone was kept frozen following [45].

Baselines. We evaluate our method against several baselines. As a lower bound, we consider the zero-shot target model (θ_B *zero-shot*), *i.e.*, the base model without any fine-tuning. As an upper bound, we report $\theta_B + \delta^\star$, obtained by adding the source task vector τ_A masked with the signs of the true task vector τ_B to the target model. We also include the performance of the fully fine-tuned target model (θ_B *fine-tune*) and the naive task arithmetic transport ($\theta_B + \tau_A$). In addition, we compare against *TransFusion* [220], which transports task vectors across transformer-based models via permutation alignment. Finally, we report the accuracy of a target model fine-tuned with the same number of randomly sampled examples per class $|D_s|$ used by our approach.

Supervision Budget $\mathcal{D}_s$. In all experiments, the subset $\mathcal{D}_s$ is drawn from the full downstream fine-tuning dataset $\mathcal{D}$ and constitutes only a fraction of its size. Throughout the tables, $|\mathcal{D}_s^c|$ indicates the number of examples *per class* used to estimate gradient signs for the target model θ_B. The subset $\mathcal{D}_s$ always represents a very small fraction of the full dataset, often well below 1% of the available supervision. We also evaluated different strategies for constructing $\mathcal{D}_s$ (random, herding, k-medoids, coreset). While structured selection yields small gains in the extreme few-shot regime, random selection remains a strong baseline and approaches structured methods as $|\mathcal{D}_s|$ increases.

13.5.1 Transport experiments

Transport in the Vision Setting. Tab. 13.1 summarizes the results of task vector transport across CLIP ViT-B/16 and ViT-L/14 architectures, averaged over multiple random seeds that determine the composition of the sampled $\mathcal{D}_s$ (standard deviations are reported across seeds). Our GradFix, denoted by $\theta_B + \delta^A$, yields a consistent improvement over naive task vector addition ($\theta_B + \tau_A$) even when using a single sample per class to approximate the true gradient signs. Notably, naive addition performs

		EUROSAT		SVHN		GTSRB		RESISC45		DTD	
Model	$\lvert\mathcal{D}_s^c\rvert$	**B/16**	**L/14**	**B/16**	**L/14**	**B/16**	**L/14**	**B/16**	**L/14**	**B/16**	**L/14**
θ_B *zero-shot*	-	49.41	62.80	50.58	37.28	48.29	56.12	67.98	73.12	55.96	63.35
θ_B *fine-tune*	-	98.70	98.95	97.45	97.80	98.65	99.16	95.66	97.06	83.19	83.56
$\theta_B + \tau_A$	-	49.58	62.77	50.84	39.09	49.31	56.03	67.87	73.49	56.27	63.56
$\theta_B + \delta^\star$	-	95.06	96.75	92.04	92.60	82.92	88.65	87.06	90.30	71.44	72.66
TransFusion	-	50.12	63.21	53.26	37.38	50.24	56.78	67.99	73.36	56.70	64.10
θ_B^{opt}	1	56.61	64.65	61.32	62.51	56.08	63.97	69.25	74.54	56.21	63.76
$\theta_B + \delta^A$	1	61.94	69.67	71.07	70.15	60.88	66.82	70.05	76.45	58.32	65.50
θ_B^{opt}	2	59.49	70.76	62.01	45.23	61.70	69.91	71.20	76.62	57.00	64.97
$\theta_B + \delta^A$	2	65.07	74.10	70.19	54.31	64.33	71.55	71.42	76.97	58.51	66.10
θ_B^{opt}	5	61.99	69.75	67.03	67.11	63.08	73.25	73.01	75.41	59.65	66.72
$\theta_B + \delta^A$	5	66.05	75.59	73.59	74.41	66.61	73.14	71.57	76.82	60.02	66.95

Table 13.1: Cross-dataset performance comparison between ViT-B/16 and ViT-L/14 models.

nearly at the level of zero-shot initialization and fails to transfer meaningful task knowledge. This confirms that GradFix effectively suppresses misaligned components of τ_A, preventing negative transfer due to pretraining mismatch.

To further evaluate our approach, we compare it against few-shot fine-tuning of θ_B, denoted as θ_B^{opt}, using the same limited target samples. A brief computational cost comparison is reported below. GradFix achieves better performance, on both ViT-B/16 and ViT-L/14, while exhibiting smaller variance across seeds with respect to few-shot fine-tuning. Moreover, as the $\mathcal{D}_s$ size increases, our method continues to provide stable gains, whereas θ_B^{opt} suffers from fluctuations and instability due to the composition of the supervision dataset. These results demonstrate that our approach ensures consistent and reliable task vector transport, remaining stable across different subsets $\mathcal{D}_s$. Importantly, this robustness is achieved with a single forward-backward pass to obtain the mask $\boldsymbol{m}$, highlighting the efficiency and simplicity of the proposed method.

Computational cost. GradFix requires a single forward-backward pass on the target model to estimate gradient signs, followed by a lightweight masked update. Given the number of parameters P, this results in approximately $8P$ FLOPs, compared to $16P$ FLOPs for a single fine-tuning step and several orders of magnitude more for full fine-tuning, making GradFix highly efficient in low-data settings.

Transport in the Language Setting. Tab. 13.2 reports results on task vector transport

Model	$\lvert\mathcal{D}_s^c\rvert$	SNLI	MNLI	RTE	QNLI	SCITAIL	AVG
θ_B *zero-shot*	-	34.24	35.21	47.20	50.54	50.38	43.51
θ_B *fine-tune*	-	88.20	86.30	84.40	92.79	95.32	89.40
$\theta_B + \tau_A$	-	31.61	30.75	47.36	50.52	50.46	42.12
$\theta_B + \delta^\star$	-	58.69	69.97	72.93	65.32	62.38	65.86
θ_B^{opt}	50	35.09	26.05	47.29	51.45	51.78	42.33
$\theta_B + \delta^A$	50	68.06	49.68	54.25	60.50	59.89	58.48

Table 13.2: Cross-dataset performance of T5 models on different NLP tasks

across T5 models evaluated on closed-vocabulary text classification benchmarks. While direct addition of τ_A to θ_B fails to transfer knowledge effectively, our method closes the gap toward full fine-tuning, confirming its ability to identify and retain task-relevant directions. Notably, the relative improvement over naive transfer is even larger than in the vision setting, underscoring the robustness of our approach in domains where task transfer is especially challenging. This demonstrates that the benefits of GradFix are not confined to vision, and that a single forward-backward pass suffices to enable reliable and efficient task vector transport also in the language domain.

13.5.2 Task vector transport for model merging

We evaluate GradFix in combination with model-merging methods, specifically Task Arithmetic [109] and TIES-Merging [285]. We consider two settings: *multi-task* (one source model, multiple tasks) and *multi-source* (multiple source models, one task).

Multi-task experiments. For this setting, all task vectors are extracted from the same source pretrained model θ_A (same pretraining, different downstream tasks) and transported to a fixed target pretrained model θ_B. For task vectors from distinct tasks, we compare two pipelines. **Mask–then–Merge**: transport each task vector with GradFix and then merge. **Merge–then–Mask**: first merge task vectors into τ_{merged}, then transport the merged vector using a consensus mask computed by estimating per-parameter anti-gradient signs on θ_B for each task and selecting the most frequent sign at each coordinate. We report results for this setting in Tab. 13.3. Here, $\theta_B + \tau_{A,j}$ and $\theta_B + \delta_j^A$ denote single-task references evaluated on task j, respectively using naive addition and GradFix. Direct Task Arithmetic and TIES merging without GradFix perform near zero-shot, indicating strong cross-model misalignment, while **Merge–then–Mask** gives the best results. Masking each vector first can discard coordinates

Pipeline	EUROSAT	SVHN	GTSRB	RESISC45	DTD	AVG
θ_B *zero-shot*	49.41	50.58	48.29	67.98	55.96	54.44
Multi-task $(\{\tau_{A,j}\}_{j=1}^{T} \to \theta_B)$						
$\theta_B + \tau_{A,j}$	49.58	50.84	49.31	67.87	56.27	54.77
$\theta_B + \delta_j^A$	65.07	70.19	**64.33**	71.42	**58.51**	65.90
Task Arithmetic						
Baseline	49.31	50.99	48.73	68.05	56.54	54.73
Mask-then-Merge	55.90	71.56	59.65	71.40	57.66	63.23
Merge-then-Mask	65.37	72.10	59.55	71.16	57.07	65.05
TIES-Merging						
Baseline	49.15	50.75	48.95	67.97	56.54	54.67
Mask-then-Merge	50.41	64.28	54.05	69.51	57.13	59.08
Merge-then-Mask	**65.62**	**72.42**	62.73	**71.57**	57.77	**66.02**
Multi-source $(\{\tau_{A_k}\}_{k=1}^{K} \to \theta_B)$						
Task Arithmetic						
Baseline	36.94	35.09	30.77	50.54	44.73	39.61
Merge-then-Mask	12.52	15.94	4.20	3.97	2.93	7.91
Mask-then-Merge	**65.96**	**72.97**	**65.80**	71.30	61.01	**67.41**
TIES-Merging						
Baseline	12.69	10.39	2.85	5.81	16.33	9.61
Merge-then-Mask	14.46	15.94	3.09	3.35	2.66	7.90
Mask-then-Merge	65.17	72.58	65.36	**71.40**	**61.12**	67.13

Table 13.3: Merging experiments on ViT-B/16 in multi-task and multi-source settings.

that would complement each other after merging; merging first preserves them and lets GradFix align one coherent update with the target loss geometry.

Multi-source experiments. In the multi-source setting, we use $K = 5$ source models $\{\theta_{A_k}\}_{k=1}^{K}$ that are pretrained on different data distributions and then fine-tuned on the same downstream task. We transport all resulting source task vectors to a single fixed target model θ_B, whose pretraining remains unchanged during transport. Here, **Merge–then–Mask** is not expected to help: because all vectors correspond to the same task, the gradient-sign mask on θ_B is shared across sources, so consensus masking after merging is effectively equivalent to masking each source separately. We therefore use **Mask–then–Merge**: transport each source vector first, then merge the transported vectors. Tab. 13.3 shows that this recovers performance from the collapse of direct merging and also improves over single-source transport, suggesting that combining multiple transported updates provides a more robust descent direction.

Model	Mask Strategy	EUROSAT	RESISC45	GTSRB	SVHN	DTD	AVG
θ_B *zero-shot*	-	49.41	67.98	48.29	50.58	55.96	54.45
θ_B *fine-tune*	-	98.70	95.66	98.65	97.45	83.19	94.73
$\theta_B + \delta^\star$	sign agreement	95.06	87.06	82.92	92.04	71.44	85.71
	sign forcing	97.95	93.51	95.94	96.60	80.59	92.92
	magnitude-scaled	49.92	67.94	51.63	50.78	56.01	55.25
$\theta_B + \delta^A$	sign agreement (ours)	61.94	70.05	60.89	71.07	58.32	64.45
	sign forcing	61.32	70.10	60.91	70.52	58.05	64.18
	magnitude-scaled	49.51	68.06	49.20	50.71	56.03	54.70
$\theta_B + \delta^A$	random	49.49	67.97	48.41	50.54	56.06	54.50

Table 13.4: Performance of θ_B with oracle or estimated gradient signs under different mask strategies: **sign agreement** retains matching signs, **sign forcing** aligns all signs, **magnitude-scaled** uses the product of task and gradient magnitudes, and **random** assigns signs uniformly. Results averaged over seeds with $|\mathcal{D}_s^c| = 1$ on CLIP ViT-B/16. The table includes θ_B *zero-shot* and θ_B *fine-tune* as lower and upper reference bounds.

13.5.3 Masking strategies

To analyze the effect of different mask construction strategies on the transport of the task vector τ_A, we compare our primary masking method (sign agreement) with three alternatives: **sign forcing**, **magnitude-scaled**, and **random** masks. Tab. 13.4 reports results using 1 sample per class on ViT-B/16, averaged across multiple random seeds.

For both δ^A (Eq. 13.5) and $\delta^\star$ (Eq. 13.2), the mask $\boldsymbol{m}$ determines which directions of τ_A are retained. Here, i indexes parameter coordinates. In **sign agreement**, $\boldsymbol{m}$ retains only the coordinates whose signs match those of the reference as in Eq. 13.4. In **sign forcing**, all signs of τ_A are aligned with the signs of the anti-gradient estimator $\hat{\boldsymbol{s}}$, obtaining:

$$m_i^{sf} = \text{sign}(\tau_{A,i}) \cdot \hat{s}_i, \qquad \delta_i^{sf} = \alpha\,(m_i^{sf}\tau_{A,i}) = \alpha\,|\tau_{A,i}|\,\hat{s}_i. \tag{13.11}$$

This flips entries that disagree with the reference and applies a forced-sign update. When the oracle τ_B is used as the reference, sign forcing generally outperforms sign agreement, as fully leveraging the true task direction maximizes transfer. In contrast, using few-shot anti-gradient-based estimates from θ_B, sign agreement performs slightly better than sign forcing. This is consistent with the fact that the anti-gradient estimate is noisy; forcing all directions can propagate errors, while keeping only agreeing entries provides a more reliable mask. We also evaluated a **magnitude-scaled** strategy to determine whether leveraging magnitude information offers additional benefits. This

strategy computes the mask based on the magnitude of both the source task vector and the target reference vector ρ (where $\rho = \tau_B$ for the oracle and $\rho = -g$ for the estimate):

$$m_i^{ms} = \max(0, \tanh(\tau_{A,i} \cdot \rho_i)), \qquad \delta_i^{ms} = \alpha\,(m_i^{ms}\tau_{A,i}). \tag{13.12}$$

This approach assigns a mask value near 1.0 only when components match in sign and possess significant magnitude, while suppressing mismatches. However, as shown in Tab. 13.4, this magnitude-aware strategy consistently underperforms sign agreement. Crucially, this failure persists even in the oracle setting ($\delta^\star$), which suggests that while *directions* are transferable across different pretraining runs, parameter *magnitudes* are highly specific to the local loss geometry of each basin. Consequently, enforcing magnitude consistency acts as an overly aggressive filter, discarding valid updates where the models agree on direction but differ on scale. In the limited-data regime (δ^A), this issue is further exacerbated by the noise inherent in estimating gradient magnitudes from small datasets $\mathcal{D}_s$, confirming that the sign structure serves as the most robust proxy for alignment.

Finally, we evaluate **random mask**, where signs are sampled uniformly from $\mathcal{U}\{-1, +1\}$. Like magnitude-scaled masking, this approach performs close to the zero-shot baseline. This highlights that although sign-based masking outperforms magnitude-based filtering, the gain is not due to masking alone: random sign choices offer no useful signal. Thus, effective transfer relies strictly on the precise geometric alignment provided by the target anti-gradients, rather than just the sparsity of the update.

13.5.4 Sensitivity to the scaling coefficient

We investigate the sensitivity of masked transport to the scaling factor $\alpha \in (0, 1]$, providing a proxy for how compatible and robust the transported task vector is with the target backbone. In addition to our proposed *majority voting* strategy, we consider a baseline where the estimated sign is taken as the sign of the averaged anti-gradient (*mean*). Results are reported in Fig. 13.2.

In the main experiments, we select α by a coarse sweep on the validation split of each dataset-backbone pair. The trends in Fig. 13.2 show that this calibration does not need to be precise: with majority voting, performance remains strong over a broad range of values. Indeed, across datasets, majority voting consistently outperforms the

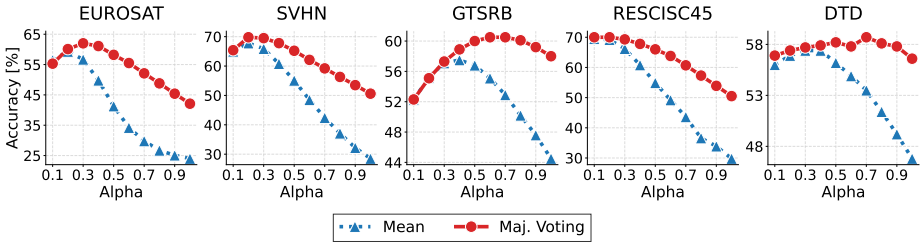


Figure 13.2: Accuracy at different α values for mean and majority voting sign selections.

mean strategy for all values of α, providing a more reliable approximation of the true gradient sign. Notably, majority voting yields smooth performance curves without sudden drops, and maintains higher accuracy over a broader range of α. This difference arises from the aggregation mechanism; averaging gradients before thresholding is highly sensitive to variance and outliers, so even a small subset of misaligned samples can flip the estimated sign and destabilize updates as α grows. Majority voting, instead, depends only on the relative frequency of signs, which concentrates rapidly around the true direction with increasing samples (as predicted by the concentration behavior of majority voting). As a result, it is inherently more stable and preserves transfer accuracy even in few-shot or noisy regimes.

From a practical perspective, this robustness means that masked transport with majority voting does not require fine-grained tuning of α to achieve good performance. The method remains effective across a wide range of scaling choices, which is particularly valuable when adapting to new datasets where validation data or tuning budgets are limited.

13.5.5 Subset data selection

Beyond random sampling, we analyzed whether structured strategies for constructing $\mathcal{D}_s$ improve anti-gradient-sign estimation. We compared random selection with feature-based alternatives from CLIP embeddings: herding, k-medoids, and a coreset-style medoid-proximity greedy selection. Across datasets, structured selectors yield small gains at very low budgets, while random sampling remains competitive and approaches their performance as the number of examples per class increases. Because random sampling adds no embedding/distance-computation overhead and does not require full target-data access, it is often preferable in constrained settings.

13.6 Conclusions

This chapter showed that the sign structure of gradients provides a powerful and robust signal for transporting task-specific knowledge across pretrained models. By masking a source task vector using gradient signs from the target model, **GradFix** filters out misaligned directions and guarantees a first-order decrease in the target loss. Empirically, this enables effective task transfer in both vision and language settings, even in low-data regimes where full fine-tuning is impractical.

Unlike methods that rely on a single global linear coordinate system, GradFix uses a local first-order approximation around the target model and retains only the coordinates of τ_A that remain descent-aligned for θ_B. This makes it more tolerant when global linear mode connectivity is weak, but it does not remove the limitation entirely: if successful transfer requires a strongly non-linear path between pretraining runs, a single masked update may recover only part of the reusable structure. Extending GradFix to iterative relinearization or curvature-aware transport is a promising direction for future work.

In the context of this dissertation, GradFix complements model merging and compositional adaptation by addressing a distinct but related problem: how to reuse fine-tuning knowledge across evolving pretraining pipelines. Together with the methods presented earlier in Part III, it contributes to a broader view of learning systems as modular, transferable, and reusable across time, tasks, and models.

Summary of Part III: Learning through model composition

This part explored model composition as an alternative paradigm for adaptation in evolving systems, shifting the focus from sequential optimization to the structured reuse of learned knowledge. Instead of updating model parameters through further training, adaptation was achieved by explicitly representing task-specific information and combining it through algebraic operations in parameter space. This perspective becomes especially relevant when target data are scarce, replay is impractical, or adaptation must happen after deployment, since new behaviors can be assembled from previously learned components rather than learned again from scratch.

Across different problem settings, composition enabled flexible forms of transfer. PASTA showed how condition-specific modules can be composed to adapt tracking models to external factors without retraining, reducing negative interference across incompatible scenarios while preserving efficient deployment. MoDER extended this idea to continual learning with vision–language models, showing that textual experts accumulated over time can be recombined to synthesize prototypes for unseen classes, thus turning incremental experience into zero-shot generalization capability.

Core Space and GradFix addressed two complementary challenges that emerge when composition is scaled up. Core Space introduced a lossless and compact repres-

entation that improves alignment and stability when combining multiple parameter-efficient updates, making strong merging operators practical even for large pretrained models. GradFix tackled the problem of task transport, enabling task vectors learned on one pretrained model to be transferred across architectures by retaining only the update directions that remain locally descent-aligned for the target model.

Together, these contributions establish model composition as a principled mechanism for knowledge reuse across tasks, conditions, and models. More broadly, they show that adaptation in modern learning systems need not rely exclusively on sequential retraining: once knowledge is encoded in modular updates, prototypes, or low-rank experts, it can be selected, merged, and transported as an object in its own right. In this sense, the final part of the dissertation completes the progression from learning through temporal structure, to learning across tasks, to learning directly through reusable model components.

Conclusion

This dissertation investigated how modern deep models can acquire, retain, and reuse knowledge across time, tasks, and model instances under realistic constraints on data, computation, and retraining. Starting from limitations of static learning paradigms in evolving settings, the work progressively developed a unified view in which adaptation is represented explicitly, manipulated algebraically, and transferred compositionally, rather than being repeatedly re-learned through end-to-end optimization.

From sequential adaptation to reusable knowledge

In the first part, the dissertation investigated learning from non-stationary visual data, where evolution arises from temporal structure in the input. Rather than focusing on task boundaries or parameter retention, the contributions showed how temporal coherence can act as an implicit supervisory signal. Across anomaly detection, multi-object tracking, and per-object distance estimation, temporal consistency, probabilistic modeling, and object-centric representations enabled robust learning under weak, noisy, or incomplete supervision. In this setting, adaptation is driven by structure in the data stream itself, without requiring explicit task identifiers or replay mechanisms.

The second part shifted the notion of evolution from data streams to sequences of tasks. Here, the problem is no longer to exploit temporal regularities in the input, but to prevent catastrophic forgetting while acquiring new knowledge over time. The proposed methods addressed this challenge through replay-based and generative mechanisms that reshape the learning signal rather than constraining parameter updates. Importantly, these works emphasized representation reuse and efficiency, showing that continual learning systems can preserve prior capabilities while supporting incremental adaptation, even in large pretrained models.

Together, these parts highlight a key insight that motivates the final stage of the dissertation: knowledge learned over time, whether from temporal structure or task sequences, need not be repeatedly re-optimized. Instead, much of it can be retained in structured representations that are stable, reusable, and amenable to recombination. This observation sets the stage for the third part, where evolution is no longer confined to data or task streams, but occurs directly in model space through composition and merging.

Composition as a unifying principle

The third part of the dissertation unified these observations under the paradigm of model composition. Rather than viewing learning as a sequence of irreversible parameter updates, this part reframed adaptation as the construction, manipulation, and transport of task representations.

Across tracking, classification, and language understanding, task knowledge was consistently represented as structured updates relative to a pretrained model, denoted as task vectors τ. Whether instantiated as low-rank adaptation modules, class-specific experts, or full fine-tuning deltas, these representations shared two crucial properties: they were compact and they admitted algebraic operations.

This perspective enabled several complementary advances. First, task representations could be composed to synthesize new behaviors without retraining, as shown in condition-aware tracking and zero-shot class composition. Second, multiple adaptations could be merged efficiently and reliably by operating in structured low-dimensional spaces, leading to accurate and scalable low-rank model merging. Third, task knowledge could be transported across different pretrained models by filtering task vectors through the local loss geometry of the target model, enabling reuse even when pretraining distributions differ.

Together, these results demonstrate that task vectors are not merely optimization artifacts, but meaningful objects that encode transferable structure. When handled carefully, they enable adaptation without data access, without replay, and without repeated fine-tuning.

Key insights and implications

Several core insights emerge from this work.

First, adaptation structure matters more than adaptation procedure. Many failures of naive merging or transfer arise not because task knowledge is absent, but because it is represented or combined in incompatible ways. Explicit representations, aligned bases, and geometry-aware filtering are sufficient to recover much of this lost potential.

Second, parameter-efficient updates are not a limitation but an enabler. Low-rank and sparse representations do not merely reduce cost. They expose the linear structure necessary for composition, interpolation, and transport, making them central to scalable lifelong learning systems.

Third, knowledge reuse can replace retraining in many settings. Across the presented chapters, competitive or superior performance was achieved with minimal or no additional optimization, suggesting a path toward models that adapt by reasoning over prior solutions rather than relearning them.

Limitations and future directions

Despite these advances, several challenges remain. Composition is not universally safe, and interference can still arise when task representations are poorly aligned or when assumptions about shared basins break down. Extending these methods to highly heterogeneous architectures, multimodal settings with asymmetric encoders, or reinforcement learning remains an open direction.

More broadly, this dissertation points toward a shift in how learning systems are designed. Rather than monolithic models that are repeatedly overwritten, future systems may resemble libraries of reusable components, where learning consists of producing, storing, and recombining task representations over time.

Final remarks

In summary, this dissertation proposed a coherent framework for learning across time, tasks, and models by treating adaptation as a compositional object. By grounding continual learning, zero-shot transfer, model merging, and cross-model transport in a shared representation space, it demonstrated that many forms of adaptation can be achieved without retraining, provided knowledge is represented explicitly and manipulated correctly. This perspective opens the door to more efficient, modular, and sustainable learning systems, capable of evolving alongside data, tasks, and models without forgetting what they already know.

Appendix

List of publications

This section lists all works published during the candidate's Ph.D., including journal articles, conference proceedings, and preprints.

Conference papers

- Filippo Rinaldi, **Aniello Panariello**, Giacomo Salici, Fengyuan Liu, Marco Ciccone, Angelo Porrello, Simone Calderara. *Gradient-Sign Masking for Task Vector Transport Across Pre-Trained Models.* International Conference on Representation Learning, 2026.
- **Aniello Panariello**, Daniel Marczak, Simone Magistri, Angelo Porrello, Bartłomiej Twardowski, Andrew D. Bagdanov, Simone Calderara, Joost van de Weijer. *Accurate and Efficient Low-Rank Model Merging in Core Space.* Proceedings of the Conference on Neural Information Processing Systems, 2025.
- **Aniello Panariello**, Emanuele Frascaroli, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, Simone Calderara. *Modular Embedding Recomposition for Incremental Learning.* Proceedings of the British Machine Vision Conference, 2025.

- Gianluca Mancusi, Mattia Bernardi, **Aniello Panariello**, Angelo Porrello, Simone Calderara, Rita Cucchiara. *Is Multiple Object Tracking a Matter of Specialization?*. Proceedings of the Conference on Neural Information Processing Systems, 2024.
- Emanuele Frascaroli, **Aniello Panariello**, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, Simone Calderara. *CLIP with Generative Latent Replay: A Strong Baseline for Incremental Learning*. Proceedings of the British Machine Vision Conference (Oral), 2024.
- Matteo Mosconi, Andriy Sorokin, **Aniello Panariello**, Angelo Porrello, Jacopo Bonato, Marco Cotogni, Luigi Sabetta, Simone Calderara, Rita Cucchiara. *Mask and Compress: Efficient Skeleton-Based Action Recognition in Continual Learning*. Proceedings of the International Conference on Pattern Recognition (Oral), 2024.
- Gianluca Mancusi, **Aniello Panariello**, Angelo Porrello, Matteo Fabbri, Simone Calderara, Rita Cucchiara. *TrackFlow: Multi-Object Tracking with Normalizing Flows*. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.
- **Aniello Panariello**, Angelo Porrello, Simone Calderara, Rita Cucchiara. *Consistency-Based Self-Supervised Learning for Temporal Anomaly Localization*. Proceedings of the European Conference on Computer Vision Workshops, 2022.

Journal articles

- **Aniello Panariello**, Gianluca Mancusi, Fedy Haj Ali, Angelo Porrello, Simone Calderara, Rita Cucchiara. *Monocular Per-Object Distance Estimation with Masked Object Modeling*. Computer Vision and Image Understanding, 2025.

Preprints

- Filippo Rinaldi, **Aniello Panariello**, Giacomo Salici, Angelo Porrello, Simone Calderara. *Transporting Task Vectors across Different Architectures without Training*. arXiv preprint arXiv:2602.12952, 2026.

Under review

- Carlos Garrido-Munoz, **Aniello Panariello**, Silvia Cascianelli, Angelo Porrello, Simone Calderara, Jorge Calvo-Zaragoza, Rita Cucchiara. *Zero-Shot Synthetic-to-Real Handwritten Text Recognition via Task Analogies*.

- Christos Georgakilas, **Aniello Panariello**, Simone Calderara, Dimosthenis Karatzas, Joost van de Weijer. *Rethinking Expert Training for Model Merging with Prompt Learning*.

Activities during the Ph.D.

This section summarizes the candidate's main activities during the Ph.D., including teaching, supervision, project participation, dissemination, and reviewing.

Teaching activities

- Laboratory assistant for the course "Machine Learning and Deep Learning" in the Master's Degree in Computer Engineering, University of Modena and Reggio Emilia. Academic years 2022/2023 to 2025/2026.
- Laboratory assistant for the course "Intelligenza Artificiale e Tecnologie Web" in the Bachelor's Degree in Computer Engineering, University of Modena and Reggio Emilia. Academic year 2025/2026.
- Lecturer for the Executive Master "Intelligenza Artificiale, Machine Learning e Deep Learning", organized by Fondazione Democenter-Sipe. Editions of June 2023 and June 2024 (Modena, Italy).
- Lecturer for the Executive Program "Machine Learning and Deep Learning", organized by the BI-REX Consortium. Editions of January–February 2024 and December 2024 (Bologna, Italy).
- Lecturer for seminars on applications of Machine Learning and Deep Learning to industrial scenarios, organized by the BI-REX Consortium in collaboration with SEW Eurodrive. Edition of 2024.
- Lecturer for the Intensive Master "AI and ML for Smart Factory", organized by Experis SRL. Edition of November 2024–February 2025 (remote).
- Lecturer for the professional training course "Variational Autoencoders and Synthetic Data", organized by a multinational energy company. May 2023 (remote).
- Lecturer for the professional training course "Weakly Supervised Anomaly Detection", organized by Ammagamma. March 2023 (Modena, Italy).

Participation in national and international projects

- **InSecTT** – Intelligent Secure Trustable Things (H2020-ICT-2018-2020).

Supervision of master's and bachelor's theses

- Object Detection: confronto tra YOLOX e DETR. Francesca Morandi, 2023 (Bachelor's Thesis).
- Analisi di Transformer Encoder in ambito Knowledge Distillation per la Person Re-Identification. Alessandro Castellucci, 2023 (Master's Thesis).
- Ottimizzazione di Tecniche di Class-Incremental Learning con l'utilizzo di Tecniche di Prompt Tuning e Latent Replay. Riccardo Santi, 2024 (Master's Thesis).

Research visits

- Visiting research period at the Computer Vision Center (CVC), Universitat Autònoma de Barcelona, under the supervision of Dr. Joost van de Weijer. April–September 2025 (Barcelona, Spain).

Dissemination activities

- Poster presentation of the work "Accurate and Efficient Low-Rank Model Merging in Core Space" at NeurIPS 2025, San Diego, California, USA.
- Poster presentation of the work "Modular Embedding Recomposition for Incremental Learning" at the 2025 British Machine Vision Conference (BMVC), Sheffield, UK.
- Participation in the 2025 International Computer Vision Summer School, Sicily, Italy.
- Oral presentation of the work "CLIP with Generative Latent Replay: A Strong Baseline for Incremental Learning" at the 2024 British Machine Vision Conference (BMVC), Glasgow, UK.
- Participation in the 2024 European Conference on Computer Vision (ECCV), Milan, Italy.
- Poster presentation of the work "TrackFlow: Multi-Object Tracking with Normalizing Flows" at ICCV 2023, Paris, France.
- Participation in the 2023 ELLIS Summer School on Large-Scale AI for Research and Industry, Modena, Italy.
- Participation in the 2023 VISMAC International Summer School on Machine Vision, Padova, Italy.

Review activities

- IEEE/CVF Computer Vision and Pattern Recognition (CVPR), 2026, 2025, 2024.
- IEEE/CVF European Conference on Computer Vision (ECCV), 2026, 2024.
- Transactions on Machine Learning Research (TMLR), 2026, 2025.
- IEEE/CVF International Conference on Computer Vision (ICCV), 2025, 2024, 2023.
- Association for the Advancement of Artificial Intelligence (AAAI), 2025.
- Neural Information Processing Systems (NeurIPS), 2025, 2024.
- British Machine Vision Conference (BMVC), 2025, 2024 (outstanding), 2023.
- Journal of Visual Communication and Image Representation (JVCI), 2024, 2023.
- International Conference on Image Analysis and Processing (ICIAP), 2023.

Bibliography

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[2] Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023.

[3] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018.

[4] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, 2017.

[5] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International Conference on Learning Representations*, 2022.

[6] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

[7] Favyen Bastani, Songtao He, and Samuel Madden. Self-supervised multi-object tracking with cross-input consistency. In *Advances in Neural Information Processing Systems*, 2021.

[8] Stefan Becker, Ronny Hug, Wolfgang Hubner, and Michael Arens. Red: A simple but effective baseline predictor for the trajnet benchmark. In *European Conference on Computer Vision Workshops*, 2018.

[9] Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger Grosse, and Jörn-Henrik Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2021.

[10] Giovanni Bellitto, Federica Proietto Salanitri, Matteo Pennisi, Matteo Boschini, Lorenzo Bonicelli, Angelo Porrello, Simone Calderara, Simone Palazzo, and Concetto Spampinato. Saliency-driven experience replay for continual learning. In *Advances in Neural Information Processing Systems*, 2024.

[11] Yannick Benezeth, P-M Jodoin, Venkatesh Saligrama, and Christophe Rosenberger. Abnormal events detection based on spatio-temporal co-occurences. In *IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

[12] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011.

[13] Michael Berger, Jose Moreno, Johnny A Johannessen, Pieternel F Levelt, and Ramon F Hanssen. Esa's sentinel missions in support of earth system science. *Remote Sensing of Environment*, 2012.

[14] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *IEEE International Conference on Computer Vision*, 2019.

[15] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[16] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, 2018.

[17] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. Monoloco: Monocular 3d pedestrian localization and uncertainty estimation. In *IEEE International Conference on Computer Vision*, 2019.

[18] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE International Conference on Image Processing*. IEEE, 2016.

[19] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*. IEEE, 2017.

[20] Jacopo Bonato, Francesco Pelosin, Luigi Sabetta, and Alessandro Nicolosi. Mind: Multi-task incremental network distillation. In *AAAI Conference on Artificial Intelligence*, 2024.

[21] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[22] Matteo Boschini, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Continual semi-supervised learning through contrastive interpolation consistency. *Pattern Recognition Letters*, 2022.

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

[24] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

[25] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *Advances in Neural Information Processing Systems*, 2020.

[26] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[27] Ruichu Cai, Hao Zhang, Wen Liu, Shenghua Gao, and Zhifeng Hao. Appearance-motion memory consistency network for video anomaly detection. In *AAAI Conference on Artificial Intelligence*, 2021.

[28] Simone Calderara, Uri Heinemann, Andrea Prati, Rita Cucchiara, and Naftali Tishby. Detecting anomalies in people's trajectories using spectral graph analysis. *Computer Vision and Image Understanding*, 2011.

[29] Luca Candeloro, Carla Ippoliti, Federica Iapaolo, Federica Monaco, Daniela Morelli, Roberto Cuccu, Pietro Fronte, Simone Calderara, Stefano Vincenzi, Angelo Porrello, et al. Predicting wnv circulation in italy using earth observation data and extreme gradient boosting model. *Remote Sensing*, 2020.

[30] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv preprint arXiv:2203.14360*, 2022.

[31] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020.

[32] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[33] Silvia Cascianelli, Gabriele Costante, Alessandro Devo, Thomas A Ciarfuglia, Paolo Valigi, and Mario L Fravolini. The role of the input in natural language video description. *IEEE Transactions on Multimedia*, 2019.

[34] Silvia Cascianelli, Gabriele Costante, Francesco Crocetti, Elisa Ricci, Paolo Valigi, and Mario Luca Fravolini. Data-based design of robust fault detection and isolation residuals via lasso optimization and bayesian filtering. *Asian Journal of Control*, 2021.

[35] Giulia Castagnolo, Concetto Spampinato, Francesco Rundo, Daniela Giordano, and Simone Palazzo. A baseline on continual learning methods for video action recognition. In *IEEE International Conference on Image Processing*, 2023.

[36] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for

temporal action localization. In *IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[37] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision*, 2018.

[38] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations*, 2019.

[39] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *IEEE International Conference on Multimedia and Expo*, 2018.

[40] Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems*, 2019.

[41] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

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

[43] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *IEEE International Conference on Computer Vision*, 2021.

[44] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *IEEE*, 2017.

[45] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[46] Hsu-kuang Chiu, Antonio Prioletti, Jie Li, and Jeannette Bohg. Probabilistic 3d multi-object tracking for autonomous driving. *arXiv preprint arXiv:2001.05673*, 2020.

[47] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Empirical Methods in Natural Language Processing*, 2014.

[48] Jiho Choi, Donggyun Kim, Chanhyuk Lee, and Seunghoon Hong. Revisiting weight averaging for model merging. *arXiv preprint arXiv:2412.12153*, 2024.

[49] 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.

[50] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Advances in Neural Information Processing Systems Workshops*, 2014.

[51] Ross A Clark, Benjamin F Mentiplay, Emma Hough, and Yong Hao Pua. Three-dimensional cameras and skeleton pose tracking for physical function assessment: A review of uses, validity, current developments and kinect alternatives. *Gait & posture*, 68:193–200, 2019.

[52] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[53] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *IEEE International Symposium on Biomedical Imaging*, 2018.

[54] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[55] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE conference on Computer Vision and Pattern Recognition*, 2005.

[56] Mohammad-Javad Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision*, 2024.

[57] Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision*, 129(4):845–881, 2021.

[58] Patrick Dendorfer, Vladimir Yugay, Aljoša Ošep, and Laura Leal-Taixé. Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking? In *Advances in Neural Information Processing Systems*, 2022.

[59] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on Computer Vision and Pattern Recognition*, 2009.

[60] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

[61] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[62] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. In *International Conference on Learning Representations*, 2015.

[63] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

[64] 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 *International Conference on Learning Representations*, 2020.

[65] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning*, 2018.

[66] Yong Du, Yun Fu, and Liang Wang. Skeleton based action recognition with convolutional neural network. In *Proceedings of the Asian Conference on Computer Vision*, 2015.

[67] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE conference on Computer Vision and Pattern Recognition*, 2015.

[68] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Dg-stgcn: dynamic spatial-temporal modeling for skeleton-based action recognition. *arXiv preprint arXiv:2210.05895*, 2022.

[69] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Pyskl: Towards good practices for skeleton action recognition. In *ACM International Conference on Multimedia*, 2022.

[70] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *IEEE International Conference on Computer Vision*, 2019.

[71] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, 2014.

[72] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Aljoša Ošep, Riccardo Gasparini, Orcun Cetintas, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. Motsynth: How can synthetic data help pedestrian detection and tracking? In *IEEE International Conference on Computer Vision*, 2021.

[73] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. Pstnet: Point spatio-temporal convolution on point cloud sequences. In *International Conference on Learning Representations*, 2021.

[74] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *IEEE International Conference on Computer Vision*, 2017.

[75] Jia-Chang Feng, Fa-Ting Hong, and Wei-Shi Zheng. Mist: Multiple instance self-training framework for video anomaly detection. In *IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[76] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, 2020.

[77] Emanuele Frascaroli, Aniello Panariello, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Clip with generative latent replay: a strong baseline for incremental learning. In *British Machine Vision Conference*, 2024.

[78] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *IEEE International Conference on Computer Vision*, 2017.

[79] Ruopeng Gao, Yijun Zhang, and Limin Wang. Multiple object tracking as id prediction. *IEEE conference on Computer Vision and Pattern Recognition*, 2025.

[80] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, 2016.

[81] Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *IEEE conference on Computer Vision and Pattern Recognition*, 2025.

[82] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, 2018.

[83] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[84] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE conference on Computer Vision and Pattern Recognition*, 2012.

[85] Ross Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, 2015.

[86] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[87] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *IEEE International Conference on Computer Vision*, 2019.

[88] Walter C Gogel. The visual perception of size and distance. *Vision Research*, 1963.

[89] Fatih Gökçe, Göktürk Üçoluk, Erol Şahin, and Sinan Kalkan. Vision-based detection and distance estimation of micro unmanned aerial vehicles. *Sensors*, 15(9):23805–23846, 2015.

[90] María Teresa González-Aparicio, Roberto García, JL Brugos, Xabiel G Pañeda, David Melendi, and Sergio Cabrero. Measuring temporal redundancy in sequences of video requests in a news-on-demand service. *Telematics and Informatics*, 2014.

[91] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[92] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and Abhinav Pandey *et al.* The llama 3 herd of models. *arXiv preprint arXiv: 2407.21783*, 2024.

[93] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *IEEE conference on Computer Vision and Pattern Recognition*, 2016.

[94] Muhammad Abdul Haseeb, Jianyu Guan, Danijela Ristic-Durrant, and Axel Gräser. Disnet: a novel method for distance estimation from monocular camera. *10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS*, 2018.

[95] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on Computer Vision and Pattern Recognition*, 2016.

[96] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision*, 2017.

[97] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[98] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IEEE International Geoscience and Remote Sensing Symposium*, 2018.

[99] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[100] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, 2016.

[101] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *IEEE International Conference on Computer Vision*, 2021.

[102] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[103] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[104] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 1963.

[105] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, 2019.

[106] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[107] Xing Hu, Shiqiang Hu, Yingping Huang, Huanlong Zhang, and Hanbing Wu. Video anomaly detection using deep incremental slow feature analysis network. *IET Computer Vision*, 2016.

[108] Hugging Face. https://huggingface.co, 2024. URL https://huggingface.co.

[109] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2022.

[110] Moritz Imfeld, Jacopo Graldi, Marco Giordano, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. Transformer Fusion with Optimal Transport. In *International Conference on Learning Representations*, 2024.

[111] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

[112] Longlong Jing, Ruichi Yu, Henrik Kretzschmar, Kang Li, Charles R Qi, Hang Zhao, Alper Ayvaci, Xu Chen, Dillon Cower, Yingwei Li, et al. Depth estimation matters most: Improving per-object depth estimation for monocular 3d detection and tracking. In *International Conference on Robotics and Automation*, 2022.

[113] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U. Stich, and Martin Jaggi. Error feedback fixes SignSGD and other gradient compression schemes. In *International Conference on Machine Learning*, 2019.

[114] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[115] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[116] Oluwafunmilola Kesa, Olly Styles, and Victor Sanchez. Multiple object tracking and forecasting: Jointly predicting current and future object locations. In *IEEE Winter Conference on Applications of Computer Vision Workshops*, 2022.

[117] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *IEEE International Conference on Computer Vision*, 2015.

[118] Jaechul Kim and Kristen Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

[119] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[120] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

[121] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[122] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018.

[123] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *National Academy of Sciences*, 2017.

[124] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, 2017.

[125] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[126] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *International Journal of Computer Vision*, 2022.

[127] Dawid J Kopiczko, Tijmen Blankevoort, and Yuki M Asano. Vera: Vector-based random matrix adaptation. In *International Conference on Learning Representations*, 2024.

[128] Dipali Koshti, Supriya Kamoji, Nehal Kalnad, Suyash Sreekumar, and Shreya Bhujbal. Video anomaly detection using inflated 3d convolution network. In *International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2020.

[129] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2013.

[130] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 2017.

[131] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[132] Suren Kumar, Tim K Marks, and Michael Jones. Improving person tracking using an inexpensive thermal infrared sensor. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

[133] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 2011.

[134] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *IEEE conference on Computer Vision and Pattern Recognition*, 2014.

[135] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[136] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[137] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *IEEE*, 1998.

[138] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[139] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[140] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[141] Bo Li, Yuchao Dai, Xuelian Cheng, Huahui Chen, Yi Lin, and Mingyi He. Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN. In *IEEE International Conference on Multimedia and Expo Workshops*, 2017.

[142] Haoling Li, Xin Zhang, Xiao Liu, Yeyun Gong, Yifan Wang, Qi Chen, and Peng Cheng. Enhancing large language model performance with gradient-based parameter selection. In *AAAI Conference on Artificial Intelligence*, 2025.

[143] Siyuan Li, Tobias Fischer, Lei Ke, Henghui Ding, Martin Danelljan, and Fisher Yu. Ovtrack: Open-vocabulary multiple object tracking. In *IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[144] Tianjiao Li, Qiuhong Ke, Hossein Rahmani, Rui En Ho, Henghui Ding, and Jun Liu. Else-net: Elastic semantic network for continual action recognition from skeleton data. In *IEEE International Conference on Computer Vision*, 2021.

[145] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[146] Yingwei Li, Tiffany Chen, Maya Kabkab, Ruichi Yu, Longlong Jing, Yurong You, and Hang Zhao. R4d: Utilizing reference objects for long-range distance estimation. In *International Conference on Learning Representations*, 2022.

[147] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *Machine Intelligence Research*, 2023.

[148] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[149] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[150] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *IEEE conference on Computer Vision and Pattern Recognition*, 2024.

[151] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 2021.

[152] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *ACM International Conference on Multimedia*, 2017.

[153] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *European Conference on Computer Vision*, 2018.

[154] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *IEEE International Conference on Computer Vision*, 2019.

[155] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[156] Weiyao Lin, Ming-Ting Sun, Radha Poovandran, and Zhengyou Zhang. Human activity recognition for video surveillance. In *IEEE International Symposium on Circuits and Systems*, 2008.

[157] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.

[158] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[159] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, 2022.

[160] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

[161] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[162] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. Gsm: Graph similarity model for multi-object tracking. In *International Joint Conference on Artificial Intelligence*, 2020.

[163] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *International Conference on Machine Learning*, 2024.

[164] Tian Yu Liu and Stefano Soatto. Tangent model composition for ensembling and continual fine-tuning. In *IEEE International Conference on Computer Vision*, 2023.

[165] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[166] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[167] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for Continual Learning. In *Advances in Neural Information Processing Systems*, 2017.

[168] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[169] Zhichao Lu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Retinatrack: Online single stage joint detection and tracking. In *IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[170] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 2021.

[171] Hui Lv, Chuanwei Zhou, Zhen Cui, Chunyan Xu, Yong Li, and Jian Yang. Localizing anomalies from weakly-labeled videos. *IEEE Transactions on Image Processing*, 2021.

[172] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *European Conference on Computer Vision*. Springer, 2020.

[173] Hanspeter A Mallot, Heinrich H Bülthoff, JJ Little, and Stefan Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 1991.

[174] Gianluca Mancusi, Aniello Panariello, Angelo Porrello, Matteo Fabbri, Simone Calderara, and Rita Cucchiara. Trackflow: Multi-object tracking with normalizing flows. In *IEEE International Conference on Computer Vision*, 2023.

[175] Gianluca Mancusi, Mattia Bernardi, Aniello Panariello, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Is multiple object tracking a matter of specialization? In *Advances in Neural Information Processing Systems*, 2024.

[176] Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *International Conference on Machine Learning*, 2025.

[177] Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, 2021.

[178] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*, 1989.

[179] Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pretrained models for continual learning. In *Advances in Neural Information Processing Systems*, 2023.

[180] Jefferson Ryan Medel and Andreas Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016.

[181] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[182] Martin Menabue, Emanuele Frascaroli, Matteo Boschini, Enver Sangineto, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Semantic residual prompts for continual learning. In *European Conference on Computer Vision*, 2024.

[183] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[184] Alessio Monti, Angelo Porrello, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. How many observations are enough? knowledge distillation for trajectory forecasting. In *IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[185] Brendan Tran Morris and Mohan Manubhai Trivedi. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[186] Matteo Mosconi, Andriy Sorokin, Aniello Panariello, Angelo Porrello, Jacopo Bonato, Marco Cotogni, Luigi Sabetta, Simone Calderara, and Rita Cucchiara. Mask and compress: Efficient skeleton-based action recognition in continual learning. In *International Conference on Pattern Recognition*, 2024.

[187] Anshul Nasery, Jonathan Hayase, Pang Wei Koh, and Sewoong Oh. Pleas-merging models with permutations and least squares. In *IEEE conference on Computer Vision and Pattern Recognition*, 2025.

[188] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems*, 2011.

[189] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[190] David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of IEEE International Conference on Neural Networks*, 1994.

[191] Jaehoon Oh, Sungnyun Kim, Namgyu Ho, Jin-Hwa Kim, Hwanjun Song, and Se-Young Yun. Understanding cross-domain few-shot learning based on domain similarity and few-shot difficulty. In *Advances in Neural Information Processing Systems*, 2022.

[192] OpenAI, Josh Achiam, and Steven Adler et al. Gpt-4 technical report. *arXiv preprint arXiv: 2303.08774*, 2023.

[193] Guillermo Ortiz-Jiménez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, 2023.

[194] Aniello Panariello, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Consistency based self-supervised learning for temporal anomaly localization. In *European Conference on Computer Vision Workshops*, 2022.

[195] Aniello Panariello, Emanuele Frascaroli, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Modular embedding recomposition for incremental learning. In *British Machine Vision Conference*, 2025.

[196] Aniello Panariello, Gianluca Mancusi, Fedy Haj Ali, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Monocular per-object distance estimation with masked object modeling. *Computer Vision and Image Understanding*, 2025.

[197] Aniello Panariello, Daniel Marczak, Simone Magistri, Angelo Porrello, Bartłomiej Twardowski, Andrew D. Bagdanov, Simone Calderara, and Joost van de Weijer. Accurate and efficient low-rank model merging in core space. In *Advances in Neural Information Processing Systems*, 2025.

[198] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[199] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context, 2016.

[200] Chanho Park, H Vincent Poor, and Namyoon Lee. Signsgd with federated voting. *arXiv preprint arXiv:2403.16372*, 2024.

[201] Jaeyoo Park, Minsoo Kang, and Bohyung Han. Class-incremental learning for action recognition in videos. In *IEEE International Conference on Computer Vision*, 2021.

[202] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. FiLM: visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.

[203] Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. Modular deep learning. *Transactions on Machine Learning Research*, 2023.

[204] Angelo Porrello, Stefano Vincenzi, Pietro Buzzega, Simone Calderara, Annamaria Conte, Carla Ippoliti, Luca Candeloro, Alessio Di Lorenzo, and Andrea Capobianco Dondona. Spotting insects from satellites: modeling the presence of culicoides imicola through deep cnns. In *International Conference on Signal-Image Technology & Internet-Based Systems*, 2019.

[205] Angelo Porrello, Lorenzo Bonicelli, Pietro Buzzega, Monica Millunzi, Simone Calderara, and Rita Cucchiara. A second-order perspective on model compositionality and incremental learning. In *International Conference on Learning Representations*, 2025.

[206] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. GDumb: A simple approach that questions our progress in Continual Learning. In *European Conference on Computer Vision*, 2020.

[207] Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. Motiontrack: Learning robust short-term and long-term motions for multi-object tracking. In *IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[208] Zheng Qin, Le Wang, Sanping Zhou, Panpan Fu, Gang Hua, and Wei Tang. Towards generalizable multi-object tracking. In *IEEE conference on Computer Vision and Pattern Recognition*, 2024.

[209] Helei Qiu, Biao Hou, Bo Ren, and Xiaohua Zhang. Spatio-temporal tuples transformer for skeleton-based action recognition. *arXiv preprint arXiv:2201.02849*, 2022.

[210] 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 *International Conference on Machine Learning*, 2021.

[211] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21, 2020.

[212] Jathushan Rajasegaran, Georgios Pavlakos, Angjoo Kanazawa, and Jitendra Malik. Tracking people by predicting 3d appearance, location and pose. In *IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[213] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[214] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE International Conference on Computer Vision*, 2021.

[215] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[216] 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.

[217] Mahdi Rezaei, Mutsuhiro Terauchi, and Reinhard Klette. Robust vehicle detection and distance estimation under challenging lighting conditions. *IEEE transactions on intelligent transportation systems*, 2015.

[218] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

[219] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. In *International Conference on Learning Representations*, 2019.

[220] Filippo Rinaldi, Giacomo Capitani, Lorenzo Bonicelli, Donato Crisostomi, Federico Bolelli, Elisa Ficarra, Emanuele Rodolà, Simone Calderara, and Angelo Porrello. Update your transformer to the latest release: Re-basin of task vectors. In *International Conference on Machine Learning*, 2025.

[221] Filippo Rinaldi, Aniello Panariello, Giacomo Salici, Fengyuan Liu, Marco Ciccone, Angelo Porrello, and Simone Calderara. Gradient-sign masking for task vector transport across pre-trained models. *International Conference on Learning Representations*, 2025.

[222] Filippo Rinaldi, Aniello Panariello, Giacomo Salici, Angelo Porrello, and Simone Calderara. Transporting task vectors across different architectures without training. *arXiv preprint arXiv: 2602.12952*, 2026.

[223] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, 2016.

[224] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 1995.

[225] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv:1606.04671*, 2016.

[226] Adrian Sanchez-Caballero, David Fuentes-Jimenez, and Cristina Losada-Gutiérrez. Exploiting the convlstm: Human action recognition using raw depth video-based recurrent neural networks. *arXiv preprint arXiv:2006.07744*, 2020.

[227] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, 1999.

[228] Christoph Schöller and Alois Knoll. Flomo: Tractable motion prediction with normalizing flows. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[229] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 2020.

[230] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, 2018.

[231] Mattia Segu, Bernt Schiele, and Fisher Yu. Darth: Holistic test-time adaptation for multiple object tracking. In *IEEE International Conference on Computer Vision*, 2023.

[232] Jenny Seidenschwarz, Guillem Brasó, Victor Castro Serrano, Ismail Elezi, and Laura Leal-Taixé. Simple cues lead to a strong multi-object tracker. In *IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[233] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *IEEE conference on Computer Vision and Pattern Recognition*, 2016.

[234] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[235] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[236] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[237] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[238] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *IEEE conference on Computer Vision and Pattern Recognition*, 2016.

[239] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *European Conference on Computer Vision*, 2018.

[240] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *European Conference on Computer Vision*, 2020.

[241] Bing Shuai, Andrew Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe. Siammot: Siamese multi-object tracking. In *IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[242] Bing Shuai, Alessandro Bergamo, Uta Buechler, Andrew Berneshawi, Alyssa Boden, and Joseph Tighe. Large scale real-world multi-person tracking. In *European Conference on Computer Vision*, 2022.

[243] Bing Shuai, Xinyu Li, Kaustav Kundu, and Joseph Tighe. Id-free person similarity learning. In *IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[244] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[245] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[246] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020.

[247] George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. In *International Conference on Learning Representations*, 2025.

[248] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[249] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[250] Zehua Sun, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. Human action recognition from various data modalities: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[251] A. Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu, Bo Du, Yixin Chen, and Dacheng Tao. Parameter efficient multi-task model fusion with partial linearization. In *International Conference on Learning Representations*, 2023.

[252] Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner. *arXiv:2210.03114*, 2022.

[253] Yu Tian, Guansong Pang, Yuanhong Chen, Rajvinder Singh, Johan W Verjans, and Gustavo Carneiro. Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In *IEEE International Conference on Computer Vision*, 2021.

[254] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, 2018.

[255] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022.

[256] Kim Tran, Anh Duy Le Dinh, Tien-Phat Nguyen, Thinh Phan, Pha Nguyen, Khoa Luu, Donald Adjeroh, Gianfranco Doretto, and Ngan Le. Z-GMOT: Zero-shot generic multiple object tracking. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2024.

[257] Shane Tuohy, Diarmaid O'Cualain, Edward Jones, and Martin Glavin. Distance determination for an automobile environment using inverse perspective mapping in opencv. In *IET Irish Signals and Systems Conference*, 2010.

[258] Gido M van de Ven and Andreas S Tolias. Three Continual Learning scenarios. In *Neural Information Processing Systems Workshops*, 2018.

[259] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017.

[260] 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.

[261] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[262] Andrés Villa, Juan León Alcázar, Motasem Alfarra, Kumail Alhamoud, Julio Hurtado, Fabian Caba Heilbron, Alvaro Soto, and Bernard Ghanem. Pivot: Prompting for video continual learning. In *IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[263] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

[264] Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jiménez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. In *International Conference on Machine Learning*, 2024.

[265] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[266] Runqi Wang, Xiaoyue Duan, Guoliang Kang, Jianzhuang Liu, Shaohui Lin, Songcen Xu, Jinhu Lü, and Baochang Zhang. Attriclip: A non-incremental learner for incremental knowledge learning. In *IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[267] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[268] Xuanzhao Wang, Zhengping Che, Bo Jiang, Ning Xiao, Ke Yang, Jian Tang, Jieping Ye, Jingyu Wang, and Qi Qi. Robust unsupervised video anomaly detection by multipath frame prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[269] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, 2022.

[270] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[271] Zirui Wang, Zihang Dai, B. Póczos, and J. Carbonell. Characterizing and avoiding negative transfer. In *IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[272] Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. On negative interference in multilingual models: Findings and A meta-learning treatment. In *Empirical Methods in Natural Language Processing*, 2020.

[273] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. In *Advances in Neural Information Processing Systems Workshops*, 2017.

[274] Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

[275] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *IEEE International Conference on Image Processing*. IEEE, 2017.

[276] 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 *IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[277] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[278] Jie Wu, Wei Zhang, Guanbin Li, Wenhao Wu, Xiao Tan, Yingying Li, Errui Ding, and Liang Lin. Weakly-supervised spatio-temporal anomaly detection in surveillance video. In *International Joint Conference on Artificial Intelligence*, 2021.

[279] Peng Wu, Jing Liu, Yujia Shi, Yujia Sun, Fangtao Shao, Zhaoyang Wu, and Zhiwei Yang. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *European Conference on Computer Vision*, 2020.

[280] Wenhan Wu, Yilei Hua, Ce Zheng, Shiqian Wu, Chen Chen, and Aidong Lu. Skeletonmae: Spatial-temporal masked autoencoders for self-supervised skeleton action recognition. In *IEEE International Conference on Multimedia and Expo Workshops*, 2023.

[281] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liangbo He. A survey of human-in-the-loop for machine learning. *Future generations computer systems*, 2021.

[282] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, 2020.

[283] Wentian Xin, Ruyi Liu, Yi Liu, Yu Chen, Wenxin Yu, and Qiguang Miao. Transformer for skeleton-based action recognition: A review of recent advances. *Neurocomputing*, 2023.

[284] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *IEEE International Conference on Computer Vision*, 2017.

[285] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

[286] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *IEEE International Conference on Computer Vision*, 2021.

[287] Hong Yan, Yang Liu, Yushen Wei, Zhen Li, Guanbin Li, and Liang Lin. Skeletonmae: graph-based masked autoencoder for skeleton sequence pre-training. In *IEEE International Conference on Computer Vision*, 2023.

[288] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, 2018.

[289] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[290] Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *ACM Computing Surveys*, 2026.

[291] Le Yang, Houwen Peng, Dingwen Zhang, Jianlong Fu, and Junwei Han. Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing*, 2020.

[292] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv: 2406.09414*, 2024.

[293] Jun Yin, Jun Han, Chenghao Wang, Bingyi Zhang, and Xiaoyang Zeng. A skeleton-based action recognition system for medical condition detection. In *IEEE Biomedical Circuits and Systems Conference*, 2019.

[294] En Yu, Songtao Liu, Zhuoling Li, Jinrong Yang, Zeming Li, Shoudong Han, and Wenbing Tao. Generalizing multiple object tracking to unseen domains by introducing natural language representation. In *AAAI Conference on Artificial Intelligence*, 2023.

[295] En Yu, Tiancai Wang, Zhuoling Li, Yuang Zhang, Xiangyu Zhang, and Wenbing Tao. Motrv3: Release-fetch supervision for end-to-end multi-object tracking. *arXiv preprint arXiv:2305.14298*, 2023.

[296] Guang Yu, Siqi Wang, Zhiping Cai, En Zhu, Chuanfu Xu, Jianping Yin, and Marius Kloft. Cloze test helps: Effective video anomaly detection via learning to complete video events. In *ACM International Conference on Multimedia*, 2020.

[297] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *IEEE conference on Computer Vision and Pattern Recognition*, 2024.

[298] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*, 2024.

[299] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[300] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *IEEE conference on Computer Vision and Pattern Recognition*, 2015.

[301] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *European Conference on Computer Vision*, 2022.

[302] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *IEEE International Conference on Computer Vision*, 2019.

[303] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 2017.

[304] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE International Conference on Computer Vision*, 2023.

[305] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: slow learner with classifier alignment for continual learning on a pre-trained model. In *IEEE International Conference on Computer Vision*, 2023.

[306] Jianyu Zhang and L. Bottou. Learning useful representations for shifting tasks and distributions. In *International Conference on Machine Learning*, 2022.

[307] Jinghan Zhang, Junteng Liu, Junxian He, et al. Composing parameter-efficient modules with arithmetic operation. In *Advances in Neural Information Processing Systems*, 2023.

[308] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023.

[309] Songyang Zhang, Yang Yang, Jun Xiao, Xiaoming Liu, Yi Yang, Di Xie, and Yueting Zhuang. Fusing geometric features for skeleton-based action recognition using multilayer lstm networks. *IEEE Transactions on Multimedia*, 2018.

[310] Wenwei Zhang, Hui Zhou, Shuyang Sun, Zhe Wang, Jianping Shi, and Chen Change Loy. Robust multi-modality multi-object tracking. In *IEEE International Conference on Computer Vision*, 2019.

[311] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021.

[312] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*, 2022.

[313] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[314] Zangwei Zheng, Mingyu Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *IEEE International Conference on Computer Vision*, 2023.

[315] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE conference on Computer Vision and Pattern Recognition*, 2016.

[316] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[317] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 2022.

[318] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, 2020.

[319] Jing Zhu and Yi Fang. Learning object-specific distance from a monocular image. In *IEEE International Conference on Computer Vision*, 2019.

[320] Yi Zhu and Shawn Newsam. Motion-aware feature for improved video anomaly detection. In *British Machine Vision Conference*, 2019.