UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

UMONS
Université de Mons

Inductive and Transductive Learning with Vision-Language Models in Low-Shot Regimes

Maxime Zanella

Thesis submitted in fulfillment
of the requirements for the degree of
Docteur en sciences de l'ingénieur et technologie

Dissertation committee:

Prof. Benoît Macq (UCLouvain, Belgium) – Advisor
Prof. Saïd Mahmoudi (UMons, Belgium) – Advisor
Prof. Christophe Craeye (UCLouvain, Belgium) – Chair
Prof. Bernard Gosselin (UMons, Belgium) – Secretary
Prof. Christophe De Vleeschouwer (UCLouvain, Belgium)
Prof. Pierre Geurts (ULiège, Belgium)
Prof. Ismail Ben Ayed (ÉTS Montréal, Canada)

Acknowledgments

I want to express my deepest gratitude to my advisors, Benoît Macq and Saïd Mahmoudi, for giving me the chance to embark on this PhD in the first place. I was initially hesitant to take this path, but Benoît possesses a unique talent for persuasion, and I am glad now that he does. It sparked what became a truly great adventure. I deeply appreciate the absolute freedom and independence I was given to conduct my research. To both Saïd and Benoît: thank you for standing by me, even when my work completely shifted away from our initial plans for active learning and mammography. Your trust in my vision were a driving force throughout this entire journey.

I am also profoundly thankful to Professor Ismail Ben Ayed, who so warmly welcomed me to ETS Montréal in Canada. He opened my eyes to the fascinating world of vision-language models, zero-shot and few-shot learning. It was a turning point. This incredible opportunity deeply impacted me and permanently shaped the research direction of this thesis. Thank you, Ismail, not just for your invaluable professional guidance, but for your welcoming and approachable nature. My sincere thanks also go to Christophe De Vleeschouwer, with whom I had the joy of collaborating on several publications. His enthusiasm is highly contagious, and I truly appreciated the brilliant, easygoing, and motivating dynamic he brought to our work.

To the members of my jury: thank you for dedicating your time and expertise to evaluate this thesis.

Furthermore, a huge thank you goes to the fantastic colleagues who were by my side. You enriched these past few years immensely, offering support that went far beyond just academic matters. To Karim, Dani, Pierre L., Antoine, Manon, Tiffanie, Nicolas D., Benoît, Mélanie, Simon, Maxence, Clément, Gauthier, Pierre R., Jonathan, Edouard, Colin, Mathias, Pauline,

Tim, Estelle, Vladimir, Tim, Corentin, Adrien, Arthur, Jérôme, Lucas, Nicolas B., Damien, Dany, Eliott, Samuel, and so many others: thank you for the countless conversations and the great times we shared at the Pilab and elsewhere. Whether we were deep in research, sharing endless delicious coffee and lunch (at 12h!) breaks, conference trips, or just chatting about work and life (with or without a well-deserved drink in hand), every single one of you played a part in this story. With some, I collaborated intensely; with others, I simply enjoyed much-needed moments of escape (and sometimes, both at the same time). You all shaped this journey in your own unique ways. Some of you became far more than just colleagues, and I am deeply grateful for the lasting friendships we forged.

I would also like to extend my gratitude to all my co-authors and collaborators inside and outside of UCLouvain and UMons. In particular, I want to thank the researchers from ULiège with whom I had the opportunity to work as part of the TRAIL initiative. Furthermore, a special thank you goes to everyone at the LIVIA lab at ETS Montréal. Your incredibly warm welcome made the few months I spent in Montreal an unforgettable experience.

Last but certainly not least, my heartfelt gratitude goes out to my close friends and family. You made this demanding journey so much easier to navigate through your endless support and encouragement. Finally, I want to end by thanking my mother and my father, who have always believed in me. Thank you for giving me the unconditional freedom to choose the path in life that truly suits me best.

Table 1 Author's Publication list. Topics legend: natural images (*Nat*), computational pathology (*Path*), satellite imagery (*RS*), other medical imagery (*Med*), audio (*Aud*). ★ denotes core contributions, * shared first authorship. Affiliations: [1] UCLouvain, [2] UMons, [3] ÉTS Montréal, [4] ULg, [5] UdeM, [6] ULB.

Id	Paper (Author names, *title*)	Venue	Organism	(Co-) First Author	*Nat*	*Path*	*RS*	*Med*	*Aud*
[1]	★ **M. Zanella** & I. Ben Ayed, *Low-Rank Few-Shot Adaptation of Vision-Language Models*.	CVPRW'24	1,2,3	†	✓				
[2]	M. Dausort*, T. Godelaine*, **M. Zanella**, K. El Khoury, I. Salmon, and B. Macq, *Exploring Foundation Models Fine-Tuning for Cytology Classification*.	ISBI'25.	1,2,6			✓			
[3]	K. El Khoury, **M. Zanella**, C. De Vleeschouwer, and B. Macq, *Few-Shot Adaptation Benchmark for Remote Sensing Vision-Language Models*.	arXiv'25	1,2				✓		
[4]	★ **M. Zanella** & I. Ben Ayed, *On the test-time zero-shot generalization of vision-language models: Do we really need prompt learning?*.	CVPR'24	1,2,3	†	✓				
[5]	★ **M. Zanella***, B. Gérin*, and I. Ben Ayed, *Boosting Vision-Language Models with Transduction*.	NeurIPS'24	1,2,3	†	✓				
[6]	**M. Zanella***, F. Shakeri*, Y. Huang, H. Bahig, and I. Ben Ayed, *Boosting vision-language models for histopathology classification: Predict all at once*.	MedAGI'24	1,2,3,5	†		✓			
[7]	K. El Khoury*, **M. Zanella***, B. Gérin*, T. Godelaine*, B. Macq, S. Mahmoudi, C. De Vleeschouwer, and I. Ben Ayed, *Enhancing remote sensing vision-language models for zero-shot scene classification*.	ICASSP'25	1,2,3	†			✓		
[8]	★ **M. Zanella***, C. Fuchs*, C. De Vleeschouwer, and I. Ben Ayed, *Realistic Test-Time Adaptation of Vision-Language Models*.	CVPR'25.	1,2,3	†	✓				
[9]	**M. Zanella***, C. Fuchs*, I. Ben Ayed, and C. De Vleeschouwer, *Vocabulary-free few-shot learning for vision-language models*.	CVPRW'25	1,2,3	†	✓				
[10]	G. Baklouti, **M. Zanella**, and I. Ben Ayed, *Language-Aware Information Maximization for Transductive Few-Shot CLIP*.	under review	1,2,3		✓				
[11]	F. Shakeri*, G. Baklouti*, J. Silva-Rodríguez, **M. Zanella**, H. Bahig, J. Dolz, and I. Ben Ayed, *Test-time adaptation of medical vision-language models*.	MedAGI'25	1,2,3,5			✓		✓	
[12]	C. Fuchs*, **M. Zanella***, and C. De Vleeschouwer, *Online Gaussian Adaptation of Vision-Language Models*.	CVPRW'25	1,2	†	✓				
[13]	B. Gérin*, **M. Zanella***, M. Wynen, S. Mahmoudi, B. Macq, and C. De Vleeschouwer, *Exploring viability of test-time training: Application to 3D segmentation in multiple sclerosis*.	CAI'24	1,2	†				✓	
[14]	S. Piérard, A. Cioppa, A. Halin, R. Vandeghen, **M. Zanella**, B. Macq, S. Mahmoudi, and M. Van Droogenbroeck, *Mixture domain adaptation to improve semantic segmentation in real-world surveillance*.	WACVW'23	1,2,4		✓				
[15]	A. Halin*, S. Piérard*, R. Vandeghen, B. Gérin, **M. Zanella**, M. Colot, J. Held, A. Cioppa, E. Jean, G. Bontempi, S. Mahmoudi, B. Macq, and M. Van Droogenbroeck, *Physically interpretable probabilistic domain characterization*.	ACCVW'24	1,2,4,6		✓				
-	T. Godelaine*, **M. Zanella***, K. El Khoury, S. Mahmoudi, B. Macq, and C. De Vleeschouwer, *Conditional Random Fields for Interactive Refinement of Histopathological Predictions*.	under review	1,2	†		✓			
-	K. El Khoury*, **M. Zanella***, T. Godelaine*, C. De Vleeschouwer, and B. Macq, *Leveraging Prediction Entropy for Automatic Prompt Weighting in Zero-Shot Audio-Language Classification*.	under review	1,2	†					✓
-	C. Fuchs, **M. Zanella**, B. Macq, and C. De Vleeschouwer, *Does base-to-novel few-shot learning for medical vision-language models actually work? A call for rigorous evaluation*.	under review	1,2					✓	

Abstract

Vision-Language Models (VLMs) have revolutionized open-vocabulary recognition, yet adapting them to specialized domains under data-scarce conditions remains a significant challenge. This thesis investigates efficient strategies to bridge the gap between *inductive* fine-tuning and *transductive* inference, focusing on minimizing supervision and computational costs.

We first address inductive few-shot learning with CLIP-LoRA, a parameter-efficient fine-tuning method. By injecting trainable low-rank matrices into the encoder, it outperforms existing approaches while eliminating dataset-specific hyperparameter tuning. For scenarios restricted to single-sample inference, we propose MTA (MeanShift for Test-time Augmentation). Formulated as a robust variation of the MeanShift algorithm, this training-free method refines predictions through iterative mode-seeking and automatic importance weighting of augmented views.

Expanding into the transductive paradigm, we introduce TransCLIP. This framework explicitly models the underlying structure of unlabeled test batches by fitting a Gaussian Mixture Model (GMM), constrained by a text-driven Kullback-Leibler (KL) regularization term. This dual objective dynamically aligns visual statistics with textual priors, achieving substantial gains in both zero-shot and few-shot settings. Finally, to address realistic deployment challenges, we propose Stat$\mathcal{A}$ (Statistical Anchor). We demonstrate that existing transductive methods fail under non-i.i.d. data streams and introduce a regularization technique based on the KL divergence with respect to initial text priors.

Collectively, this work provides a suite of transparent, low-compute algorithms that, through the use of fixed hyperparameters, ensure operational simplicity and robust performance across challenging and diverse benchmarks.

Contents

1

Introduction

For the past decade, deep learning has redefined the landscape of computer vision, yet until very recently models were constrained in a *closed-world* paradigm. They were typically trained to recognize a predefined set of categories, but would remain very limited when used outside their fixed ontology. The introduction of *Vision-Language Models* (VLMs), and most notably the *Contrastive Language-Image Pre-training* (CLIP) framework has profoundly changed this aspect, marking a true revolution in how vision models are pre-trained and consequently, how they encode information about the visual world. Instead of relying on labor-intensive, crowd-sourced labels, which can quickly act as a bottleneck to scalability in data but also the number of concepts presented, CLIP leverages *natural language supervision*, learning from a massive scale of 400 million image-text pairs naturally available on the web (e.g., pictures along with their descriptive captions). The core idea is simple yet powerful: the system uses two distinct models (one per modality) that represent their respective input by vectors (i.e., embeddings); during training, the vector representation of an image is pulled closer to the vector of its matching text description, teaching the model to understand images through the lens of language. This paradigm shift transforms the model from a specialized classifier into a robust *generalist*: it enables *open-vocabulary* recognition, allowing the model to identify visual concepts it was never explicitly asked to learn by simply matching images to arbitrary descriptions. For instance, to classify a

"sleepy cat" (a really important task for which we did not collect annotated data and the model was not specifically trained) it simply determines whether the photo vector is closer to the encoded text "a photo of a sleepy cat" or "a photo of a moving cat," effectively using natural language as a flexible classification tool. This foundational capability to generalize to new tasks without retraining known as *zero-shot* transfer has fundamentally changed the research trajectory in computer vision, establishing these models as flexible foundations upon which diverse downstream applications can be built, as we will see throughout this thesis.

However, the capabilities of these generalist models are not without limitations. While they offer a strong starting point, their zero-shot performance is not guaranteed to translate effectively to every downstream task. When facing significant domain shifts, such as analyzing satellite imagery or diagnosing pathologies in medical slides, two application domains studied in this thesis alongside more traditional benchmarks, the initial predictions often fall short as we will show later. In traditional deep learning, the standard solution would be to collect a massive dataset and train a new model; yet, in many real-world scenarios, acquiring such large-scale annotations is practically impossible. This creates a critical bottleneck: we rarely have the luxury of abundant supervision, particularly in high-stakes applications like identifying rare diseases or assessing damage from natural disasters in remote sensing imagery. Fortunately, research has shown that the powerful representations of these models often require only minimal interventions to unlock great capabilities in new domains. As a consequence, the core challenge has shifted from training to *adaptation* specifically, from learning with little labeled data (i.e., *few-shot learning*) to learning with no labels at all (i.e., *zero-shot learning*). These two learning paradigms are central to this thesis, with specific chapters dedicated to addressing the unique constraints of each.

A review of the literature on adapting contrastive Vision-Language Models (VLMs) in both few-shot and zero-shot learning reveals a clear pattern: the vast majority of research efforts are concentrated on the *inductive* paradigm. In this standard setting, adaptation is treated as a distinct phase: we first update the model parameters using the few available labeled examples (if any), and subsequently process each test image in isolation. For instance, when classifying a satellite image of a forest, an inductive model relies solely on its internal knowledge to make a decision, ignoring poten-

tially helpful context from other images. While this approach has driven recent progress, it essentially leaves an important source of information untapped: the test data itself. Unlike induction, *transduction* does not treat test samples as isolated queries. Instead, it asks what can be gained by analyzing the structure of the unlabeled test data as a whole. In the previous example, a transductive approach could notice that the ambiguous forest image shares similar geographic features with other clear examples of "rainforest", and use this to correct its prediction. While leveraging such data structures has proven powerful in traditional computer vision, its direct application to VLMs has been largely underexplored or ineffective. Addressing the gap between inductive and transductive approaches defines a critical objective of this thesis.

Vision-Language Models are fundamentally reshaping the traditional machine learning pipeline into three critical, yet often disjoint, stages: *pre-training* (e.g., CLIP), *fine-tuning* (e.g., few-shot learning), and *inference* (e.g., with transduction). While massive pre-training provides a powerful open-vocabulary foundation, the two subsequent steps remain fragmented. Inductive fine-tuning is necessary to specialize models for data-scarce domains, while transductive methods are essential to unlock the latent potential of test data at deployment. From these key steps, we propose a comprehensive roadmap that bridges the gap between inductive fine-tuning and transductive inference, articulated around efficiency without sacrificing performance. We begin this journey by introducing **CLIP-LoRA** (Chapter 3), a parameter-efficient fine-tuning method for *inductive* few-shot learning. This provides great insights to better comprehend these VLMs and the challenges addressed in subsequent chapters. Moving to the inference stage, we present **MTA** (Chapter 4), a lightweight, training-free algorithm for robust single-sample adaptation. This approach serves as a critical step in this work, demonstrating that a smartly designed, training-free mechanism can outperform heavy optimization procedures. We then shift to the transductive paradigm with **TransCLIP** (Chapter 5), a novel framework that leverages batch statistics and textual priors to enhance performance even without additional supervision, showcasing the great potential of transduction in this new multi-modal era. Finally, we tackle deployment *in the wild* with **StatA** (Chapter 6), a robust transductive method designed to handle realistic, non-ideal data distributions.

1.1 Motivation and Research Questions

This thesis presents four core contributions within the theme of *inductive* and *transductive* learning for *contrastive vision-language models*. For readers unfamiliar with these concepts, Chapter 2 provides the necessary background and definitions.

This section is articulated around the fundamental research questions that motivated this work and the contributions that emerged from addressing them. A comprehensive summary of the publications resulting from this research is provided in Table 1, and a general overview of the thesis roadmap is illustrated in Figure 1.1.

The challenge of learning with little to no data is vast, spanning everything from active, unsupervised, semi-supervised, self-supervised learning to few-shot, continual, and test-time adaptation. With such a rapidly expanding field, it is easy to get lost. While it is tempting to try and tackle all these problems at once, we realized that maintaining a clear research trajectory required us to narrow our focus. A single guiding theme runs through this entire thesis: the pursuit of *efficiency* without sacrificing performance. When we say efficiency, we mean it in three specific pillars:

(i) *Supervision scarcity:* We aimed to maximize performance while minimizing reliance on large-scale manual annotation. Whether adapting to new tasks with just a handful of examples (inductive few-shot learning) or exploiting the intrinsic structure of unlabeled data (transductive zero-shot learning), our objective was to propose high-performing procedures that could fit a large variety of scenarios.

(ii) *Hyperparameter robustness:* we wanted methods that work as plug-and-play tools. Tuning hyperparameters is difficult enough, but it becomes nearly impossible when evaluating across diverse benchmarks with little (labeled) data. Therefore, we prioritized methods with fixed or very limited hyperparameters.

(iii) *Computational viability*: we focused on lightweight algorithms. The goal was to develop methods that are computationally friendly yet can still compete with, or even outperform, much more training-intensive approaches.

These three principles shaped every contribution of this thesis, though the balance between them shifts from chapter to chapter. Initially, they drove the research presented in Chapters 3 and 4, where the goal was to achieve efficient adaptation within the inductive framework. Building on this foundation, we extended the scope of this research to the transductive paradigm, which had been largely overlooked in the VLM literature. Consequently, Chapters 5 and 6 apply the same rigorous standards of computational viability and hyperparameter robustness to the challenge of leveraging unlabeled test data.

It is from this perspective that we identified the gaps we aimed to fill, leading to the following set of research questions that guide the remainder of this work:

1. **Inductive Few-Shot Learning:** Inductive few-shot learning aims to adapt pre-trained models to new visual categories using only a handful of labeled examples per class (typically between 1 and 16). The advent of Vision-Language Models (VLMs) has significantly advanced this field, pushing the boundaries of generalization with minimal supervision. However, this promising, already quite abundant literature has focused principally on prompt learning and, to a lesser extent, on adapters. Furthermore, existing few-shot learning methods for VLMs often rely on computationally intensive training procedures and/or carefully chosen, task-specific hyper-parameters, which might impede their applicability. This led to the following research question: *Can we develop few-shot adaptation methods that are simultaneously high-performing, computationally efficient, and straightforward to deploy across diverse tasks without requiring per-task hyperparameter customization?*
2. **Single Sample Test-Time Adaptation:** The development of large vision-language models, notably CLIP, has catalyzed research into effective adaptation techniques, with a particular focus on soft prompt tuning, for both zero- and few-shot learning. Conjointly, test-time augmentation, which utilizes multiple augmented views of a single image to enhance zero-shot generalization, is emerging as a significant area of interest. This has predominantly directed research efforts toward test-time prompt tuning, and its inherent high computational cost. This observation led to the following research question: *Leveraging multimodal information (visual embeddings and text priors), how can a*

computationally efficient, training-free mechanism robustly refine a single-sample prediction without performing any parameter updates?

3. **Transductive Learning for Vision-Language Models:** Transduction is a powerful paradigm that leverages the structure of unlabeled data to boost predictive accuracy. Within deep learning, transduction has primarily been explored in few-shot learning to address the challenges of training with limited supervision, focusing on standard vision-only pre-trained models. However, the direct application of existing transductive few-shot methods to VLMs yields poor performance, sometimes even underperforming relative to inductive zero-shot predictions. This performance drop underscores the need for novel, VLM-native transductive methods. This challenge motivates the following research question: *How can the transductive paradigm be rethought for Vision-Language Models by integrating textual knowledge, for both zero-shot and few-shot classification?*
4. **Transductive Learning in the Wild:** The current test-time adaptation literature addresses adaptation in somewhat idealized settings. While convenient for benchmarking and fast comparison to previous methods, these scenarios often lack the challenging conditions found in real-world applications. For example, many transductive methods assume that test data is well-behaved (e.g., that all classes are present or that data arrives in i.i.d. streams). Such assumptions are often violated in practice. This discrepancy reveals a critical research gap and motivates our final research question: *How can transductive methods for Vision-Language Models be designed to remain robust and reliable when deployed "in the wild"—that is, under realistic conditions, where strong assumptions about the data distribution do not hold?*

1.2 Core Contributions

This thesis presents four core technical contributions, which are detailed across four dedicated chapters. The first three chapters establish strong baselines on conventional benchmarks, beginning with an efficient method for inductive few-shot learning in Chapter 3. The work then progresses to adaptation with unlabeled test data, first by proposing a lightweight method for single-image adaptation in Chapter 4, followed by a novel

framework for transductive learning for VLMs in Chapter 5. The final technical chapter shifts focus to "learning in the wild," tackling the challenges of realistic deployment scenarios and proposing a robust method to handle them in Chapter 6. Lastly, Chapter 7 adopts a more exploratory stance, offering a unifying view of the presented methods, alongside future perspectives on the field and concluding remarks.

Concretely, these core contributions are: (i) **CLIP-LoRA**, an efficient inductive few-shot learning method; (ii) **MTA**, a lightweight adaptation method on a single image; (iii) **TransCLIP**, a novel approach for transductive learning with VLMs; (iv) **Stat**$\mathcal{A}$, a robust transductive learning method designed for challenging data distributions. These primary contributions led to several publications, which are detailed below.

1. **CLIP-LoRA (Chapter 3):**

    - Maxime Zanella, Ismail Ben Ayed, *Low-Rank Few-Shot Adaptation of Vision-Language*, IEEE / CVF Computer Vision and Pattern Recognition Conference Workshops (CVPRW) [1]
    - Manon Dausort, Tiffanie Godelaine, Maxime Zanella, Karim El Khoury, Isabelle Salmon, Benoît Macq, *Exploring Foundation Models Fine-Tuning for Cytology Classification*, IEEE International Symposium on Biomedical Imaging (ISBI) [2]
    - Karim El Khoury, Maxime Zanella, Benoît Macq, Christophe De Vleeschouwer, *Few-Shot Adaptation Benchmark for Remote Sensing Vision-Language Models*, [Preprint, Under Review] [3]

    Summary: We introduce Low-Rank Adaptation (LoRA) in few-shot learning for VLMs, and show its potential in comparison to current state-of-the-art prompt- and adapter-based approaches. Surprisingly, our simple CLIP-LoRA method exhibits substantial improvements, while reducing the training times and keeping the same hyperparameters in all the target tasks, i.e., across all the datasets and numbers of shots. Our method stands as a strong baseline and our codebase can be used to evaluate progress in more fair scenarios in few-shot VLMs. We further assess Low-Rank Adaptation of VLMs in the context of cytology and remote sensing scene classification demonstrating its wide applicability.

2. **MTA (Chapter 4)**

    - Maxime Zanella, Ismail Ben Ayed, *On the Test-Time Zero-Shot Generalization of Vision-Language Models: Do We Really Need Prompt Learning?*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) [4]

    Summary: This work explores test-time augmentation which is a technique that generates multiple views of an image to enhance generalization. Current VLM approaches rely on test-time prompt tuning, an intensive procedure that is impractical for API-based models. In contrast, we introduce MeanShift for Test-time Augmentation (MTA), a method that directly refines the image's visual representation instead of the textual prompts. MTA is a robust, training-free, and hyperparameter-free algorithm that uses a mode-seeking process to jointly optimize an *inlierness* score for each augmented view, automatically down-weighting outliers without ad-hoc thresholds. This efficient, black-box approach functions as a simple plug-and-play module, demonstrating superior performance and consistent improvements when applied on top of both zero-shot and few-shot models.

3. **TransCLIP (Chapter 5):**

    - Maxime Zanella, Benoît Gérin, Ismail Ben Ayed, *Boosting Vision-Language Models with Transduction*, Advances in Neural Information Processing Systems (NeurIPS) [5]
    - Maxime Zanella, Fereshteh Shakeri, Yunshi Huang, Houda Bahig, Ismail Ben Ayed, *Boosting Vision-Language Models for Histopathology Classification: Predict All at Once*, Int. Workshop on Foundation Models for General Medical AI (MedAGI) [6]
    - Karim El Khoury, Maxime Zanella, Benoît Gérin, Tiffanie Godelaine, Benoît Macq, Saïd Mahmoudi, Christophe De Vleeschouwer, Christophe, Ismail Ben Ayed, *Enhancing Remote Sensing Vision-Language Models for Zero-Shot Scene Classification*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) [7]

    Summary: We present TransCLIP, a novel and computationally efficient transductive approach designed for Vision-Language Mod-

els (VLMs). TransCLIP is applicable as a plug-and-play module on top of popular inductive zero- and few-shot models, consistently improving their performance. Our new objective function can be viewed as a regularized maximum-likelihood estimation, constrained by a KL divergence penalty that integrates the text-encoder knowledge and guides the transductive learning process. We further derive an iterative Block Majorize-Minimize (BMM) procedure for optimizing our objective, with guaranteed convergence and decoupled sample-assignment updates, yielding computationally efficient transduction for large-scale datasets. We further validate TransCLIP on high-stakes downstream applications: histopathology diagnosis and remote sensing scene classification. These extensive experiments confirm that transduction is not merely effective on standard datasets, but can serve as a versatile and reliable tool for specialized domains.

4. **StatA (Chapter 6):**

   - Maxime Zanella, Clément Fuchs, Christophe De Vleeschouwer, Ismail Ben Ayed, *Realistic Test-Time Adaptation of Vision-Language Models*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) [8]

   Summary: While Test-Time Adaptation (TTA) methods show promise, they often rely on unrealistic assumptions about the test data, such as the presence of all classes or i.i.d. data streams. Our work challenges these favorable deployment scenarios and introduces a more realistic evaluation framework, including (i) a variable number of effective classes for adaptation within a single batch, and (ii) non-i.i.d. batches of test samples in online adaptation settings. We demonstrate how current transductive or TTA methods for VLMs systematically compromise the models' initial zero-shot robustness across various realistic scenarios, favoring performance gains under advantageous assumptions about the test sample distributions. Furthermore, we introduce Stat$\mathcal{A}$, a versatile method designed to handle such challenging deployments, including batches with a variable number of effective classes. Its core is a novel regularization term, the Statistical Anchor, which preserves the initial text-encoder knowledge. This anchor makes StatA robust, particularly in the low-data or imbalanced regimes common in real-world applications.

1.3 Complementary Contributions

Beyond the core contributions detailed above, this body of work encompasses additional research efforts that align closely with the central thesis but excluded from it for brevity.

- **SiM (Chapter 3):**
  - Maxime Zanella, Clément Fuchs, Ismail Ben Ayed, Christophe De Vleeschouwer, *Vocabulary-free few-shot learning for vision-language models*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) [9]

  Summary: Recent advances in few-shot adaptation for Vision-Language Models (VLMs) have greatly expanded their ability to generalize across tasks using only a few labeled examples. However, existing approaches primarily build upon the strong zero-shot priors of these models by leveraging carefully designed, task-specific prompts. This dependence on predefined class names can restrict their applicability, especially in scenarios where exact class names are unavailable or difficult to specify. To address this limitation, we introduce vocabulary-free few-shot learning for VLMs, a setting where target class images are available but their corresponding names are not. We propose Similarity Mapping (SiM), a simple yet effective baseline that classifies target instances solely based on similarity scores with a set of generic prompts (textual or visual), eliminating the need for carefully handcrafted prompts. SiM demonstrates strong performance, operates with high computational efficiency, and provides interpretability by linking target classes to generic prompts.

- **LIMO (Chapter 5):**
  - Ghassen Baklouti, Maxime Zanella, Ismail Ben Ayed, *Language-Aware Information Maximization for Transductive Few-Shot CLIP*, [Under Review] [10]
  - Fereshteh Shakeri, Ghassen Baklouti, Julio Silva-Rodriguez, Maxime Zanella, Houda Bahig, Jose Dolz, Ismail Ben Ayed, *Test Time Adaptation of Medical Vision-Language Models*, Int. Workshop on Foundation Models for General Medical AI [11]

Summary: We leverage information-theoretic concepts and recent progress in parameter-efficient fine-tuning (PEFT), developing a highly competitive transductive few-shot CLIP method. Specifically, we introduce a novel Language-aware Information MaximizatiOn (LIMO) loss integrating three complementary terms: (i) the mutual information between the vision inputs and the textual class descriptions; (ii) a Kullback-Leibler divergence penalizing deviation of the probabilistic outputs from the text-driven zero-shot predictions; and (iii) a standard cross-entropy loss based on the labeled shots. Furthermore, we challenge the commonly followed fine-tuning practices in the context of transductive few-shot learning, and explore PEFT strategies, completely overlooked in this context. We further assess its versatiliy on medical contrastive vision language-models.

- **OGA (Chapter 6):**
  - Clément Fuchs, Maxime Zanella, Christophe De Vleeschouwer, *Online Gaussian Test-Time Adaptation of Vision-Language Models*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) [12]

Summary: We study online test-time adaptation (OTTA) of VLMs to take advantage of data observed along a stream to improve future predictions. Unfortunately, existing methods rely on dataset-specific hyper-parameters and incomplete evaluation protocols, limiting their generalization to new tasks. Thus, we propose Online Gaussian Adaptation (OGA), a novel method that models the likelihoods of visual features using Gaussian distributions and incorporates zero-shot priors into a concise *Maximum A Posteriori* (MAP) estimation framework with fixed hyper-parameters across all datasets. Besides, our experimental study reveals that common OTTA evaluation protocols, which average performance over at most three runs per dataset, are inadequate due to the variability observed across runs. Hence, we advocate for more rigorous evaluation practices, including increasing the number of runs and considering additional quantitative metrics, such as our proposed Expected Tail Accuracy (ETA), calculated as the average accuracy in the worst 10% of runs.

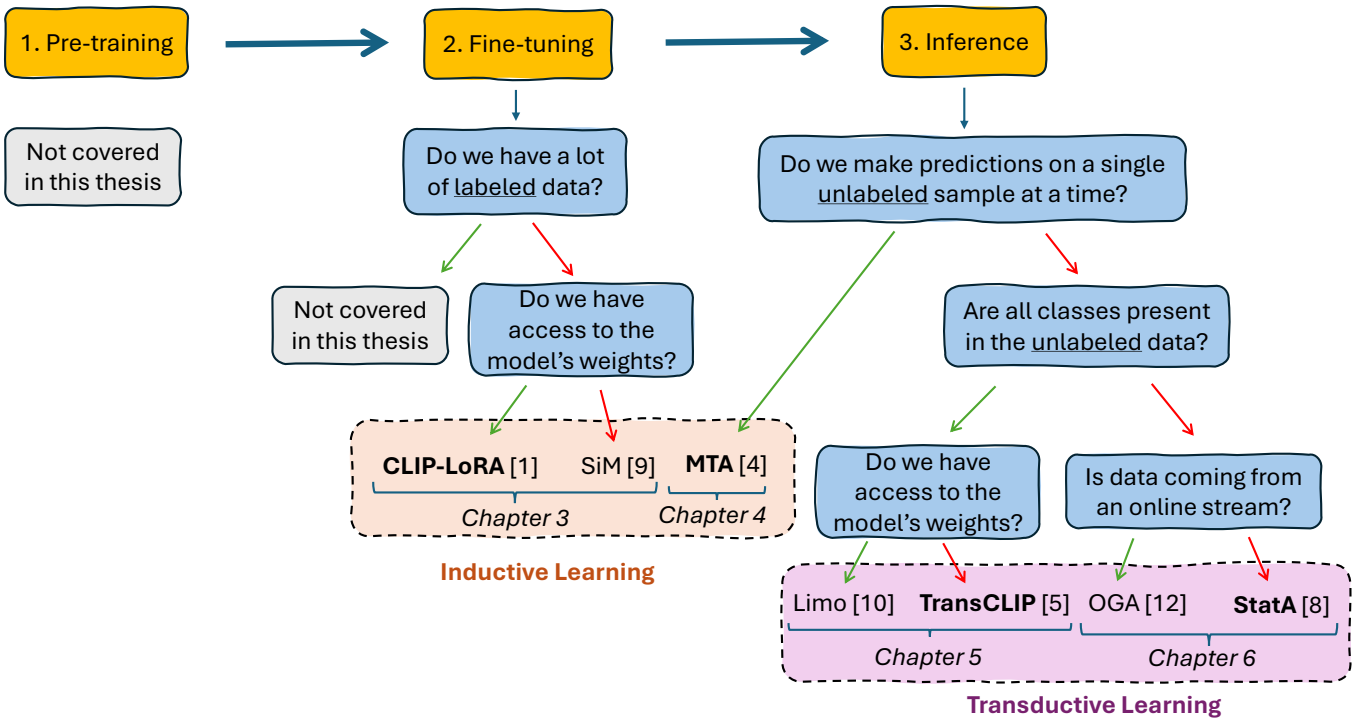


Fig. 1.1 Thesis roadmap integrated into the standard VLM development pipeline. We organize our contributions based on key practical constraints, such as label scarcity, access to model weights, and the nature of the test stream. The flowchart illustrates the decision paths (green arrows for "Yes", red arrows for "No") that partition our work into Inductive Learning (Chapters 3–4) and Transductive Learning (Chapters 5–6). Core contributions, which are developed in detail in this thesis, are highlighted in **bold**.

2

Background

Preliminary Note:

This chapter establishes some fundamental concepts and terminology used throughout this thesis. It begins by formalizing the **supervision regimes** (*zero-shot learning*, *few-shot learning*, *test-time adaptation*) and the **learning paradigms** (*inductive* vs. *transductive*) that structure our contributions, followed by a presentation of the **Contrastive Language-Image Pre-training (CLIP)** framework.
Readers might need this chapter to get used to the notations; however, those who are already familiar with transductive zero-shot/few-shot learning may safely skip Section 2.2, and those familiar with the inner workings of CLIP, Section 2.3.

2.1 Introduction

This thesis investigates the adaptation of foundational models to downstream visual recognition tasks. Before detailing specific model architectures and adaptation techniques, it is essential to formalize the supervision regimes and the learning paradigms that structure our contributions. We distinguish between *zero-shot* and *few-shot* availability of labeled data as well as between *inductive* and *transductive* learning paradigms (based on how test data is processed). Finally, we introduce the *contrastive vision-language model CLIP* architecture that serves as the backbone for all subsequent chapters.

2.2 Learning Paradigms

2.2.1 Supervision Regimes

We can categorize our contributions based on the availability of labeled data for the downstream task. Let $\mathbf{x}$ denote an input image sample and $y \in \{1, \dots, K\}$ its corresponding ground-truth class label, where K represents the total number of classes.

Few-shot supervision

Often referred to in the literature as *few-shot learning*, this regime is characterized by the availability of a limited number of labeled examples. We identify a set of indices $\mathcal{S}$, referred to as the *support set*, containing the K classes with N_{shots} examples per class (we will refer later to this setting as "N_{shots}-shot"), of total size $N_{\mathcal{S}} = N_{shots}K$. The goal is to predict the labels of a set of $N_{\mathcal{Q}}$ unlabeled *query* samples, indexed by the set $\mathcal{Q}$:

$$\begin{aligned} \mathcal{S} &= \{i \in \mathbb{N} \mid \mathbf{x}_i \text{ is a labeled support sample}\} \\ \mathcal{Q} &= \{j \in \mathbb{N} \mid \mathbf{x}_j \text{ is an unlabeled query sample}\} \end{aligned} \tag{2.1}$$

For the sake of brevity, in the remainder of this thesis, we will sometimes refer to the sets of samples $\{\mathbf{x}_i\}_{i \in \mathcal{S}}$ and $\{\mathbf{x}_j\}_{j \in \mathcal{Q}}$ simply as the support set $\mathcal{S}$ and the query set $\mathcal{Q}$, respectively, when the distinction between indices and samples is clear from context.

Zero-shot learning
In this regime, the model has access to the class names but observes no labeled images ($\mathcal{S} = \emptyset$). The model must rely entirely on its pre-trained knowledge to correctly associate each image in $\mathcal{Q}$ with the corresponding textual descriptions of these classes.

Test-time adaptation (TTA)
TTA represents a specific case of unsupervised domain adaptation where the model adapts to a stream of test data $\mathcal{Q}$ at inference time. TTA can be sample-by-sample (i.e., *inductive*) or batch-by-batch (i.e., *transductive*), often aiming to mitigate distribution shifts between the (pre-)training data and $\mathcal{Q}$.

2.2.2 Inductive vs. Transductive Learning

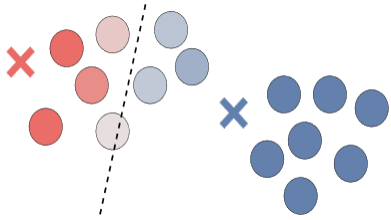
The fundamental distinction between the methods proposed in this thesis lies in how they process test data: whether predictions are made independently for each query sample (i.e., inductive inference) or jointly across a batch of unlabeled query samples (i.e., transductive inference). A schematic comparison is provided in Figure 2.1.

Inductive learning
In the standard inductive setting, the model optimizes its parameters θ to map inputs to predictions using a training set (i.e., to learn a general mapping rule). At inference time, the prediction for a specific query sample $\mathbf{x}$ depends solely on that sample and the learned parameters. This paradigm strictly separates the acquisition of knowledge from its application and does not leverage the distribution of other test samples.

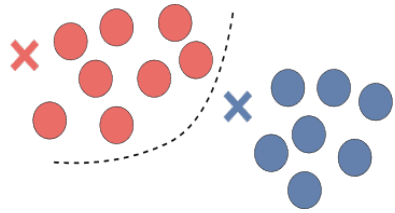
$$\hat{y}_i = \underset{y \in \{1,\dots,K\}}{\arg\max}\, P(y|\mathbf{x};\theta). \tag{2.2}$$

This is the standard setting for Chapter 3 (CLIP-LoRA) and Chapter 4 (MTA), where adaptation occurs either prior to inference at the fine-tuning stage or on a single-sample basis.

Transductive learning
In the transductive setting, the goal is to predict the labels for the entire query set $\mathcal{Q}$, given the support set $\mathcal{S}$ (where $\mathcal{S} = \emptyset$ in the zero-shot regime). Unlike inductive inference, transduction performs joint prediction, leveraging the structure of the unlabeled data (e.g., the manifold struc-



(a) Inductive inference: Each query sample is classified independently based on proximity to the nearest centroid. The decision boundary remains fixed regardless of the query distribution.



(b) Transductive inference: The manifold structure of the unlabeled query data is leveraged to adjust the decision boundary, so that close samples have similar predictions.

Fig. 2.1 Comparison of inductive and transductive prediction in a binary classification task. Crosses represent learned class prototypes (centroids), while circles denote unlabeled query samples. Note how transductive inference shifts the decision boundary (dashed line) to better fit the data distribution.

ture or class distribution) to improve accuracy.

$$\{\hat{y}_1, \ldots, \hat{y}_{N_\mathcal{Q}}\} = \underset{y_1,\ldots,y_{N_\mathcal{Q}} \in \{1,\ldots,K\}^{N_\mathcal{Q}}}{\arg\max} P(y_1, \ldots, y_{N_\mathcal{Q}} | \mathcal{Q}, \mathcal{S}; \theta). \tag{2.3}$$

It is important to note that transductive methods often involve optimizing an objective function over the test batch to refine internal representations or model parameters. Because this process entails an active optimization step, the distinction between "learning" and "inference" becomes blurred. Consequently, we will sometimes use the terms *transductive inference* and *transductive learning* interchangeably in this thesis. This paradigm is central to Chapter 5 (TransCLIP) and Chapter 6 (Stat.$\mathcal{A}$), where we exploit the statistics of the test batch to refine predictions.

Now that we have established the learning paradigms, let us turn our attention to the *Contrastive Language-Image Pre-training (CLIP)* framework. This architecture serves as the backbone for the zero-shot and few-shot capabilities across both inductive and transductive inference that we explore throughout the remainder of this thesis.

2.3 Contrastive Language-Image Pre-training (CLIP)

2.3.1 Overview

Unlike standard vision models trained on fixed label sets, CLIP [16] is composed of two encoders, one per modality, trained on image-text pairs collected from the internet. This allows the model to learn a broad range of visual concepts and semantic relationships. The core components of this framework are detailed below.

Data

Standard computer vision models are typically trained on datasets containing a fixed set of discrete categories (e.g., the 1,000 classes of ImageNet [17]). In contrast, CLIP leverages natural language supervision, learning from a massive dataset of 400 million image-text pairs collected from the internet. By training on the vast amount of text naturally associated with images on the web, the model can learn a much broader range of visual concepts and semantic relationships, effectively scaling up supervision without the bottleneck of human labeling. We elaborate on this comparison in Section 2.3.2.

Model architecture

CLIP employs a dual-encoder architecture consisting of two distinct neural networks: a vision encoder and a text encoder. As we will detail in Section 2.3.3, the text encoder is a transformer, whereas the vision encoder can be instantiated as either a convolutional neural network (e.g., ResNet) or a vision transformer (ViT). These two encoders operate independently to extract features from their respective inputs, projecting them into a shared, joint embedding space where visual and textual vectors can be directly compared via cosine similarity (see Figure 2.2).

Training objective

The alignment of the visual and textual embedding spaces is achieved through a contrastive learning objective. Instead of predicting a specific word or class label, CLIP is trained to solve an association task: given a batch of N (image, text) pairs, the model must predict which of the $N \times N$ possible pairings across the batch actually occurred. This effectively forces the model to learn representations where semantically similar images and texts are close. This contrastive objective function is further explained in

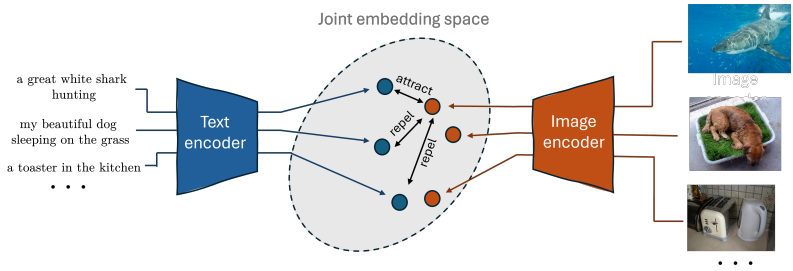


Fig. 2.2 Illustration of cross-modal alignment in the joint embedding space. The model minimizes the distance between corresponding image-text pairs (attraction) and maximizes the distance between mismatched pairs (repulsion) to align the two modalities.

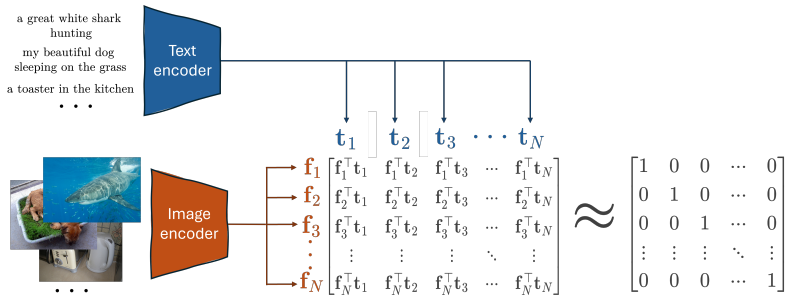


Fig. 2.3 Illustration of the batch-wise contrastive learning mechanism. For a batch of N image-text pairs, CLIP computes an $N \times N$ matrix of pairwise cosine similarities. The training objective maximizes the diagonal elements (correct pairs) while minimizing off-diagonal values, effectively driving the similarity matrix toward the identity matrix.

Section 2.3.4. As illustrated in Figure 2.3, the loss function maximizes the cosine similarity of the N correct pairs (the diagonal of the similarity matrix) while simultaneously minimizing the similarity of the $N^2 - N$ incorrect pairings.

Inference

One of the most impressive results of the CLIP paper was its ability to perform zero-shot transfer to downstream tasks. Because the model has learned to align images with arbitrary text descriptions, it is not bound by a predefined classification head. During inference, the names of the target classes can be inserted into a prompt template (e.g., `a photo of a`

[`class name`]) and encoded by the text encoder. The model then classifies an image by assigning it to the class whose prompt embedding has the highest similarity score with the image embedding. This will be detailed in Section 2.3.5.

2.3.2 Natural Language Supervision

To better understand the paradigm shift introduced by CLIP, it is essential to contrast its data structure with that of traditional computer vision.

Traditional supervised learning
Standard vision models are typically trained on datasets constructed through crowd-sourced manual annotation, such as ImageNet. Formally, such a dataset consists of pairs $\{(\mathbf{x}, y)\}$ where the label y belongs to a fixed, predefined ontology of K classes. While effective for specific tasks, this approach is fundamentally limited by the cost of human annotation and the fixed label set.

Natural language supervision
CLIP overcomes these scalability constraints by leveraging *natural language supervision*. Instead of discrete labels, the model is trained on a dataset of image-text pairs, with the text being a raw natural language description (e.g., a caption or alt-text) associated with the image $\mathbf{x}$. This approach offers two critical advantages. First, it removes the annotation burden, as the process can be totally unsupervised and rely on data naturally available on the web, effectively scaling up supervision without the bottleneck of human labeling. Second, the dataset presents a much richer semantic structure than one with a fixed ontology. For instance, where a traditional dataset might collapse semantically distinct images (e.g., "`a puppy running in the grass`" and "`an old dog sleeping`") into a single class index y (e.g., `dog`), natural language supervision preserves these nuances. Hence, this paradigm is not bound by a closed set of categories, enabling *open-vocabulary* learning where the model can acquire visual concepts that were not explicitly enumerated before training.

2.3.3 Model Architecture: Dual Encoder

CLIP employs a dual-encoder architecture consisting of a vision encoder θ_v and a text encoder θ_t. The text encoder employs a transformer [18], which is the current default architecture for natural language processing. While

the original work explores ResNet-based vision encoders [19], this thesis focuses primarily on the Vision Transformer (ViT) architecture [20], which has become the de facto standard for scalable vision-language modeling.

Transformer block

Both θ_v and θ_t consist of a stack of L identical layers. Each layer comprises two primary sub-layers: a multi-head self-attention (MHSA) mechanism and a feed-forward network. Transformers process sequential data via *self-attention*, enabling the model to weigh the relative importance of sequence elements regardless of their position. Specifically, CLIP utilizes self-attention where queries, keys, and values are derived from the same input (in contrast to the cross-attention found in decoder architectures, where queries and keys/values originate from different sources). We specifically detail the MHSA mechanism below, as it contains the parameter matrices that serve as the primary targets for the adaptation strategies discussed in Chapter 3. For a comprehensive description of the transformer architecture, we refer the reader to [18].

Multi-head self-attention (MHSA)

The MHSA mechanism allows the model to jointly attend to information from different representation subspaces. Given an input sequence of token embeddings $\mathbf{E} \in \mathbb{R}^{L \times d_{model}}$ (where L is the sequence length), the mechanism computes scaled dot-product attention multiple times in parallel.

For each head $h \in \{1, \ldots, H\}$, the input is projected using learnable parameter matrices $\mathbf{W}_Q^{(h)}, \mathbf{W}_K^{(h)}, \mathbf{W}_V^{(h)} \in \mathbb{R}^{d_{model} \times d_{head}}$, yielding queries $\mathbf{Q}_h$, keys $\mathbf{K}_h$, and values $\mathbf{V}_h$:

$$\mathbf{Q}_h = \mathbf{E}\mathbf{W}_Q^{(h)}, \quad \mathbf{K}_h = \mathbf{E}\mathbf{W}_K^{(h)}, \quad \mathbf{V}_h = \mathbf{E}\mathbf{W}_V^{(h)} \tag{2.4}$$

The scaled dot-product attention for head h is computed as:

$$\text{head}_h = \text{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h) = \text{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{d_{head}}}\right)\mathbf{V}_h \tag{2.5}$$

where $d_{head} = d_{model}/H$. The outputs from all H heads are concatenated and projected via a final linear transformation $\mathbf{W}_O \in \mathbb{R}^{d_{model} \times d_{model}}$ to produce the final output:

$$\text{MHSA}(\mathbf{E}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)\mathbf{W}_O \tag{2.6}$$

These weight matrices $\{\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O\}$ constitute the primary targets for the parameter-efficient fine-tuning methods (specifically CLIP-LoRA) discussed in Chapter 3.

From input to output

The vision encoder θ_v processes the input image $\mathbf{x}$ as a sequence of flattened patches, while the text encoder θ_t processes BPE-tokenized text. Both sequences are linearly projected to the hidden dimension d_{model} and augmented with learnable position embeddings to retain spatial and sequential information. The final global representations are derived from specific pooling tokens: the learnable classification token (`[CLS]`) for the image and the end-of-sequence token (`[EOS]`) for the text. For more details, we refer the reader to [18] and [20]. Crucially, these pooled representations (in $\mathbb{R}^{d_{model}}$) are passed through a final linear projection layer to map them to the joint embedding space (in $\mathbb{R}^d$) [16].

We now have the necessary components to map raw images and natural language descriptions into fixed-dimensional embedding vectors. However, we still require a mechanism to define how their learnable parameters are optimized simultaneously. This missing piece is the objective function that dictates training, ensuring that these two distinct modalities are effectively aligned within a shared embedding space. The following section details this training objective, known as contrastive learning.

2.3.4 Contrastive Learning

CLIP is pre-trained via a contrastive learning objective on a massive dataset of image-text pairs [16]. The fundamental goal is to construct a shared multi-modal embedding space where the representations of matched image-text pairs are pulled together, while those of non-matching pairs are pushed apart.

Image and text embeddings

Consider a batch of N image-text pairs. The model consists of a vision encoder θ_v and a text encoder θ_t. For each pair, we compute the normalized feature embeddings $\mathbf{f}_i, \mathbf{t}_i \in \mathbb{R}^d$, with $\mathbf{c}_i$ representing the tokenized version of the input text:

$$\mathbf{f}_i = \frac{\theta_v(\mathbf{x}_i)}{\|\theta_v(\mathbf{x}_i)\|_2}, \quad \mathbf{t}_i = \frac{\theta_t(\mathbf{c}_i)}{\|\theta_t(\mathbf{c}_i)\|_2}, \quad i = 1, \ldots, N. \tag{2.7}$$

We obtain N image embeddings $\{\mathbf{f}_1, \ldots, \mathbf{f}_N\}$ and N text embeddings $\{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$. Then we compute the cosine similarity for all N^2 possible pairings within the batch, resulting in a similarity matrix where the (i, j)-th entry is given by $\mathbf{f}_i^\top \mathbf{t}_j$ for $i, j \in \{1, \ldots, N\}$ (see Figure 2.3).

Symmetric loss function

The training objective maximizes the cosine similarity of the N correct pairs (diagonal elements) while minimizing the similarity of the $N^2 - N$ incorrect pairs (off-diagonal elements). This is implemented via a symmetric cross-entropy loss (InfoNCE). Specifically, for a given image embedding $\mathbf{f}_i$, the probability of matching it to the text embedding $\mathbf{t}_j$ is modeled using a softmax distribution over all texts in the batch, scaled by a learnable temperature parameter τ:

$$s_{i,j}^{\text{image}} = \frac{\exp(\mathbf{f}_i^\top \mathbf{t}_j / \tau)}{\sum_{k=1}^{N} \exp(\mathbf{f}_i^\top \mathbf{t}_k / \tau)} \tag{2.8}$$

Note that this matching problem effectively reduces to a classification task over the batch. Crucially, this training mechanism mirrors the zero-shot inference mechanism (Section 2.3.5).

The image-to-text loss for sample i is then defined as the negative log-likelihood of the correct text $\mathbf{t}_i$:

$$\mathcal{L}_i^{\text{image}} = -\log(s_{i,i}^{\text{image}}) = -\log\left(\frac{\exp(\mathbf{f}_i^\top \mathbf{t}_i / \tau)}{\sum_{k=1}^{N} \exp(\mathbf{f}_i^\top \mathbf{t}_k / \tau)}\right) \tag{2.9}$$

Symmetrically, a text-to-image loss $\mathcal{L}_i^{\text{text}}$ is computed for each text embedding $\mathbf{t}_i$ over all images. The total contrastive loss $\mathcal{L}_{\text{CLIP}}$ is the average of these two losses over the batch:

$$\mathcal{L}_{\text{CLIP}} = \frac{1}{2N} \sum_{i=1}^{N} \left(\mathcal{L}_i^{\text{image}} + \mathcal{L}_i^{\text{text}}\right) \tag{2.10}$$

Minimizing this loss over the 400 million pairs dataset jointly optimizes θ_v and θ_t to create an aligned multi-modal embedding space. This contrastive objective enables the model to learn robust visual representations guided by natural language supervision, establishing the foundation for the zero-shot capabilities detailed next.

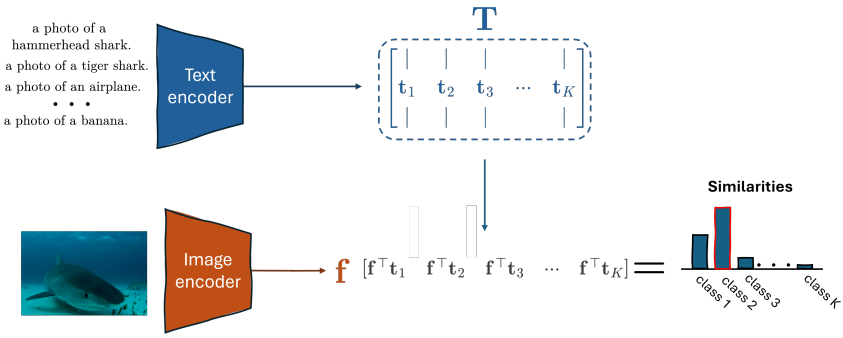


Fig. 2.4 For zero-shot classification, the text encoder transforms K class prompts into embedding vectors $\mathbf{t}_1, \ldots, \mathbf{t}_K$, forming a classification matrix $\mathbf{T}$. Prediction is performed by computing the dot product between the image feature vector $\mathbf{f}$ and each text vector to find the maximum cosine similarity.

2.3.5 Zero-Shot Classification

For a classification task with K candidate classes, we leverage the aligned embedding space of the pre-trained VLM. We first construct textual descriptions, termed prompts, for each category. For instance, a prompt may take the form of a template `a photo of a` $[\texttt{class}_k]$, for $k = 1, \ldots, K$. These prompts are processed by the text encoder θ_t to produce normalized class embeddings $\mathbf{t}_k \in \mathbb{R}^d$. Simultaneously, a test image $\mathbf{x}_i$ is mapped by the visual encoder θ_v to its normalized visual embedding $\mathbf{f}_i \in \mathbb{R}^d$.

Prediction

Classification is performed by computing the cosine similarity between the image embedding and the set of class embeddings. The prediction logit for class k is given by:

$$l_{i,k} = \mathbf{f}_i^\top \mathbf{t}_k. \tag{2.11}$$

These logits are converted into a text-based posterior probability distribution over the K classes via the softmax function:

$$s_{i,k} = P(y = k|\mathbf{x}_i) = \frac{\exp(l_{i,k}/\tau)}{\sum_{j=1}^{K} \exp(l_{i,j}/\tau)}. \tag{2.12}$$

This formulation mirrors the contrastive training objective (Section 2.3.4) by design. By treating the K class prompts as a batch of candidate texts,

zero-shot inference effectively repurposes the pre-training mechanism to discriminate between categories without requiring a specialized classification head. The predicted label $\hat{k}_i$ for image $\mathbf{x}_i$ is simply the index maximizing this probability:

$$\hat{k}_i = \underset{k \in \{1,\dots,K\}}{\arg\max}\, s_{i,k}. \tag{2.13}$$

This allows the model to adapt to arbitrary categories defined in natural language without retraining, as illustrated in Figure 2.4.

Analogy to linear classifiers

Intuitively, this process can be viewed as generating a dynamic classification weight matrix $\mathbf{T} \in \mathbb{R}^{d \times K}$ by concatenating the text embeddings $\mathbf{t}_1, \dots, \mathbf{t}_K$ as columns. The vector of logits is then obtained via the matrix-vector multiplication $\mathbf{l}_i = \mathbf{f}_i^\top \mathbf{T}$. Unlike standard supervised classifiers where the weight matrix contains fixed parameters optimized for a closed set of classes, here the weights are dynamically generated by the text encoder.

2.4 Limitations and Motivation for Adaptation

While CLIP's zero-shot capabilities are impressive, they are not without limitations. The model's performance is heavily dependent on the alignment between the pre-training distribution and the downstream task. Significant domain shifts such as those encountered in satellite imagery (Chapter 3) or medical histology (Chapter 5) can degrade zero-shot accuracy. Moreover, even on the standard natural image benchmarks studied in this thesis, which are often considered as in-domain, our experiments consistently demonstrate that carefully designed adaptation yields substantial performance gains.

These challenges motivate the need for robust adaptation strategies. The remainder of this thesis explores methods to mitigate these issues: improving performance through inductive few-shot learning (Chapter 3), adapting to single samples at test time (Chapter 4), and leveraging transductive inference to exploit unlabeled data structures (Chapters 5 and 6).

3

Inductive Few-Shot Learning

Preliminary Note:

This chapter expands on the conference proceedings [1] and includes additional results from [3]. It investigates **inductive fine-tuning** approaches for Vision-Language Models under **weak supervision** (i.e., *few-shot learning*). Specifically, we propose a parameter-efficient strategy that injects trainable **low-rank matrices** into the attention layers, allowing for robust adaptation while keeping the pre-trained backbone frozen.
This chapter also serves as an introduction to common adaptation techniques for CLIP, such as popular few-shot adaptation methods, the principles of which are shared with Chapters 4, 5, 6, and 7.

3.1 Introduction

This chapter marks the beginning of our technical contributions by tackling the fundamental challenge of *few-shot learning*, i.e., adapting models using only a few labeled examples. Given the extensive existing literature on this topic, it serves as an ideal entry point for understanding the mechanisms of adapting contrastive Vision-Language Models (VLMs). Specifically, we focus on the *inductive* paradigm (defined in Section 2.2), where adaptation is confined entirely to the *fine-tuning* phase. In this setting, the model is fine-tuned on the support set and then deployed to make predictions on test samples independently, without further intervention.

3.1.1 Context and Research Positioning

Vision-Language Models (VLMs), such as CLIP [16], have emerged as powerful tools for learning cross-modal representations [16,21–24]. Pre-trained on extensive collections of image-text pairs with a contrastive objective, they learn to align these two modalities, enabling zero-shot prediction by matching the visual embeddings of the images and text descriptions (prompts) representing the target tasks. This joint representation space of visual and textual features has also opened up new possibilities in the *pre-training*, *fine-tuning*, *inference* paradigm, through adaptation with very limited amounts of task-specific, labeled data [25–27], i.e., few-shot adaptation. In this chapter, we will focus on the fine-tuning phase while the other chapters will be oriented towards the inference phase.

The fine-tuning stage has triggered wide interest in NLP, where the increase in the sizes of foundational models is going at a fast pace, with some models boasting over 176 billion parameters [28], and even up to 540 billion [29]. To address this challenge, *Parameter-Efficient Fine-Tuning (PEFT)* methods, which attempt to tune only a small subset of parameters (in comparison to the original large models), have gained substantial attention [30]. Popular PEFT methods include selecting a subset of the existing parameters [31], adding small trainable modules called adapters [32–36], or adding trainable (prompt) tokens [37–39].

The original CLIP paper demonstrated that better textual descriptions could greatly impact the zero-shot prediction [16]. This observation has

been a strong motivation for the emergence of prompt tuning [37], a strategy that has been widely adopted within the vision-language community, following the seminal work of CoOp [25]. Indeed, CoOp popularized prompt tuning in the setting of few-shot VLMs. This has triggered a quite substantial recent literature focusing on improving prompt-learning performances for VLMs, in both the few-shot [25, 26, 40–45] and unsupervised settings [46–48]. While prompt learning methods improve the zero-shot performances, they incur heavy computational load for fine-tuning and might be hard to optimize, since every gradient update of the input requires back-propagating through the entire model.

Alongside this expanding prompt-tuning literature, there have been a few attempts to propose alternative approaches for few-shot VLMs, generally relying on adapters [27, 49, 50]. However, the performances of such adapters depend strongly on a set of hyperparameters (such as the learning rate, number of epochs, or parameters controlling the blending of image and text embeddings) [51], which must be tuned specifically for each target dataset. This is done via intensive searches over validation sets, requiring additional labeled samples and incurring computational overhead, which reduces their portability to new tasks.

Recently, a novel approach consisting of solely fine-tuning low-rank matrices called Low-Rank Adaptation (LoRA) has appeared as a promising and practical method [52]. PEFT approaches, such as LoRA, have democratized the fine-tuning of large language models [53]. Far from merely enhancing computational efficiency, empirical evidence has shown that, in the large-scale fine-tuning setting, LoRA could match or even exceed the performance of full fine-tuning (i.e., updating all model parameters) [52]. *Although very promising/popular, and quite surprisingly, this fast-growing PEFT literature [52–57] has found little echo in few-shot vision-language, where the dominant approaches have mainly focused on prompt tuning [25, 26, 40, 43–45] or adapters [27, 49, 50].*

3.1.2 Summary of Contributions

We propose **CLIP-LoRA**, a robust parameter-efficient fine-tuning method for inductive few-shot learning. Our contributions are threefold:

(i) We introduce and systematically investigate the deployment of Low-Rank Adaptation (LoRA) for few-shot VLMs. We thoroughly examine the

primary design choices, including the selection of encoders (vision, language, or joint), the specific attention weight matrices to adapt, and the optimal rank (r).

(ii) We establish CLIP-LoRA as a high-performing and efficient baseline. Through comprehensive empirical comparisons on ten natural image datasets and ten satellite imagery datasets, we demonstrate that our simple baseline consistently outperforms state-of-the-art methods by significant margins, all while substantially reducing computational overhead.

(iii) We demonstrate that our approach is straightforward to deploy, as it eliminates the need for intensive, per-task hyperparameter searches over dataset-specific validation sets. We show that a single, consistent hyperparameter configuration achieves robust performance across all tasks and shot counts, establishing a strong new baseline for future research.

3.1.3 Chapter Outline

The remainder of this chapter is structured as follows. Section 3.2 provides the necessary context by reviewing the principles of few-shot fine-tuning and the state-of-the-art inductive literature. Next, Section 3.3 details the core contribution of this chapter, low-rank adaptation (CLIP-LoRA) and its main design choices. Section 3.4 presents a comprehensive experimental validation through a systematic comparison of inductive methods on conventional natural image benchmarks. Section 3.5 provides ablation studies on those design choices such as rank and encoder selection. The evaluation is then extended in Section 3.6 to remote sensing imagery. Finally, Section 3.7 concludes the chapter by summarizing the key findings regarding the efficiency and performance of low-rank few-shot adaptation in VLMs.

3.2 Few-Shot Fine-Tuning for VLMs

This section provides a broad overview of recent few-shot fine-tuning methods designed for VLMs, summarized in Figure 3.1.

3.2.1 Fundamentals of VLM Classification

We build upon the zero-shot classification framework established in Section 2.3.5; we kindly invite the reader to refer to this section to familiarize

themselves with the notations and concepts. Recall that for a test image $\mathbf{x}_i$ and a set of K class prompts, we can use the cosine similarity between the visual embedding $\mathbf{f}_i$ and the text embedding $\mathbf{t}_k$. This yields a probabilistic prediction, in the form of a posterior softmax probability of class k given test input $\mathbf{x}_i$, where τ is a softmax-temperature parameter:

$$s_{i,k} = P(y = k|\mathbf{x}_i) = \frac{\exp(l_{i,k}/\tau)}{\sum_{j=1}^{K} \exp(l_{i,j}/\tau)}, \quad l_{i,k} = \mathbf{f}_i^\top \mathbf{t}_k. \tag{3.1}$$

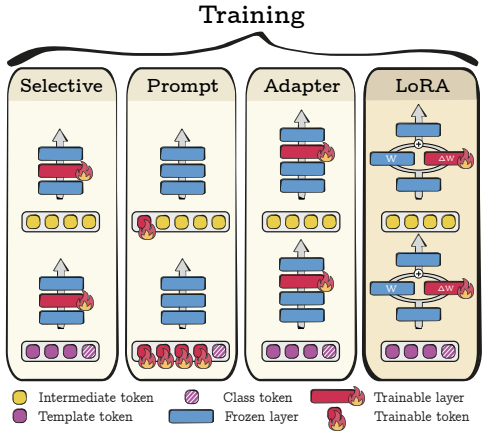
Hence, zero-shot classification of image $\mathbf{x}_i$ is done by finding the class with the highest posterior probability:

$$\hat{k}_i = \underset{k \in \{1,\ldots,K\}}{\arg\max}\, s_{i,k}. \tag{3.2}$$

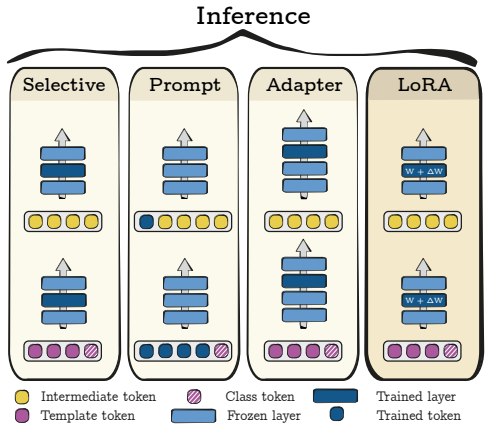
In the few-shot setting, and to further adapt these models, we assume that we have access to N_{shots} labeled samples for each target class, the so-called *support* set $\mathcal{S}$. $|\mathcal{S}| = N_{\mathcal{S}} = N_{shots}K$ denotes the total number of support samples, and N_{shots} (the number of shots per class) is typically small (less than 16). Let $y_{i,k}$ denote the one-hot encoded label for a labeled support image $\mathbf{x}_i$, i.e., $y_{i,k} = 1$ if k is the class of image $\mathbf{x}_i$ and 0 otherwise. Then, we minimize the cross-entropy (CE) loss:

$$\mathcal{L}_{CE} = -\frac{1}{N_{\mathcal{S}}} \sum_{i \in \mathcal{S}} \sum_{k=1}^{K} y_{i,k} \ln s_{i,k}. \tag{3.3}$$

This is done either (i) by fine-tuning the input prompts, $\mathbf{c}_k, k = 1, \ldots, K$, as in prompt-tuning methods [25, 26, 40, 43, 45, 58]; or (ii) by updating a set of additional parameters, as in adapters [27, 49, 50]. Note that other methods propose to tune additional intermediate tokens [58], which we include under the category prompt tuning for a more general terminology. We will now detail the two current strategies used in VLMs: Prompt tuning (P) and Adapters (A), as well as Selective (S) methods before detailing our proposed approach based on low-rank matrices (LoRA). All these approaches can be encompassed into the broad term Parameter-Efficient Fine-Tuning (PEFT). PEFT seeks to reduce the high expense of fine-tuning large-scale models by concentrating on (re-)training a relatively small number of parameters. The choice of parameters often results in a trade-off among memory footprint, computational overhead, and performance [30]. A summary is depicted in Figure 3.1.



(a) Prompt, Adapter and Low-rank techniques introduce extra parameters for training, which may potentially extend training duration and/or memory footprint in comparison to selective methods. However, they have the advantage of being more flexible and are often easier to use.



(b) Prefix and Adapter methods result in extra parameters after adaptation, potentially increasing inference time and memory footprint relative to the vanilla model. Conversely, LoRA merges newly trained low-rank matrices with the original frozen ones, eliminating additional parameters at inference.

Fig. 3.1 Different categories of Parameter-Efficient Fine-Tuning (PEFT) methods during (a) training, and (b) inference.

3.2.2 Review of the Inductive Few-Shot Literature

Prompt tuning (P)
The way prompting is performed in VLMs could significantly impact the ensuing performances [16]. To address this issue, soft prompt tuning [37, 38] optimizes text-input tokens, which could be extended to intermediate layers [38]. Similarly, if a transformer-based architecture is used, these learnable tokens can be inserted in intermediate layers of vision models [39]. In the context of few-shot VLMs, context optimization (CoOp) [25] constructs text input $\mathbf{c}_k$ as continuous trainable vectors:

$$\mathbf{c}_k = (\mathbf{v}_k^1, \ldots, \mathbf{v}_k^M, [\texttt{class}_k]) \tag{3.4}$$

Where M denotes a hyper-parameter, $(\mathbf{v}_k^l)_{1 \leq l \leq M}$ are trainable text tokens, and $[\text{class}_k]$ is a fixed token. The latter is the word embedding vector of the name of the k^{th} class. These trainable vectors are updated as task-specific text prompts by using the standard supervised CE classification loss in Eq. (3.3), along with the few-shot labeled samples. Prompt tuning has a clear advantage over adapter-based methods: They remove the need for heuristically choosing the text prompts, which are specifically engineered for each task, and whose choice might affect the performances significantly.

The success of prompt tuning for adapting VLMs is undeniable [25, 26, 40–45]. To describe a few in addition of CoOp. CoCoOp [40] trains a neural network to generate instance-conditioned tokens based on the image. Further efforts like ProGrad [26] and KgCoOp [43], among others [44], guide prompts towards predefined handcrafted ones, for example, thanks to gradients projection [26] with the idea of preserving initial knowledge during learning. PLOT [45] is one of the first to adapt jointly the text and image modalities. They propose to align learned prompts with finer-grained visual features through an optimal transport formulation. Following a similar trajectory, MaPLe [58] introduces intermediate learnable tokens within both the vision and text encoders, while making them interdependent at each level. While prompt-tuning methods improves significantly the performances of classification, they incur heavy computational load for fine-tuning and might be hard to optimize, since every gradient update of the text input requires back-propagating through the entire model.

Adapters (A)

Instead of updating the text prompts, another class of methods, called adapters, augment the pre-trained model with extra parameters while keeping the existing ones frozen [33]. This provides an efficient way to control the number of trainable parameters. The idea has been recently explored in the few-shot vision-language setting [27, 49, 50]. In this setting, adapters could be viewed as feature transformations, via some multi-layer modules, appended to the encoder's bottleneck. This enables to learn transformations blending image and text features, with logits taking the following form:

$$l_{i,k} = \theta_a(\mathbf{f}_i, \mathbf{t}_k) \tag{3.5}$$

where θ_a denotes the additional trainable parameters of the adapter. These are fine-tuned by minimizing the CE loss in (3.3), using the labeled support set but now with logits $l_{i,k}$ transformed by (3.5). This class of methods can reduce the computational load in comparison with prompt-tuning techniques. However, as pointed out recently in the experiments in [51], their performances might depend strongly on some key hyperparameters that have to be adjusted specifically for each downstream task, which reduces their portability to new tasks.

Nonetheless, adapter-based methods offer an alternative strategy to prompt tuning and are increasingly studied [27, 49, 50, 59]. For instance CLIP-Adapter [49] learns visual adapters to combine adapted and original features. A few other methods only access the final embedding state. Examples include Tip-Adapter(-F) [27] using a cache model in few-shot learning and TaskRes [50] which keeps the original text weights frozen and introduces task residual tuning to learn task-specific adapters built on the initial knowledge.

Selective (S)

For completeness, we further discuss other methods in this paragraph but they will not be studied in the remainder of this chapter. *Selective* methods focus on a subset of the existing model weights. Among these, BitFit [31] fine-tunes only the biases of both the attention and MLP layers in the transformer blocks, while other approaches prune the model to create a task-specific subset of parameters [60]. Specifically for few-shot learning with CLIP, fine-tuning the affine parameters of the Layer Normalization layers has been demonstrated to be an effective adaptation strategy [61].

Low-Rank Adaptation (LoRA)
LoRA [52] models the incremental update of the pre-trained weights as the product of two small matrices, $\mathbf{A}$ and $\mathbf{B}$, based on the idea of "intrinsic rank" [62, 63] of a downstream task. For an input $\mathbf{x}$, a hidden state $\mathbf{h}$, and a weight matrix $\mathbf{W} \in \mathbb{R}^{d_{out} \times d_{in}}$, the modified forward pass, following the application of a LoRA module, is:

$$\mathbf{h} = \mathbf{W}\mathbf{x} + \gamma \Delta \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{x} + \gamma \mathbf{B}\mathbf{A}\mathbf{x} \tag{3.6}$$

where $\mathbf{A} \in \mathbb{R}^{r \times d_{in}}$ and $\mathbf{B} \in \mathbb{R}^{d_{out} \times r}$ with rank $r \ll \min(d_{in}, d_{out})$, and γ is a scaling factor. $\mathbf{A}$ is initialized with random Gaussian noise and $\mathbf{B}$ with zeros, ensuring that the update $\Delta\mathbf{W}$ is initially zero. This implies that there is no incremental update before training, and therefore, the output remains unchanged. LoRA is typically applied to the multi-head self-attention (MHSA) weight matrices $\{\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O\}$. Rather than re-introducing the architecture here, we refer the reader to Section 2.3.3 (specifically Eqs. 2.4–2.6), which details the MHSA mechanism.

Crucially, the learned matrices $\mathbf{A}$ and $\mathbf{B}$ can be merged into $\mathbf{W}$ after training ($\mathbf{W}' = \mathbf{W} + \gamma\mathbf{B}\mathbf{A}$), ensuring zero additional inference latency. Note that other works extend this approach to the feed-forward module's weight matrices [64]. LoRA can be viewed as an adapter approach, yet it offers the advantages of selective methods by providing a direct aggregation of its module, eliminating the need for extra parameters at the inference stage. Several versions of the original LoRA have since appeared, some focusing on making the rank adaptive for each matrix [54, 55], others pushing its performance [56, 57, 65] or reducing its memory footprint through quantization [53, 66].

3.3 CLIP-LoRA: Few-Shot Low-Rank Adaptation of VLMs

Main design choices
We identify three principal design considerations: (i) the choice between tuning the vision encoder, the text encoder, or both, including the specific layers to adjust; (ii) the selection of attention matrices for tuning; and (iii) the determination of the appropriate rank for these matrices. We explore these aspects across three datasets: ImageNet [17], Stanford Cars [67], and

EuroSAT [68]. ImageNet was selected for its broad diversity, while the latter two were chosen for their distinctive behaviors. Ablation studies on these design choices are discussed in Section 3.5 for seven different groups of adapted attention matrices and increasing rank value.

Implementation details
A straightforward way to apply LoRA in vision-language is to apply it to all the matrices of the vision and text encoders. However, due to the relatively limited supervision inherent to the few-shot setting, we only apply low-rank matrices to the query, key and value matrices ($\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$) with a rank of $r = 2$. We regularize the input of the LoRA module by a dropout layer with $p = 0.25$ [52]. The number of iterations is set equal to 500 times N_{shots} (the number of labeled samples per class). We use a learning rate of 2×10^{-4}, with a cosine scheduler and a batch size of 32, so that all training could be performed on a single GPU of 24 GB. *These hyper-parameters are kept fixed across all the experiments*. The input prompt is simply set to "`a photo of a` [$\texttt{class}_k$]", $k = 1, \ldots, K$, for every dataset, to emphasize the applicability of CLIP-LoRA without resorting to complex initial manual prompting. Note that the LoRA modules are positioned at all levels of both encoders. The impact of the rank and the location of the LoRA modules is studied in the following, before comparing low-rank adaptation to other existing inductive few-shot methods.

3.4 Experimental Validation on Natural Images

3.4.1 Experimental Setup

Datasets
We use conventional benchmarks composed of 11 datasets for fine-grained classification, namely Imagenet (I.) [17], SUN397 (SUN) [69], Aircraft (Air.) [70], EuroSAT (SAT) [68], StanfordCars (Cars) [67], Food101 (Food) [71], OxfordPets (Pets) [72], Flowers102 (Flow.) [73], Caltech101 (Calt.) [74], DTD (DTD) [75], UCF101 (UCF) [76]. These datasets offer a thorough benchmarking framework for evaluating few-shot visual classification tasks.

Models
We report performance on the CLIP ViT-B/16 backbone [16] and additional

results on the ViT-B/32 and ViT-L/14 versions.

Evaluation protocol
We follow the standard few-shot learning evaluation protocol by varying the number of labeled support examples per class, denoted as $N_{\text{shots}} \in \{1, 2, 4, 8, 16\}$. For each N_{shots} configuration, we randomly sample the support set $\mathcal{S}$ from the training data. The models are then fine-tuned using the samples indexed by $\mathcal{S}$ and evaluated on the full test set. To account for the variability inherent in few-shot sampling, we report the top-1 classification accuracy averaged over three random seeds.

Comparative methods
We compare CLIP-LoRA to several prompt-based methods: CoOp [25] (16) with 16 learnable tokens, CoCoOp [40], PLOT++ [45] which is an adaptation of the original PLOT proposed by the same authors specifically designed for transformer architectures, KgCoOp [43], MaPLe [58] for which we follow the training procedure of their "base-to-new" setting, ProGrad [26] with 16 tokens. We also report adapter-based methods: Tip-Adapter-F [27] for which we reduce the validation set to a reasonable size of $\min(N_{\text{shots}}, 4)$, TaskRes [50] for which we only report the not enhanced base performance due to its unavailability for all datasets, shots, and backbones studied in this paper. This diversity of techniques provides valuable insight into which parameters are most relevant to tune. For each method, we adopt the original hyperparameter settings specified in their respective publication if not specified otherwise. For example, for TaskRes, despite some questionable arbitrary choices as discussed in [51], we keep their specific hyperparameters while CLIP-LoRA uses the same hyperparameters (see Section 3.3) for every task.

3.4.2 Results

Comparison to state-of-the-art
The strongest adapter-based method in Table 3.1 is Tip-Adapter-F, which is not competitive with CLIP-LoRA despite relying heavily on arbitrary hyper-parameters for each dataset (namely the starting value of their α, β as well as the search range during validation). We can conclude the same for TaskRes, which also relies on arbitrary choices for a given dataset, i.e., a specific learning rate for ImageNet and a specific scaling factor for the Flowers dataset. Regarding prompt-based approaches, Table 3.1 shows

Table 3.1 Detailed results for 11 datasets with the ViT-B/16 as visual backbone. Highest value is highlighted in **bold**, and the second highest is underlined.

	Method	I.	SUN	Air.	SAT	Cars	Food	Pets	Flow.	Calt.	DTD	UCF	Average
	CLIP	66.7	62.6	24.7	47.5	65.3	86.1	89.1	71.4	92.9	43.6	66.7	65.1
1-shot	CoOp	65.7	67.0	20.8	56.4	67.5	84.3	90.2	78.3	92.5	50.1	71.2	67.6
	CoCoOp	69.4	68.7	28.1	55.4	67.6	84.9	91.9	73.4	94.1	52.6	70.4	68.8
	TIP-Adapter-F	69.4	67.2	28.8	67.8	67.1	85.8	90.6	**83.8**	94.0	51.6	73.4	70.9
	CLIP-Adapter	67.9	65.4	25.2	49.3	65.7	86.1	89.0	71.3	92.0	44.2	66.9	65.7
	PLOT++	66.5	66.8	28.6	65.4	68.8	86.2	91.9	80.5	**94.3**	**54.6**	74.3	70.7
	KgCoOp	68.9	68.4	26.8	61.9	66.7	**86.4**	92.1	74.7	94.2	52.7	72.8	69.6
	TaskRes	69.6	68.1	**31.3**	65.4	68.8	84.6	90.2	81.7	93.6	53.8	71.7	70.8
	MaPLe	69.7	69.3	28.1	29.1	67.6	85.4	91.4	74.9	93.6	50.0	71.1	66.4
	ProGrad	67.0	67.0	28.8	57.0	68.2	84.9	91.4	80.9	93.5	52.8	73.3	69.5
	CLIP-LoRA	**70.4**	**70.4**	30.2	**72.3**	**70.1**	84.3	**92.3**	83.2	93.7	54.3	**76.3**	**72.5**
4-shot	CoOp	68.8	69.7	30.9	69.7	74.4	84.5	92.5	92.2	94.5	59.5	77.6	74.0
	CoCoOp	70.6	70.4	30.6	61.7	69.5	86.3	92.7	81.5	94.8	55.7	75.3	71.7
	TIP-Adapter-F	70.7	70.8	35.7	76.8	74.1	86.5	91.9	92.1	94.8	59.8	78.1	75.6
	CLIP-Adapter	68.6	68.0	27.9	51.2	67.5	86.5	90.8	73.1	94.0	46.1	70.6	67.7
	PLOT++	70.4	71.7	35.3	83.2	76.3	86.5	92.6	92.9	95.1	62.4	79.8	76.9
	KgCoOp	69.9	71.5	32.2	71.8	69.5	**86.9**	92.6	87.0	95.0	58.7	77.6	73.9
	TaskRes	71.0	72.7	33.4	74.2	76.0	86.0	91.9	85.0	95.0	60.1	76.2	74.7
	MaPLe	70.6	71.4	30.1	69.9	70.1	86.7	**93.3**	84.9	95.0	59.0	77.1	73.5
	ProGrad	70.2	71.7	34.1	69.6	75.0	85.4	92.1	91.1	94.4	59.7	77.9	74.7
	CLIP-LoRA	**71.4**	**72.8**	**37.9**	**84.9**	**77.4**	82.7	91.0	**93.7**	**95.2**	**63.8**	**81.1**	**77.4**
16-shot	CoOp	71.9	74.9	43.2	85.0	82.9	84.2	92.0	96.8	95.8	69.7	83.1	80.0
	CoCoOp	71.1	72.6	33.3	73.6	72.3	**87.4**	93.4	89.1	95.1	63.7	77.2	75.4
	TIP-Adapter-F	73.4	76.0	44.6	85.9	82.3	86.8	92.6	96.2	95.7	70.8	83.9	80.7
	CLIP-Adapter	69.8	74.2	34.2	71.4	74.0	87.1	92.3	92.9	94.9	59.4	80.2	75.5
	PLOT++	72.6	76.0	46.7	92.0	84.6	87.1	**93.6**	97.6	96.0	71.4	85.3	82.1
	KgCoOp	70.4	73.3	36.5	76.2	74.8	87.2	93.2	93.4	95.2	68.7	81.7	77.3
	TaskRes	73.0	**76.1**	44.9	82.7	83.5	86.9	92.4	97.5	95.8	71.5	84.0	80.8
	MaPLe	71.9	74.5	36.8	87.5	74.3	**87.4**	93.2	94.2	95.4	68.4	81.4	78.6
	ProGrad	72.1	75.1	43.0	83.6	82.9	85.8	92.8	96.6	95.9	68.8	82.7	79.9
	CLIP-LoRA	**73.6**	**76.1**	**54.7**	**92.1**	**86.3**	84.2	92.4	**98.0**	**96.4**	**72.0**	**86.7**	**83.0**

that CoOp and ProGrad are outperformed by a large margin. The strongest competitor is PLOT++. PLOT++ necessitates a two-stage training (each of 50 epochs for ImageNet) as well as several dataset-specific textual templates for their optimal transport formulation, reducing its portability to other downstream tasks. Overall, CLIP-LoRA performs better, especially on ImageNet, UCF101 and Aircraft, while being more practical. However, it underperforms on two datasets, Food101 and OxfordPet, where few-shot learning offers minimal improvement. This may be attributed to the lack of regularization, considering we use straightforward cross-entropy loss. We observe a similar trend with CoOp, whereas approaches that incorporate explicit regularization, such as ProGrad, do not exhibit this issue. Note that more detailed results, including for 2 and 8 shots, are available in the Appendix of the CLIP-LoRA paper [1].

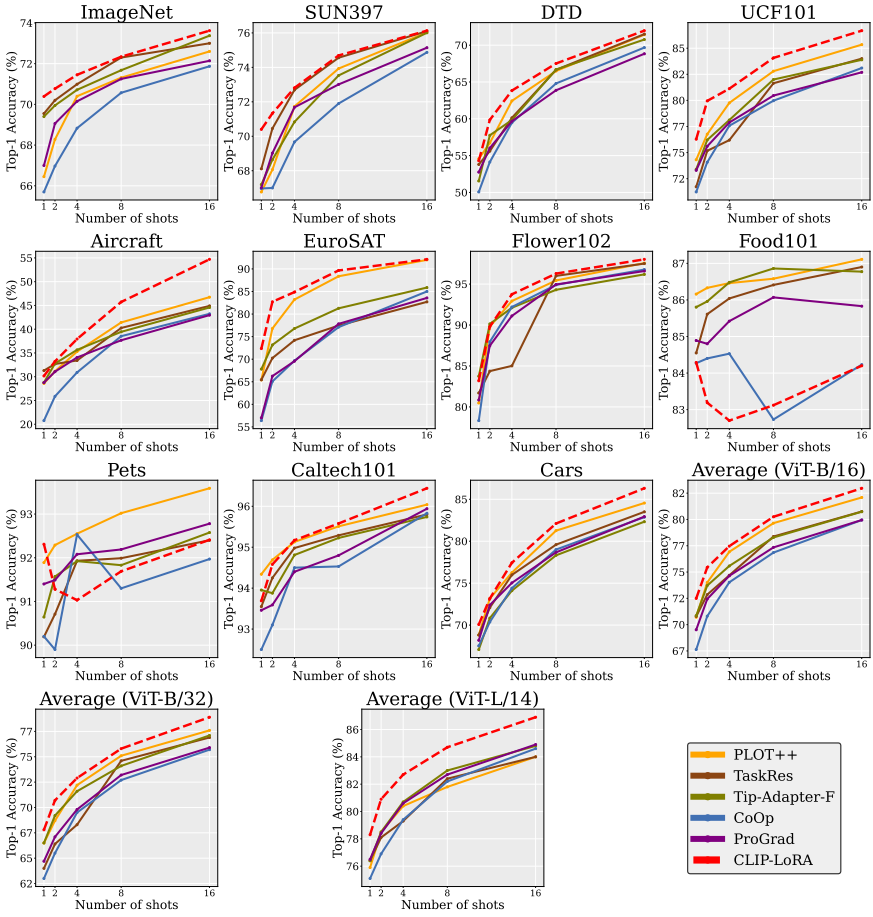


Fig. 3.2 Detailed few-shot learning results on the 10 fine-grained datasets and ImageNet with the ViT-B/16 visual backbone. Average performance for the ViT-B/16, ViT-B/32 and ViT-L/14 on the same 11 datasets is reported in the last three plots, respectively.

Vision encoders

As depicted in Figure 3.2, CLIP-LoRA surpasses, on average, the other few-shot methods with both the ViT-B/32 architecture and the larger ViT-L/14. This further supports the versatility of our approach. Detailed results for the three backbones are available in the Appendix of the CLIP-LoRA paper [1].

Table 3.2 Training time on 16-shot ImageNet task. Experiments were conducted on a single A100 80GB with the original code provided by the authors. For PLOT++ the time reported includes the 2 training stages.

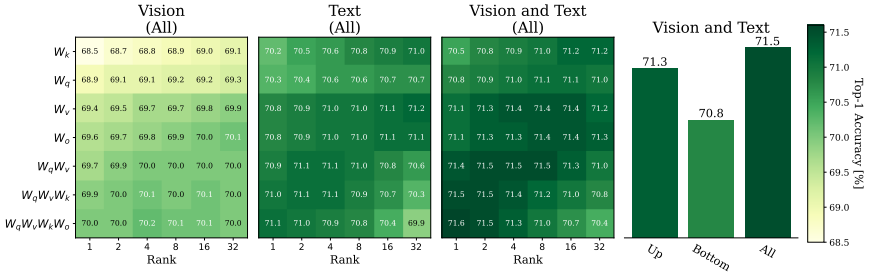
Method	Training time
CoOp	2h
PLOT++	15h30
ProGrad	3h20
CLIP-LoRA	50 min.

Computational efficiency
Table 3.2 compares the training time of the leading prompt-learning methods; CLIP-LoRA achieves better performance with shorter training. Moreover, the best performing adapter method, namely Tip-Adapter-F, depends on a large cache model that stores embeddings for all instances across every class. In contrast, LoRA merges its adapted matrices at the inference stage, thereby eliminating the need for extra memory beyond what is required by the original model.

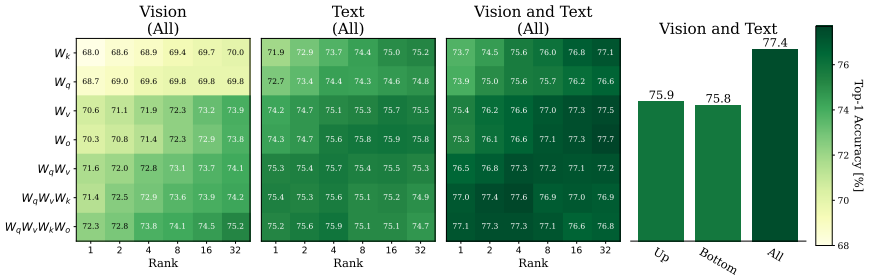
3.5 Ablation Studies

Encoder selection
With the exception of EuroSAT, where adapting solely the vision encoder shows marginally better stability, tuning both encoders concurrently is the most effective strategy, leading to significant enhancements. This aligns with recent approaches that incorporate additional vision tokens [45, 58] to augment performance beyond text-only prompt tuning, as seen in CoOp [25].

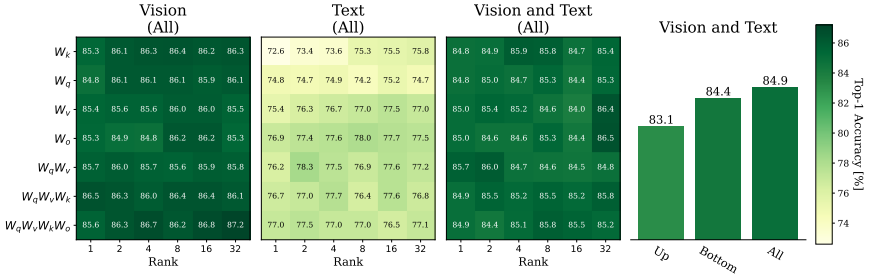
Target matrices
Among the four attention matrices studied, adapting value or output matrices ($\mathbf{W}_V$ and $\mathbf{W}_O$) appears to be the best strategy, showing quite consistent differences in performance. Moreover, as discussed in the original LoRA paper and subsequent works [52, 55], adapting a larger number of weight matrices can lead to better results. However, it can also decrease performance, as demonstrated on ImageNet and StanfordCars with high rank. This is in line with recent methods that aim to dynamically adjust the rank of the matrices [54, 55].



(a) ImageNet



(b) Stanford Cars



(c) EuroSAT

Fig. 3.3 Top-1 accuracy with 4-shot for different matrices of the attention bloc and increasing rank, when the low-rank matrices are positioned at every level of the encoders (All). The fourth bar plot study the impact of positioning the low-rank matrices only on the half last levels (Up), the first half levels (Bottom), or at every level (All).

Layer-wise placement

Finally, the impact of LoRA module placement (on the lower half (bottom) or the upper half (up)) is illustrated in the bar plots of Figure 3.3 with varying performance and no clear winner. We found it more effective to add

LoRA modules across all layers. In comparison, in the context of LLMs, AdaLoRA [55] suggests that allocating a larger rank to the middle and last layers rather than the first ones yields better results. Similar strategies applied for VLMs could reveal promising avenues for future research.

3.6 Extension to Satellite Imagery

Remote Sensing (RS) imagery has become essential in a wide range of applications, ranging from environmental monitoring, to data-driven farming, as well as emergency disaster response [77–79]. These diverse applications require swift and precise scene classification to extract valuable insights from available visual data. Early works in RS scene classification primarily relied on supervised learning methods [80]. Despite their success, supervised learning methods face significant limitations, such as poor generalization capabilities. Moreover, supervised learning requires extensive carefully labeled datasets for training which is a big issue when considering the large-scale nature of RS imagery. As a result, researchers have shifted focus towards alternative strategies that aim to reduce dependency on extensive annotations and improve the scalability of RS image scene classification models [81–84].

While the above few-shot adaptation methods have been extensively explored on natural image classification on the base CLIP model, no work has provided a structured benchmark for few-shot adaptation on RS-specific tasks such as scene classification. We introduce the first structured benchmark to evaluate and compare RS-VLMs under a few-shot setting; performing extensive experiments across ten RS datasets, three specialized vision-language models, and five state-of-the-art few-shot adaptation techniques including our proposed CLIP-LoRA.

3.6.1 Experimental Setup

Datasets

The few-shot benchmark is composed of ten RS scene classification datasets: AID (AID) [85], EuroSAT (SAT.) [68], MLRSNet (MLRS) [86], OPTIMAL31 (OPT.) [87], PatternNet (Pat.) [88], RESISC45 (RES.) [89], RSC11 (RSC.) [90], RSICB128 (R128) [91], RSICB256 (R256) [91], and WHURS19 (WHU.) [92].

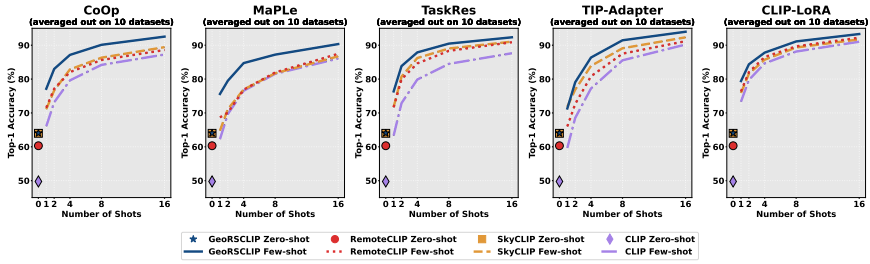


Fig. 3.4 Performance evaluation of four vision-language models (**GeoRSCLIP**, **RemoteCLIP**, **SkyCLIP**, and **CLIP**) using five different few-shot adaptation methods. Results are computed with ViT-B/32 backbone and represent the average performance across the ten benchmark datasets.

These datasets cover a wide range of sizes, with sample counts ranging from $\sim 10^3$ to $\sim 10^5$ per dataset. They also include varying numbers of classes, from coarse-grained classification with 10 classes to fine-grained scene classification with up to 46 classes.

Models

We implemented three RS-VLMs in our benchmark: RemoteCLIP [93], GeoRSCLIP [94], SkyCLIP [95], as well as the original CLIP [16]. These three RS-VLMs are all fine-tuned variants of the original CLIP. By including this variety, we aim to assess how different pretraining data influences performance on RS scene classification tasks. All four models are tested with RS-specific text-prompt templates of the form `"a satellite photo of a [class]."` to generate text embeddings.

Comparative methods

In our proposed benchmark, we assess few-shot adaptation using five methods: CoOp [25], MaPLe [58], TaskRes [50], Tip-Adapter [27], and CLIP-LoRA [1]. These popular methods encompass a diverse range of adaptation techniques, including prompt tuning (which introduces learnable tokens as new trainable parameters), adapter-based strategies (which insert additional trainable layers, typically at the output of the model), and low-rank fine-tuning (which incorporates learnable low-rank matrices into the intermediate weights). This diversity of techniques provides valuable insight into which parameters are most relevant to tune. For each method, we adopt the original hyperparameter settings specified in their respective publication.

Table 3.3 Detailed results of five different few-shot adaptation methods on GeoRSCLIP ViT-B/32 backbone. Experiments are conducted the ten benchmark datasets for shot values of 1, 2 and 4. For each scenario, the highest value is marked in **bold** and the second highest value is underlined.

	Method	AID	SAT	MLRS	OPT.	Pat.	RES.	RSC.	R128	R256	WHU.	Average
	GeoRSCLIP	70.9	53.2	64.9	79.6	76.1	69.1	65.9	28.8	45.1	87.0	64.1
1-shot	CoOp	80.0	68.9	69.0	83.6	88.1	74.0	82.5	61.7	72.8	90.8	77.1
	MaPLe	79.8	64.3	69.7	85.2	86.1	77.5	79.2	57.3	63.9	92.8	75.6
	TaskRes	**81.7**	71.8	**73.7**	**88.0**	**89.5**	**79.9**	83.7	64.9	70.5	**93.5**	**79.7**
	Tip-Adapter	75.0	60.1	69.9	83.0	85.6	74.7	73.8	40.7	58.8	91.5	71.3
	CLIP-LoRA	75.8	**73.4**	70.4	87.0	89.1	75.9	**85.0**	**66.8**	**77.2**	**93.5**	79.4
2-shot	CoOp	**86.1**	76.0	74.7	87.2	91.3	79.7	86.6	72.5	81.5	94.5	83.0
	MaPLe	82.9	75.4	74.0	86.4	88.3	80.2	83.4	57.9	71.3	95.4	79.5
	TaskRes	84.5	76.5	**77.9**	**89.1**	**91.7**	**83.6**	84.7	68.5	75.6	93.7	82.6
	Tip-Adapter	81.3	71.5	73.0	86.2	90.0	80.0	81.9	58.1	72.6	96.1	79.1
	CLIP-LoRA	80.0	**83.3**	73.2	88.2	90.9	79.2	**88.1**	**77.9**	**86.6**	**96.3**	**84.4**
4-shot	CoOp	87.9	85.7	77.6	89.0	93.0	82.3	89.1	82.7	87.8	96.4	87.1
	MaPLe	86.3	84.7	77.0	89.5	92.1	83.6	89.0	67.7	80.9	96.4	84.7
	TaskRes	87.8	84.4	**79.4**	**91.4**	92.8	**85.2**	88.0	71.5	78.4	95.0	85.4
	Tip-Adapter	**89.3**	80.0	77.9	90.2	93.6	84.3	88.1	77.3	85.7	**97.2**	86.4
	CLIP-LoRA	83.8	**88.3**	76.8	90.2	**93.9**	81.8	**89.2**	**85.2**	**92.5**	96.5	**87.8**

3.6.2 Results

RS-VLM Evaluation

Figure 3.4 presents the average performance of the evaluated models across the ten datasets in our benchmark. All models are compared using the same backbone (ViT-B/32), which allows us to isolate the impact of the pretraining procedure (i.e., the dataset used, objective functions, and domain alignment) from the effects of architectural differences. The results clearly show that GeoRSCLIP consistently outperforms the other models across all few-shot adaptation methods.

An interesting observation is that zero-shot performance is not always a reliable indicator of few-shot adaptation performance. For instance, while GeoRSCLIP and SkyCLIP exhibit similar zero-shot accuracy on average, GeoRSCLIP demonstrates better performance in the few-shot setting. Similarly, although SkyCLIP slightly outperforms RemoteCLIP in the zero-shot

Table 3.4 Average accuracy across the ten benchmark datasets for higher shot values of 8 and 16 using GeoRSCLIP ViT-B/32 backbone. For each scenario, the highest value is marked in **bold** and the second highest value is underlined.

Method	1-shot	8-shot	16-shot
CoOp	77.1	90.2	92.6
MaPLe	75.6	87.2	90.4
TaskRes	**79.7**	86.5	90.0
Tip-Adapter	71.3	**91.5**	**94.0**
CLIP-LoRA	79.4	91.2	93.4

setting, their few-shot performance is nearly equivalent across different values of N_{shots}. Additionally, it is evident that even one-shot adaptation leads to significant improvements over zero-shot evaluation across all four models considered.

Comparison of few-shot adaptation strategies

Table 3.3 presents detailed results across the ten datasets for the GeoRSCLIP ViT-B/32 model. While LoRA performs best on average for 1 and 2 shots, the results vary significantly depending on the dataset and the number of shots. This suggests that the most effective parameters to tune may depend on the task. For example, CoOp, which adapts the input text prompts, performs particularly well on AID. TaskRes, which modifies output text embeddings, excels on RSICB128. CLIP-LoRA, which tunes intermediate model weights, offers strong average performance overall. Additionally, Tip-Adapter tends to underperform in low-shot settings, but the performance gap narrows as the number of shots increases to four. MaPLe performs the worst among the methods studied, likely because it was primarily designed for generalization to new classes and/or domains. These differences underline the need for more robust and adaptable few-shot adaptation methods tailored to the unique challenges of RS.

This highlights the importance of selecting the most suitable adaptation strategy based on the available level of supervision. It also points to promising research directions for developing methods that can maintain strong performance across a broad range of supervision levels.

Table 3.5 Detailed results of GeoRSCLIP with various backbones. For each scenario, the highest value is marked in **bold** and the second highest value is underlined.

	Method	ViT-B/32	ViT-L/14	ViT-H/14
1-shot	CoOp	77.1	81.1	73.9
	MaPLe	75.6	78.9	67.4
	TaskRes	**79.7**	80.3	71.9
	TIP-Adapter	71.3	76.9	78.9
	CLIP-LoRA	79.4	**83.4**	**84.3**
2-shot	CoOp	83.0	86.4	80.6
	MaPLe	79.5	82.8	69.4
	TaskRes	82.6	83.6	76.8
	TIP-Adapter	79.1	82.9	83.7
	CLIP-LoRA	**84.4**	**87.2**	**87.1**
4-shot	CoOp	87.1	89.5	85.6
	MaPLe	84.7	85.9	74.5
	TaskRes	85.4	83.6	77.9
	TIP-Adapter	86.4	89.0	89.2
	CLIP-LoRA	**87.8**	**90.2**	**90.4**

Model size

Finally, in Table 3.5, we analyze how the trends observed in earlier sections evolve when scaling to larger backbones. Interestingly, TaskRes demonstrates superior performance compared to LoRA on the larger ViT-H/14 backbone (986M parameters), a stark contrast to its performance on smaller backbones such as ViT-L/14 (307M parameters) and ViT-B/32 (87M parameters). Furthermore, prompt-tuning methods like CoOp and MaPLe experience significant performance drops when scaled to the ViT-H/14 backbone. The results suggest that the configuration of TaskRes, initially developed for smaller backbones, may exhibit better generalization compared to more complex methods like LoRA or CoOp.

These results emphasize the need of evaluating performance across different backbone sizes, as the choice of backbone may be influenced by the specific practical constraints of each application.

3.7 Conclusion of Chapter 3

In this chapter, we addressed the challenge of inductive few-shot adaptation for contrastive Vision-Language Models. We introduced **CLIP-LoRA**, a method that integrates low-rank adaptation directly into the encoder architecture, offering a computationally efficient and stable alternative to prevailing prompt-tuning and adapter-based approaches. Our extensive evaluation demonstrated that CLIP-LoRA yields state-of-the-art performance across diverse benchmarks, including specialized domains like remote sensing. We showed that this method is competitive with, and in the majority of cases outperforms, existing techniques, all while relying on fixed hyperparameters for every dataset and number of shots. It is important to note that using fixed hyperparameters is not the current standard in this literature; through CLIP-LoRA and its widely accessible open-source codebase[1], we aimed to raise awareness within the community regarding this critical issue. Consequently, the constraint of achieving generalization without dataset-specific hyperparameters will serve as a central theme for this thesis, guiding the design of the test-time adaptation and transductive methods developed in the next chapters.

[1] https://github.com/MaxZanella/CLIP-LoRA

4

Test-Time Adaptation from a Single Sample

Preliminary Note:

This chapter expands on the conference proceeding [4], presenting a novel **MeanShift**-inspired algorithm for **test-time augmentation** that jointly optimizes a density mode and an *inlierness score* for each augmented view. Mathematical details regarding convergence are provided in the Appendix (Section 4.6).
Before starting the chapter, readers are encouraged to familiarize themselves with the basic principles of CLIP (Chapter 2) and the overview of the few-shot adaptation literature (Section 3.2.2). Additionally, familiarity with the Concave-Convex Procedure (CCCP) [96] is recommended.

4.1 Introduction

Following our investigation of the *fine-tuning* phase in the previous chapter, a natural progression is to now address the *inference* stage. In this chapter, we focus specifically on *inductive single-sample adaptation*, exploring how to enhance the accuracy of either (i) a zero-shot model or (ii) a model previously tuned via inductive few-shot learning, using only the test instance at hand. This approach stands in contrast to the methods presented in later chapters, which target prediction improvements over an entire batch of data simultaneously (i.e., *transductive* inference). By showing that robust performance improvements can be realized on a single image without heavy computation, this chapter sets the stage for a broader exploration of efficient test-time adaptation strategies.

4.1.1 Context and Research Positioning

The joint feature space of visual and textual features enables zero-shot recognition, without any task-specific data. For instance, given a set of candidate classes, one can create textual class descriptions (e.g., "`a photo of a [class`$_k$`]`"). This text is then converted into an input embedding sequence $\mathbf{c}_k$, which is mapped to a final embedding representation $\mathbf{t}_k = \theta_t(\mathbf{c}_k)$ via the language encoder. Similarly, an image $\mathbf{x}$ is projected in the same embedding space $\mathbf{f} = \theta_v(\mathbf{x})$ using the visual encoder. Then, one can classify this image by measuring the similarity between these two encoded modalities and predicting the class corresponding to the most similar embedding, $\hat{k} = \arg\max_k \mathbf{f}^\top \mathbf{t}_k$.

Despite their impressive capabilities, these models still encounter substantial challenges and may yield unsatisfactory responses in complex situations [16,97]. These issues are particularly pronounced when confronted with the pragmatic constraints of real-world scenarios, where labeled data can be scarce (i.e., few-shot scenarios) or completely absent (i.e., zero-shot scenarios [98]), thus limiting their broader usage. Consequently, there has been a growing interest in enhancing the test-time generalization capabilities of these vision-language models [47,48,99–101].

As discussed in Chapter 3, empirical findings that improved textual descriptions can positively impact zero-shot predictions have sparked in-

terest in refining prompt quality for downstream tasks. Soft prompt learning, which utilizes learnable continuous tokens as input has rapidly gained popularity [25]. Since then, prompt tuning has appeared as the prominent approach for adapting vision-language models [102] across both unsupervised [46–48] and few-shot scenarios [25, 26, 40–45].

In parallel, test-time augmentation, which has been extensively used in the computer vision community [103, 104], is now emerging in the vision-language field, with a focus on prompt tuning [47, 48]. Instead of exploiting a single image $\mathbf{x}$, *test-time prompt tuning* techniques leverage multiple embeddings $(\mathbf{f}_p)_{1\leq p\leq N}$, each derived from a different augmented view $(\mathbf{x}_p)_{1\leq p\leq N}$ of the same original image $\mathbf{x}$. Afterwards, the prompt is optimized by forcing consistency of the predictions among these different views [47]. The final classification step is then performed by computing the similarity between the original image encoding and the optimized textual embedding $\mathbf{t}_k^*$, $\hat{k} = \arg\max_k \mathbf{f}^\top \mathbf{t}_k^*$.

These novel research directions underscore the increasing attention in enhancing these models' robustness, especially in zero-shot scenarios, through data augmentation at test-time. Alongside this expanding literature, we ask the following question: *Can we improve the image representation* $\mathbf{f}$ *directly in the embedding space, achieving superior results in a way that is more efficient than prompt tuning?*

Concurrently, there has been a surge in the use of proprietary and closed APIs that encapsulate advanced machine learning functionalities, often termed *black boxes* due to their limited transparency, offering little insight into their internal mechanisms or architectures. Yet, they are crucial in executing a wide spectrum of tasks in vision and NLP, introducing new challenges in model adaptation [105]. The field of NLP, in particular, has seen an emerging literature on few-shot adaptation of black-box models [106], driven by the reality that large-scale models (e.g., GPT family [107, 108], Palm [109]) are only accessible via APIs and their pretrained weights are not publicly available. Optimizing prompts, which necessitates gradient computation from output back to input, a memory-intensive and time-consuming process, is impractical in the context of API-reliant applications. In contrast, our approach does not require extra assumptions about the model's internal states or architecture, making it suitable for black-box applications.

4.1.2 Summary of Contributions

We introduce a robust multi-modal **M**eanShift **T**est-time **A**ugmentation (**MTA**), which enhances the zero-shot generalization of CLIP models, leveraging different augmented views of a given image. Our key contributions are as follows:

(i) We propose a robust MeanShift formulation, which automatically manages augmented views in test-time augmentation scenarios by optimizing *inlierness* variables. Used as a versatile plug-and-play tool, MTA improves the zero-shot performance of various models on a large variety of classification tasks, without any hyperparameter tuning.

(ii) We report comprehensive evaluations and comparisons to the existing test-time prompt tuning techniques on 15 datasets, showing MTA's highly competitive performances, although it operates in limited access (i.e., final embedding) and training-free mode. This makes MTA suitable for both standalone and API-based applications.

(iii) Deployed easily atop current state-of-the-art few-shot learning methods, MTA brings consistent improvements, a benefit not observed with test-time prompt tuning.

4.1.3 Chapter Outline

The remainder of this chapter is structured as follows. Section 4.2 introduces the concept of test-time augmentation as a methodology to enhance model robustness by generating multiple variations of a single sample at inference. Next, we detail our core contribution: the robust multi-modal MeanShift (MTA) algorithm, explaining its foundational formulation and the block coordinate descent optimization procedure. Section 4.3 validates the effectiveness of MTA with extensive experiments, showcasing its superior performance over prompt tuning in zero-shot and few-shot settings across 15 datasets. Section 4.4 provides ablation studies on the key components of MTA. The chapter concludes in Section 4.5 with a summary of key findings regarding MTA's efficiency and plug-and-play applicability for single sample adaptation. Finally, the Appendix (Section 4.6) provides the necessary mathematical proofs for the convergence of the proposed algorithm.

4.2 Robust Multi-Modal MeanShift

Test-time augmentation

Data augmentation during training is widely recognized for its capacity to enhance model robustness [110, 111]. Also, its utility extends to test-time applications [103, 104]. In particular, test-time augmentation can be used on a single image [103] to adapt models with an entropy minimization term. The latter is often used in the context of unsupervised adaptation, but is deployed differently in this augmentation setting, enforcing consistent predictions across the various augmented views.

This idea is further developed for vision-language models with test-time prompt tuning (TPT) [47], where a prompt is optimized to make consistent predictions among light augmentations inspired by Augmix [110]. DiffTPT [48] builds up on this work by adding generated images from Stable Diffusion [112] to acquire more diverse views. Both works show improvements when selecting a subset of the augmented views. Specifically, TPT utilizes only the 10% most confident views, and DiffTPT measures the similarity with the original image, keeping the unconfident but correctly classified augmentations. We also demonstrate that filtering the augmented views can substantially improve test-time augmentation techniques. Additionally, our method does not rely on arbitrary hard thresholds; instead, we directly integrate the weighting of the augmented views in our optimization procedure thanks to *inlierness* variables.

Problem formalization

Similarly to the test-time generalization setting recently introduced in TPT [47], let us assume that we are given a set of image samples $(\mathbf{x}_p)_{1\leq p\leq N}$, which correspond to N distinct augmented views of a given test sample $\mathbf{x}$. It is important to note that our method is applicable on top of any type of augmentations. Let $\mathbf{f}_p = \theta_v(\mathbf{x}_p)$ denote the vision-encoded feature embedding corresponding to augmented sample $\mathbf{x}_p$, θ_v being the vision encoder of the CLIP pre-trained model.

A straightforward way to use the ensemble of augmentations is to perform the zero-shot prediction based on their mean embedding, thereby giving exactly the same importance to all augmented samples, independently

of the structure of the data. However, the augmentations may include degenerated views, which correspond to *outliers*, e.g., in the form of isolated data points or small regions with little structure within the feature space. Such outliers may bias global statistics like the mean.

4.2.1 Formulation

We hypothesize that *robust statistics* like the modes of the density of the set of augmented views could provide better representations. Augmented views presenting major characteristics of the concept to be recognized are likely to be projected close to the original image and close to each other. This motivates our formulation, which could be viewed as a novel robust and multi-modal extension of the popular MeanShift algorithm [113], an unsupervised procedure for finding the modes of the distribution of a given set of samples. In our case, we explicitly model and handle the potential presence of outliers in the estimation of the kernel density of the set of feature vectors $(\mathbf{f}_p)_{1\leq p\leq N}$. To do so, we introduce a latent assignment vector $\mathbf{u} = (u_p)_{1\leq p\leq N} \in \Delta^{N-1}$, with $\Delta^{N-1} = \{\mathbf{u} \in [0,1]^N \mid \mathbf{1}^\top \mathbf{u} = 1\}$ the probability simplex, and propose to minimize the following objective function:

$$\min_{\mathbf{m},\mathbf{u}} \mathcal{L}(\mathbf{m},\mathbf{u}) \quad \text{s.t.} \quad \mathbf{u} \in \Delta^{N-1} \quad \text{with}$$

$$\mathcal{L}(\mathbf{m},\mathbf{u}) = -\sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}) - \frac{\lambda}{2}\sum_{p,q} w_{p,q} u_p u_q - \lambda_{\mathbf{u}} H(\mathbf{u}) \tag{4.1}$$

In the following, we describe the notations occurring in our model in Eq. (4.1), as well as the effect of each of its terms:

Robust KDE (first term)
K is a kernel function measuring a robust affinity between $\mathbf{f}_p$ and $\mathbf{m}$, e.g., a Gaussian kernel [114]: $K(\mathbf{f}_p - \mathbf{m}) \propto \exp(-\|\mathbf{f}_p - \mathbf{m}\|^2/h^2)$, where h is the kernel bandwidth. When variables u_p are fixed to 1 $\forall p$, the first term reduces to the kernel density estimate (KDE) of the distribution of features at point $\mathbf{m}$. Clearly, minimizing this term w.r.t $\mathbf{m}$ yields the standard MeanShift algorithm for finding the mode of the density (i.e., the point maximizing it). In our model, the additional latent variable u_p evaluates the *inlierness* of the p^{th} augmented view, i.e., the model's belief in $\mathbf{f}_p$ being an inlier or not within the whole set of augmented-view embeddings $(\mathbf{f}_p)_{1\leq p\leq N}$.

Score $u_p \in [0,1]$ is high when the model considers the p^{th} sample as an inlier, enabling it to contribute more in the KDE evaluation in (4.1), and small (closer to 0) otherwise.

Text-knowledge guided quadratic term (second term)
This term encourages samples with nearby text-based zero-shot predictions to have similar *inlierness* scores u_p. Specifically, we construct the pairwise affinities $w_{p,q}$ in the second term of (4.1) from both the text and vision embeddings as follows. Let $\mathbf{s}_p \in \mathbb{R}^K$ denote the text-driven softmax prediction based on the zero-shot text embedding for the p^{th} sample, i.e., the k^{th} component of $\mathbf{s}_p$ is given by:

$$s_{p,k} = \frac{\exp(l_{p,k}/\tau)}{\sum_{j=1}^{K}(\exp l_{p,j}/\tau)}; \; l_{p,k} = \mathbf{f}_p^\top \mathbf{t}_k \tag{4.2}$$

where τ is the temperature scaling parameter of the CLIP model. Affinities $w_{p,q}$ are given by:

$$w_{p,q} = \mathbf{s}_p^\top \mathbf{s}_q \tag{4.3}$$

The Shannon entropy (third term)
$H(\mathbf{u})$ is the Shannon entropy defined over simplex variables as follows:

$$H(\mathbf{u}) = -\sum_{p=1}^{N} u_p \ln u_p \tag{4.4}$$

This term acts as a barrier function, forcing latent variable $\mathbf{u}$ to stay within the probability simplex. Furthermore, this entropic regularizer is necessary to avoid a trivial solution minimizing the quadratic term (i.e., $u_p = 1$ $\exists p$.), as it pushes the solution toward the middle of the simplex (i.e., $u_p = 1/N \;\; \forall p$).

4.2.2 Block coordinate descent optimization

Our objective in (4.1) depends on two types of variables: $\mathbf{m}$ and $\mathbf{u}$. Therefore, we proceed with block-coordinate descent alternating two sub-steps: one optimizing (4.1) w.r.t $\mathbf{u}$ and keeping density modes $\mathbf{m}$ fixed, while the other minimizes (4.1) w.r.t $\mathbf{m}$ with the *inlierness* variables fixed.

Optimization w.r.t y via the concave-convex procedure

When $\mathbf{m}$ is fixed, our objective $\mathcal{L}(\mathbf{u}, \mathbf{m})$ could be minimized efficiently w.r.t $\mathbf{u}$ using the Concave-Convex Procedure (CCCP) [96], with convergence guarantee. At each iteration, we update the current solution $\mathbf{u}^{(n)}$ as the minimum of a tight upper bound on $\mathcal{L}$, which ensures the objective does not increase. For the sum of concave and convex functions, as for our sub-problem, the CCCP replaces the concave part by its linear first-order approximation at the current solution, which is a tight upper bound, while keeping the convex part. For (4.1), the quadratic term could be written as $\mathbf{u}^\top W \mathbf{u}$, with $W = [w_{p,q}]$. It is easy to see that this term is concave as affinity matrix W is positive semi-definite, whereas the remaining part of $\mathcal{L}$ is convex. Therefore, we replace this quadratic term by $\mathbf{u}^\top W^\top \mathbf{u}^{(n)}$, obtaining, up to an additive constant, the following tight bound:

$$\mathcal{L}(\mathbf{u}, \mathbf{m}) \overset{c}{\leq} -\sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}) - \lambda \mathbf{u}^\top W^\top \mathbf{u}^{(n)} - \lambda_{\mathbf{u}} H(\mathbf{u}) \tag{4.5}$$

Solving the Karush-Kuhn-Tucker (KKT) conditions for minimizing bound (4.9), s.t. simplex constraint $\mathbf{u} \in \Delta^{N-1}$, gives the following updates for $\mathbf{u}$:

$$u_p^{(n+1)} = \frac{\exp\left((K(\mathbf{f}_p - \mathbf{m}) + \lambda \sum_{q=1}^{N} w_{p,q} u_q^{(n)}) / \lambda_{\mathbf{u}} \right)}{\sum_{j=1}^{N} \exp\left((K(\mathbf{f}_j - \mathbf{m}) + \lambda \sum_{q=1}^{N} w_{j,q} u_q^{(n)}) / \lambda_{\mathbf{u}} \right)} \tag{4.6}$$

which have to be iterated until convergence. The complete derivation of Eq. (4.6) is provided in Appendix (Section 4.6.1).

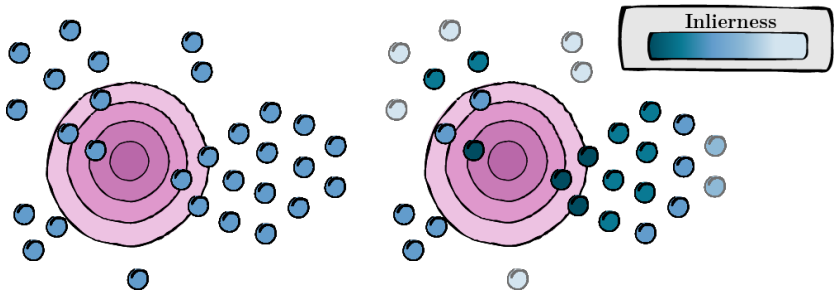
Optimization w.r.t m via fixed-point iterations

This sub-step fixes $\mathbf{u}$, and minimizes the objective in (4.1) w.r.t the density modes $\mathbf{m}$. Setting the gradient of $\mathcal{L}$ w.r.t $\mathbf{m}$ to 0 yields the following necessary condition for a minimum, taking the form of a fixed-point equation:

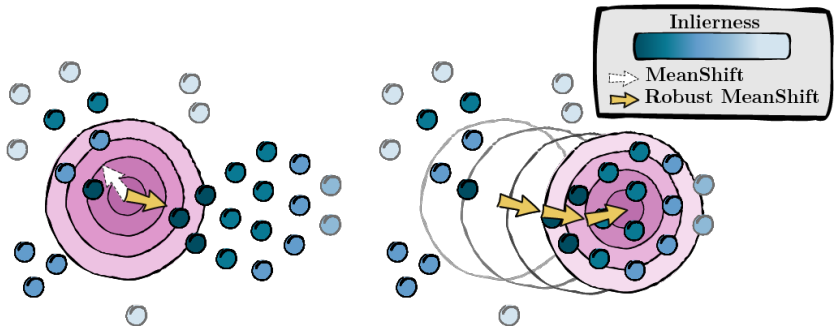
$$\mathbf{m} - g(\mathbf{m}) = 0; \quad g(\mathbf{m}) = \frac{\sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}) \mathbf{f}_p}{\sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m})} \tag{4.7}$$

The solution to (4.7) could be obtained by the following fixed-point iterations:

$$\mathbf{m}^{l+1} = g(\mathbf{m}^l) = \frac{\sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}^l) \mathbf{f}_p}{\sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}^l)} \tag{4.8}$$



(a) ***Inlierness scores distribution.*** Blue gradation represents the *inlierness* score: the darker, the higher.



(b) ***Inlierness-density mode seeking.*** The white arrow represents the update direction of the traditional MeanShift; instead our robust MeanShift follows the orange ones.

Fig. 4.1 Interpretation of Eqs. (4.6) and (4.8) in (a) and (b) respectively. Our robust MeanShift for test-time augmentation (MTA) alternatively solves these 2 equations until convergence.

This yields a Cauchy sequence $\{\mathbf{m}^l\}_{l\in\mathbb{N}}$, which converges to a unique value: $\mathbf{m}^* = \lim_{l\to\infty} \mathbf{m}^{l+1} = \lim_{l\to\infty} g(\mathbf{m}^l) = g(\lim_{l\to\infty} \mathbf{m}^l) = g(\mathbf{m}^*)$, and $\mathbf{m}^*$ is the unique solution of the fixed-point in Eq. (4.7) (Appendix 4.6.2).

Final prediction

The class prediction is computed by the cosine similarity between the mode minimizing our objective in Eq. (4.1), i.e., $\mathbf{m}^*$ obtained at convergence, and the encoded prompts of each class k, i.e., $\mathbf{t}_k$:

$$\hat{k} = \arg\max_k \, (\mathbf{m}^*)^\top \mathbf{t}_k$$

Interpretation

Eqs. (4.6) and (4.8) can be nicely interpreted in Figure 4.1. Sub-figure 4.1a shows how the *inlierness* scores are spreading. A data point is given a high *inlierness* score if it is close to the mode and/or close to other data points with high *inlierness* scores. Therefore, *inlierness* scores are spreading iteratively from data points close to the mode toward other data points controlled by their affinity relations. Sub-figure 4.1b shows the update direction followed by traditional MeanShift, i.e., updates (4.8) but with fixed $u_p = 1, \forall p$. The orange arrows follow the update of our robust MeanShift, i.e. joint updates in (4.6) and (4.8). Our mode update is directed toward dense regions with high *inlierness* scores, thus avoiding close regions with few or isolated data points.

Implementation details.

In the MeanShift algorithm, the bandwidth is a sensitive hyperparameter that can cause the algorithm to become stuck in small, locally dense areas if set too low, or escape significant dense regions if set too high [115]. We utilize the Gaussian kernel [114] as described in Section 4.2 and adopt a variable bandwidth [115] in which each point is assigned a unique bandwidth value h_p^2. The bandwidth of a point is estimated with a ratio ρ of its closest neighbors: $h_p^2 = \frac{1}{\rho(N-1)} \sum_{q \in I_p} \|\mathbf{f}_p - \mathbf{f}_q\|^2$ with ρ set to 0.3 inspired by the implementation of Scikit-learn [116]. Here, I_p represents the set of indices corresponding to the neighbors of point p. Initial guess of the mode can also impact the final solution and is set to the embedding of the original (i.e., non-augmented) image. Affinities are based on the text prediction as described in Section 4.2. Finally, λ and $\lambda_{\mathbf{u}}$ are set to 4 and 0.2 respectively and remain fixed for every CLIP visual encoder and dataset. This leads to a hyperparameter-free method across all experiments.

4.3 Experimental Validation on Natural Images

4.3.1 Experimental Setup

Datasets

We follow conventional benchmarks composed of 11 datasets for fine-grained classification, namely ImageNet (I.) [17], SUN397 (SUN) [69], Aircraft (Air.) [70], EuroSAT (SAT) [68], StanfordCars (Cars), Food101 (Food) [71],

OxfordPets (Pets) [72], Flowers102 (Flow.) [73], Caltech101 (Calt.) [74], DTD (DTD) [75], UCF101 (UCF) [76]. We also report results on four ImageNet variants: Adversarial (Adv.) [117], V2 (V2) [118], Rendition (Rend.) [111] and Sketch (Sketch) [119].

Models

We mainly report performance for the CLIP ViT-B/16 backbone [16]. Summarized results for zero-shot ResNet-50, ResNet-101, ViT-B/32, and ViT-L/14 are provided in Table 4.4.

Evaluation protocol

We evaluate our method under two scenarios: (i) zero-shot models, where adaptation methods are applied on top of the original CLIP model typically with the basic "`a photo of a` [$\texttt{class}_k$]" as prompt; (ii) few-shot models, where adaptation methods are applied on top of few-shot adapted CLIP model. In our experiments, we choose CoOp for its wide popularity, meaning that we use the original CLIP model with 16-shot ImageNet tuned text tokens (more details in the next paragraph). Because of the randomness inherent in test-time augmentation or some training procedures (e.g., prompt tuning), each reported performance is the averaged top-1 accuracy on the test set with three different random seeds.

Comparative methods

We employ the term ensemble for the majority vote among the 80 predefined handcrafted prompts of CLIP [16]. For the zero-shot scenario, TPT is performed with one step as suggested in their work [47]. We evaluate the zero-shot setting [98] with the standard prompt template "`a photo of a` [$\texttt{class}_k$]" as prompt initialization and with the ensemble for final prediction if mentioned. Note that combining ensemble with TPT is not straightforward as the goal is to use the optimized prompt for prediction, hence we only use ensemble with our approach. The weights of CoOp [25] are from 16-shot ImageNet with 4 tokens. For the few-shot scenario, the number of tokens of CoOp [25] and ProGrad [26] are specified in the caption of each table. As proposed in [120], to establish a fair comparison between few-shot methods, we limit the number of validation samples to $\min(N_{\text{shots}}, 4)$, notably for Tip-Adapter and Tip-Adapter-F [27] that tune hyperparameters.

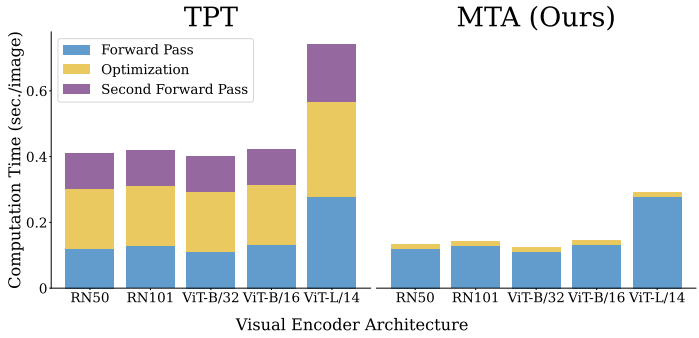


Fig. 4.2 Runtime in seconds per image on ImageNet for TPT and MTA with 5 different backbones: RN50 (ResNet-50), RN101 (ResNet-101), ViT-B/32, ViT-B/16 and ViT-L/14. Experiences were performed on a single A100 40GB GPU.

Table 4.1 Zero-shot methods on ImageNet datasets. We use "`a photo of a` [`class`$_k$]" as prompt. Majority vote of the 80 handcrafted prompts of Table 8.2 is used for final prediction when Ensemble is specified. CoOp is a pretrained prompt on 16-shots ImageNet. We highlight the best and second best results by **bolding** and underlining them, respectively. ✓ for training-free methods at test-time, ✗ otherwise.

Method		ImageNet	-Adv.	-V2	-Rend.	-Sketch	Average
CLIP	✓	66.7	47.9	60.9	74.0	46.1	59.1
CLIP + Ensemble	✓	68.4	50.0	62.1	77.4	48.0	61.2
TPT	✗	68.9	54.6	63.4	77.0	48.0	62.4
MTA	✓	69.3	57.4	63.6	76.9	48.6	63.2
MTA + Ensemble	✓	**70.1**	**58.1**	**64.2**	**78.3**	**49.6**	**64.1**
CoOp	✓	71.5	49.7	64.2	75.2	48.0	61.7
TPT + CoOp	✗	73.6	57.9	66.7	78.0	49.6	65.1
MTA + CoOp	✓	**74.0**	**59.3**	**67.0**	**78.2**	**50.0**	**65.7**

Data augmentation

Our proposed method uses random cropping (RandomCrop) as an augmented view generator identical to the one in TPT [47]. We also study at the end of Section 4.3.2 the impact of a more complex data augmentation based on diffusion as done in DiffTPT [48]. Note that TPT and DiffTPT are using slightly stronger augmentations for the 10 fine-grained classification datasets inspired by AugMix [110]. In our case, for a more realistic zero-shot setting, we keep the simple RandomCrop for all the 15 datasets.

Table 4.2 Zero-shot methods on 10 fine-grained classification datasets. We highlight the best result by **bolding**. +E. means that majority vote with the 80 handcrafted prompts of Table 8.2 is used for final prediction.

Method	SUN	Air.	SAT	Cars	Food	Pets	Flow.	Calt.	DTD	UCF	Average
CLIP	62.6	23.7	42.0	65.5	83.7	88.3	67.4	93.4	44.3	65.1	63.6
CLIP + E.	66.0	23.9	**48.8**	66.1	83.8	**88.4**	67.8	93.9	46.0	66.8	65.2
TPT	65.4	23.1	42.9	66.4	84.6	87.2	**68.9**	94.1	**47.0**	68.0	64.8
MTA	65.0	25.3	38.7	68.1	85.0	88.2	68.3	94.1	45.6	68.1	64.6
MTA +E.	**66.7**	**25.2**	45.4	**68.5**	**85.0**	88.2	68.1	**94.2**	45.9	**68.7**	**65.6**

Table 4.3 Comparison of augmentation strategies: RandomCrop Vs Diffusion-based. Note that the test set of each dataset is reduced to 1000 samples for computation reasons and batch size of 128 was used as in the DiffTPT paper [48] (64 randomly cropped and 63 diffusion-generated images for Diffusion). Hence reported performance can vary from Table 4.1. We highlight the best result by **bolding**.

Augmentation	Method	ImageNet	-Adv.	-V2	-Rend.	-Sketch	Average
RandomCrop	TPT	68.2	51.2	**66.2**	76.9	49.3	62.4
	MTA	**69.1**	**55.3**	65.7	**77.5**	**50.2**	**63.6**
Diffusion	DiffTPT	67.8	53.4	**65.2**	**76.9**	50.2	62.7
	MTA	**69.2**	**54.5**	64.8	76.8	**51.1**	**63.3**

4.3.2 Results on Zero-Shot Models

Comparison with test-time prompt tuning

Table 4.1 demonstrates MTA's stable improvement over TPT on ImageNet and its variants with both the basic "`a photo of a` [`class`$_k$]" and CoOp's pretrained prompt. Moreover, Table 4.2 indicates that, except for the EuroSAT dataset [68], MTA consistently enhances baseline performance across fine-grained datasets. It also surpasses TPT when combined with ensembling, which respects the black-box assumption. TPT is not able to outperform the majority vote strategy on average for the 10 fine-grained classification datasets, questioning its applicability for a larger variety of tasks. Finally, we compare runtimes on ImageNet in Figure 4.2. MTA is nearly three times faster than TPT, notably due to the quick optimization step and the removed second forward pass.

Improved prompt strategy

Table 4.1 and 4.2 concur in suggesting that the improvements brought by our approach complement those from refined prompt strategies, i.e., basic

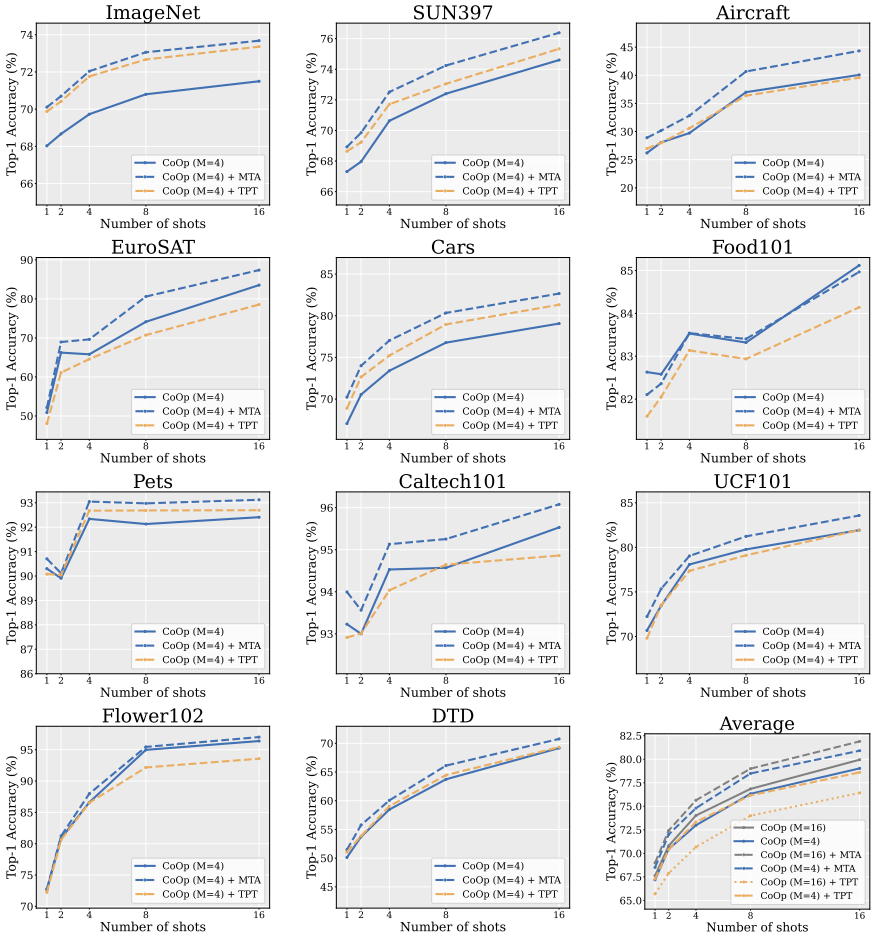


Fig. 4.3 Few-shot learning results on the 10 fine-grained datasets and ImageNet. We compare MTA and TPT when added on top of CoOp prompts (M=4 tokens) for increasing number of shots. Averaged top-1 accuracy over the 11 datasets is shown on the bottom right, we additionally show the averaged top-1 accuracy for CoOp with M=16 tokens.

ensemble of handcrafted prompts or soft pretrained prompts. This may imply that leveraging both modalities in the same optimization process, as suggested in [120] for the few-shot setting, could further enhance performance.

Table 4.4 Performance on ImageNet and its 4 variants for different visual encoders of CLIP. Hyperparameters are kept identical across datasets and visual encoder architectures. +E. stands for Ensemble.

Method	ResNet-50	ResNet-101	ViT-B/32	ViT-B/16	ViT-L/14
CLIP	44.1	49.5	50.7	59.1	70.7
Ensemble	46.2	51.6	52.2	61.1	72.6
TPT	47.2	52.2	53.3	62.4	73.7
MTA	47.2	53.2	55.2	63.2	73.9
MTA + E.	**48.4**	**54.2**	**56.1**	**64.1**	**74.7**

Domain generalization with pre-trained prompts

We use CoOp pretrained on ImageNet to measure the domain generalization ability of our method as in [25] in Table 4.1. Although the gains of CoOp over the ensembling strategy remain ambiguous, the combination of CoOp with MTA significantly enhances accuracy for all datasets. This notable improvement suggests that the mode found by MTA is able to highlight the generalization ability of pretrained prompts. This illustrates an additional use case of MTA, which can be used in synergy with fine-tuning for a specific task (e.g., prompt tuning). This aspect is emphasized in Section 4.3.3.

Visual encoders

Generalization across various model architectures is a desirable attribute of zero-shot methods, as it eliminates the need for laborious and computationally demanding hyperparameter tuning. This becomes increasingly critical in the context of the current scaling trend, where model complexity is rapidly expanding [107, 109]. Table 4.4 shows consistent improvement on the baseline for five different visual backbones of CLIP with fixed hyperparameters. It demonstrates the generalization ability of MTA across architectures and model scales.

Data augmentation strategies

We explore the compatibility of MTA with different types of data augmentation. Specifically, we follow the protocol of DiffTPT [48] to generate augmented views from cropping and diffusion model [112]. We employ a batch of 128 images composed of the original one, 63 diffusion-generated and 64 randomly cropped views. Since generating images by diffusion is

Table 4.5 Improvement of few-shot learning methods on ImageNet when MTA is added on top. CoOp and ProGrad are using 16 tokens. Δ highlights the gain. ✓ for training-free methods, ✗ otherwise.

Method		1-shot	2-shot	4-shot	8-shot	16-shot
Tip-Adapter	✓	68.9	69.2	69.8	70.2	70.5
Tip-Adapter + MTA	✓	71.1 **+2.1**	71.1 **+1.9**	71.4 **+1.6**	71.9 **+1.7**	72.1 **+1.6**
Tip-Adapter-F	✗	69.4	70.0	70.7	71.8	73.4
Tip-Adapter-F + MTA	✗	71.4 **+2.0**	71.5 **+1.5**	72.2 **+1.5**	73.2 **+1.4**	74.2 **+1.8**
CoOp	✗	65.7	67.0	68.8	70.6	71.9
CoOp + MTA	✗	67.8 **+2.1**	69.1 **+2.1**	71.1 **+2.3**	72.9 **+2.3**	74.2 **+2.3**
ProGrad	✗	67.0	69.1	70.2	71.3	72.1
ProGrad + MTA	✗	69.3 **+2.3**	71.4 **+2.3**	72.5 **+2.3**	73.7 **+2.4**	74.4 **+2.3**

more computationally intensive than RandomCrop (generating 63 images takes approximately 2 minutes on an A100 40GB GPU), we present these results separately. Inspired by the observations of DiffTPT about the reliance of the images generated by diffusion, we increase ρ to a slightly less restrictive value of 0.5 for the purpose of this experiment. Otherwise, we keep the same hyperparameters λ and λ_y and do not treat the two kinds of augmentation differently. As demonstrated in Table 4.3, our method surpasses DiffTPT on average, further evidencing its effectiveness across a broad range of applications.

4.3.3 Integration with Few-Shot Learning

Enhancing few-shot pre-trained prompts

As depicted in Figure 4.3, MTA improves CoOp performance across shots and datasets with notably Aircraft 40.07% → 44.33% (+ 4.26%), EuroSAT 83.53% → 87.38% (+ 3.85%), and Cars 79.06% → 82.66% (+ 3.6%) with 16 shots. In contrast, TPT generally diminishes performance across most datasets, highlighting its limitations in enhancing prompts for fine-grained tasks. While the accuracy for 16 tokens is on average higher for MTA and lower for TPT, we report the 4 tokens results in alignment with the TPT paper [47] to maintain fairness.

Atop few-shot learning methods

As demonstrated in Table 4.5, applying MTA to prompt-based and adapter-

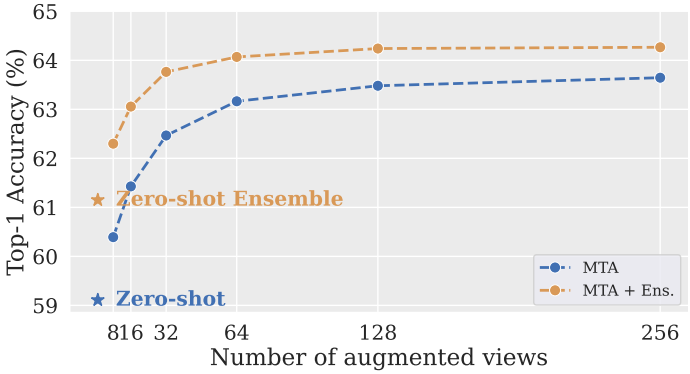


Fig. 4.4 Averaged top-1 accuracy of MTA with and without majority vote for final prediction on the 5 ImageNet variants with increasing number of augmented views.

based few-shot learning methods on ImageNet results in notable performance improvements. Specifically, prompt-based methods see an average gain exceeding 2% across shots, with adapter-based methods showing slightly lower yet significant improvements. Indeed, our affinity term, outlined in Eq. (4.3), can greatly benefit from refined prompts. Nevertheless, we can notice that a training-free approach, Tip-Adapter, combined with MTA is competitive with prompt tuning methods without MTA. This could present a compelling trade-off: opting for more intensive computational efforts during training or at test-time.

4.4 Ablation Studies

Number of augmented views

In Figure 4.4, the accuracy for MTA and MTA + Ensemble increases as the number of augmented views grows until reaching a plateau around 128. Even when we restrict the number of augmented views to 16, our method still brings about 2.3% gain, which makes it a useful tool for lighter applications, up until 4.4% gain for batches of 128 views. We can observe similar gains when combined with ensemble of prompts, with 1.9% gain for batches of 16 up until 3.1% for larger batches.

Table 4.6 Ablation studis on two main components of MTA: the *inlierness* scores and the affinity measure. Reported value is the averaged top-1 accuracy over the 15 datasets studied in this chapter.

Baseline	CLIP	62.1
FilteringStrategy	MeanShift (no *inlierness* scores)	59.9
	Confidence thresh. (10%)	63.3
	Inlierness scores	**64.1**
AffinityMeasure	Vision-based	63.2
	Text-based	**64.1**

Table 4.7 Effect of λ and $\lambda_{\mathbf{y}}$. Reported value is the top-1 accuracy on the ImageNet dataset.

λ \ $\lambda_{\mathbf{y}}$	0.01	0.05	0.1	0.2	0.4	0.8	1.6	3.2	10	100	$\to \infty$ (MeanShift)
0	66.7	66.7	66.8	68.3	65.8	65.3	65.6	65.9	66.0	66.1	66.1
0.5	66.7	66.7	66.8	68.7	67.7	66.8	66.4	66.2	66.1	66.1	-
1	66.7	66.7	66.9	68.9	68.2	67.4	66.9	66.5	66.3	66.1	-
2	66.8	66.8	67.1	69.1	68.8	68.0	67.4	67.0	66.5	66.1	-
4	66.6	66.5	66.9	**69.3**	69.1	68.6	68.0	67.5	66.8	66.2	-
8	62.0	62.5	64.2	68.7	69.3	69.0	68.5	68.1	67.2	66.3	-
16	57.3	58.5	61.0	65.8	69.1	69.3	68.9	68.5	67.7	66.4	-

Inlierness scores

We compare performance with equal weights on each augmented view (i.e., traditional MeanShift), with a confidence threshold as in TPT [47] and with our *inlierness* formulation in Table 4.6. The *inlierness* formulation yields better performance on average over the 15 datasets. Note that the relatively high score of confidence threshold is mainly due to peak performance on ImageNet-A, a trend not consistent on other datasets (see detailed results in the Appendix of the MTA paper [4]).

Affinity measure

Rather than using the affinity measure based on text described in Section 4.2, one could use only visual features $(\mathbf{f}_p)_{1\leq p\leq N}$ to compute the affinity between two embeddings $w_{p,q} = \mathbf{f}_p^t \mathbf{f}_q$. For the sake of fairness, we reperform a comprehensive hyperparameter search (λ ranging from 0.1 to 10 and $\lambda_{\mathbf{y}}$ from 0.1 to 0.5). Best performance is reported in Table 4.6. We observe that vision-based affinity measure degrades performance in comparison to text-based affinity measure.

Hyperparameter sensitivity

Table 4.7 shows the intertwined effect of λ and $\lambda_{\mathbf{u}}$ across a wide range of values. We can observe that increasing values of $\lambda_{\mathbf{u}}$ tends to perform as the MeanShift (weighting uniformly each augmented view) while as $\lambda_{\mathbf{u}}$ approaches 0, it tends toward a peak selection and trivial solutions. Best performing values are around $(\lambda, \lambda_{\mathbf{u}}) = (4, 0.2), (8, 0.4), (16, 0.8)$.

4.5 Conclusion of Chapter 4

This chapter served as an initial exploration of test-time adaptation strategies operating under minimal assumptions regarding data availability. By focusing on the single-sample regime, we demonstrated that effective accuracy improvements are achievable without requiring access to batches or external datasets. This finding establishes a strong baseline for the subsequent chapters where we will expand this scope to transductive scenarios, leveraging the availability of multiple data points to deploy more powerful procedures. Beyond its technical results, this chapter had another purpose. While we have seen the rapid emergence of multiple solutions based on prompt tuning and heavy training procedures, we demonstrated that well-designed, training-free methods can still achieve superior performance (an open-source implementation is available[1]). This highlights new opportunities to develop more efficient and transparent algorithms. In this case, we developed a novel method based on the intuition that robust statistics could lead to improved performance. Hence, the natural choice was to investigate clustering algorithms that allow for the integration of this intuition, such as MeanShift. The notable performance gains achieved by modifying the visual representation directly within the embedding space, without relying on gradient updates or task-specific prompts, challenge the field's dominant focus on prompt learning for test-time adaptation. This success has been a key motivation in the design process of our TransCLIP procedure, presented in the next chapter.

[1] https://github.com/MaxZanella/MTA

4.6 Appendix of Chapter 4

4.6.1 Proof of Equation 4.6

Here, we give details of the derivation of the $\mathbf{u}$-updates in Eq. (4.6) from the upper bound (majorizing function) in (4.9). Given a solution $\mathbf{u}^{(n)}$ at iteration n, the goal is to find the next iterate $\mathbf{u}^{(n+1)}$ that minimizes the following tight upper bound, s.t. simplex constraint $\mathbf{u} \in \Delta^{N-1}$:

$$\mathcal{B}(\mathbf{u}, \mathbf{u}^{(n)}) = -\mathbf{u}^\top \mathbf{k} - \frac{\lambda}{2}\mathbf{u}^\top W^\top \mathbf{u}^{(n)} - \lambda_{\mathbf{u}} H(\mathbf{u}) \tag{4.9}$$

where $\mathbf{k} = (K(\mathbf{f}_p - \mathbf{m}))_{1 \leq p \leq N}$.

The objective function of (4.9) is strictly convex. Taking into account the simplex constraint on $\mathbf{u}$, the associated Lagrangian reads:

$$\mathcal{L}(\mathbf{u}, \mathbf{u}^{(n)}) = -\mathbf{u}^\top \mathbf{k} - \frac{\lambda}{2}\mathbf{u}^\top W^\top \mathbf{u}^{(n)} - \lambda_{\mathbf{u}} H(\mathbf{u}) + \gamma(\mathbf{u}^\top \mathbf{1}_N - 1) \tag{4.10}$$

where γ is the Lagrange multiplier for simplex constraint $\mathbf{u} \in \Delta^{N-1}$ and $\mathbf{1}_N$ is the vector of ones. Note that we do not impose explicitly the constraints on the non-negativity of the components of $\mathbf{u}$ because these are implicitly enforced with the entropic barrier term in (4.9), i.e., $-\lambda_{\mathbf{u}} H(\mathbf{u})$. Now, computing the gradient of $\mathcal{L}(\mathbf{u}, \mathbf{u}^{(n)})$ w.r.t $\mathbf{u}$ yields:

$$\nabla_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \mathbf{u}^{(n)}) = -\mathbf{k} - \lambda W^\top \mathbf{u}^{(n)} + (\gamma + \lambda_{\mathbf{u}})\mathbf{1}_N + \lambda_{\mathbf{u}} \log(\mathbf{u}) \tag{4.11}$$

By setting the gradients of (4.11) to 0, we get the optimal solution:

$$u_p^{(n+1)} = \exp\left((K(\mathbf{f}_p - \mathbf{m}) + \lambda \sum_{q=1}^{N} w_{p,q} u_q^{(n)}) / \lambda_{\mathbf{u}} \right). \\ \exp\left(-(\gamma + \lambda_{\mathbf{u}})\right) \tag{4.12}$$

Using this expression in the simplex constraint $\sum_p u_p^{(n+1)} = 1$ enables to recover the following expression of $\exp(\gamma + \lambda_{\mathbf{u}})$:

$$\sum_{j=1}^{N} \exp\left((K(\mathbf{f}_j - \mathbf{m}) + \lambda \sum_{q=1}^{N} w_{j,q} u_q^{(n)}) / \lambda_{\mathbf{u}} \right)$$

Plugging this expression back in (4.12), we get the final updates:

$$u_p^{(n+1)} = \frac{\exp\left((K(\mathbf{f}_p - \mathbf{m}) + \lambda \sum_{q=1}^{N} w_{p,q} u_q^{(n)})/\lambda_{\mathbf{u}}\right)}{\sum_{j=1}^{N} \exp\left((K(\mathbf{f}_j - \mathbf{m}) + \lambda \sum_{q=1}^{N} w_{j,q} u_q^{(n)})/\lambda_{\mathbf{u}}\right)} \tag{4.13}$$

4.6.2 Cauchy and convergent sequence proof

Let us consider iteration l, with the associated current *inlierness* scores $\mathbf{u}$. Let us prove that $\{\mathbf{m}^l\}_{l\in\mathbb{N}}$ is a Cauchy sequence. Recall the recursive relation:

$$\mathbf{m}^{l+1} = \frac{\sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}^l)\mathbf{f}_p}{\sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}^l)} \tag{4.14}$$

with $K(\mathbf{f}_p - \mathbf{m}^l) = \exp(-\|\frac{\mathbf{f}_p - \mathbf{m}^l}{h}\|^2)$, for some $h > 0$. We define:

$$k(x) = \exp(-x) \tag{4.15}$$

$$g^l = \sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}^l) \tag{4.16}$$

$$v^l = \sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}^l)\mathbf{f}_p \tag{4.17}$$

Step 1: First, let us prove that $\{g^l\}_{l\in\mathbb{N}}$ is a Cauchy sequence. Recall that in a metric space, a convergent sequence is necessarily a Cauchy sequence. Therefore, we only need to show that g^l is convergent (i.e bounded and strictly monotonic).
Notice that for $x > 0$, $0 \leq k(x) \leq 1$. Therefore:

$$g^l = \sum_{p=1}^{N} u_p k(\|\frac{\mathbf{f}_p - \mathbf{m}^l}{h}\|^2) \tag{4.18}$$

$$\leq \sum_{p=1}^{N} u_p \leq 1 \tag{4.19}$$

Therefore, g^l is bounded between 0 and 1. Now, let us study the consecutive differences $\Delta^l = g^{l+1} - g^l$:

$$\Delta^l = \sum_{p=1}^{N} u_p \left[k(\frac{\|\mathbf{f}_p - \mathbf{m}^{l+1}\|^2}{h^2}) - k(\frac{\|\mathbf{f}_p - \mathbf{m}^{l}\|^2}{h^2}) \right] \tag{4.20}$$

Because k is convex, one can say that $\forall a, b \in \mathbb{R}$:

$$k(a) - k(b) \geq k'(b)(a - b) \tag{4.21}$$

And because $k'(b) = -k(b)$ in our case, one ends up with:

$$k(a) - k(b) \geq k(b)(b - a) \tag{4.22}$$

Applied with $a = \frac{\|\mathbf{f}_p - \mathbf{m}^{l+1}\|^2}{h^2}$ and $b = \frac{\|\mathbf{f}_p - \mathbf{m}^{l}\|^2}{h^2}$, one can obtain:

$$\begin{aligned}
\Delta^l \geq & \sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}^l) \left[\frac{\|\mathbf{f}_p - \mathbf{m}^l\|^2}{h^2} - \frac{\|\mathbf{f}_p - \mathbf{m}^{l+1}\|^2}{h^2} \right] \\
= & \frac{1}{h^2} \sum_{p=1}^{N} u_p K(\mathbf{f}_p - \mathbf{m}^l) \Big[\|\mathbf{m}^l\|^2 - \|\mathbf{m}^{l+1}\|^2 \\
& -2 < \mathbf{m}^l, \mathbf{f}_p > +2 < \mathbf{m}^{l+1}, \mathbf{f}_p > \Big]
\end{aligned}$$

Now is time to recall recursive relation $\mathbf{m}^{l+1} = \frac{v^l}{g^l}$. By simply expanding, one can end up with:

$$\begin{aligned}
\Delta^l \geq & \frac{1}{h^2} \left[\|\mathbf{m}^l\|^2 g^l - \frac{\|v^l\|^2}{g^l} - 2 < \mathbf{m}^l, v^l > +2\frac{\|v^l\|^2}{g^l} \right] \\
= & \frac{1}{h^2} g^l \left[\|\mathbf{m}^l\|^2 - 2 < \mathbf{m}^l, \mathbf{m}^{l+1} > + \|\mathbf{m}^{l+1}\|^2 \right] \\
= & \frac{1}{h^2} g^l \|\mathbf{m}^l - \mathbf{m}^{l+1}\|^2
\end{aligned} \tag{4.23}$$

Therefore, $\Delta^l > 0$, which shows that $\{g^l\}_{l \in \mathbb{N}}$ is strictly increasing. This concludes the proof that g^l is a convergent sequence, and therefore a Cauchy one.

Step 2: Now, on top of concluding the proof that $\{g^l\}_{l\in\mathbb{N}}$ is a Cauchy sequence, Eq. (4.23) also offers an interesting relation between $\{\Delta^l\}_{l\in\mathbb{N}}$ and the sequence of interest $\{\mathbf{m}^l\}_{l\in\mathbb{N}}$, which we can use. Indeed, for any $l_0, m \in \mathbb{N}$, we can sum Eq. (4.23):

$$\sum_{l=l_0}^{l_0+m} \Delta^l \geq \frac{1}{h^2} \sum_{l=l_0}^{l_0+m} g^l \|\mathbf{m}^l - \mathbf{m}^{l+1}\|^2 \tag{4.24}$$

$$\geq \frac{g^{l_0}}{h^2} \sum_{l=l_0}^{l_0+m} \|\mathbf{m}^l - \mathbf{m}^{l+1}\|^2 \tag{4.25}$$

$$\geq \frac{g^{l_0}}{h^2} \|\mathbf{m}^{l_0+m} - \mathbf{m}^{l_0}\|^2 \tag{4.26}$$

Where Eq. (4.25) follows because $\{g^l\}_{l\in\mathbb{N}}$ is strictly increasing, and Eq. (4.26) follows from the triangle inequality. Now, the left-hand side of Eq. (4.24) can be reduced to $\sum_{l=l_0}^{l_0+m} \Delta^l = g^{l_0+m+1} - g^{l_0}$. But because we proved in Step 1 that $\{g^l\}_{l\in\mathbb{N}}$ was a Cauchy sequence, this difference is bounded by a constant. This concludes the proof that $\{\mathbf{m}^l\}_{l\in\mathbb{N}}$ is itself a Cauchy sequence in the Euclidean space.

Step 3: We just proved that $\{\mathbf{m}^l\}_{l\in\mathbb{N}}$ was a Cauchy sequence. Therefore $\{\mathbf{m}^l\}_{l\in\mathbb{N}}$ can only converge to a single value $\mathbf{m}^*$. We now use the continuity of function g to conclude that $\mathbf{m}^*$ has to be a solution of the initial equation (4.7):

$$\mathbf{m}^* = \lim_{l\to\infty} \mathbf{m}^{l+1} = \lim_{l\to\infty} g(\mathbf{m}^l) \tag{4.27}$$

$$= g(\lim_{l\to\infty} \mathbf{m}^l) = g(\mathbf{m}^*) \tag{4.28}$$

5

Transductive Learning for Vision-Language Models

Preliminary Note:

This chapter expands primarily on the conference proceedings [5–7], introducing **TransCLIP**, a novel framework for *transductive* inference. Unlike inductive methods that process test samples independently, it leverages the structure of unlabeled batches to refine predictions. TransCLIP achieves this by optimizing a new objective function that integrates a **Gaussian Mixture Model (GMM)** with a **text-driven Kullback-Leibler (KL) divergence penalty**, solved via an efficient Block Majorize-Minimize (BMM) procedure.

We recommend that readers first review the fundamental principles of CLIP in Chapter 2. Additionally, the concepts introduced in Section 3.2 of Chapter 3, specifically the overview of inductive few-shot adaptation (Section 3.2.2), are important prerequisites.

5.1 Introduction

In the previous chapters, we focused on *inference* performed on a per-instance basis, aligning with the inductive paradigm in few-shot learning (Chapter 3) and single-sample adaptation (Chapter 4). In this chapter, we transition to the *transductive* setting (defined in Section 2.2), which relies on a stronger assumption regarding data availability during inference. Rather than processing a single sample in isolation, we now leverage access to an entire batch of unlabeled test data. For the purposes of this chapter, we initially posit that this batch is well-distributed, containing multiple samples for each target class. It serves as a necessary baseline to isolate and validate the core benefits of transductive inference on several application cases. This assumption of a well-behaved batch will be relaxed in Chapter 6, which addresses more *realistic* deployment settings.

5.1.1 Context and Research Positioning

We saw in the previous chapters that the literature on adapting VLMs has grown substantially, in both the zero-shot and few-shot learning settings [25–27,40,46,47,49]. However, so far, these techniques predominantly align with *induction*, i.e., inference for each test sample is performed independently from the other samples within the target dataset. In contrast, *transduction* performs joint inference on all the test samples of a task, leveraging the statistics of the target unlabeled data [121–123]. In the context of standard vision-based classifiers, this has enabled transductive methods to outperform inductive-inference approaches as evidenced by benchmarks over large-scale datasets such as ImageNet [124].

Within the scope of deep learning, transduction has mainly been explored for few-shot learning to address the inherent challenges of training under limited supervision. This recent and quite abundant few-shot literature, e.g., [125–132], among others, has focused on adopting standard vision-based pre-training models (such as ImageNet pre-training). However, as we will show in our experiments, the direct application of existing transductive few-shot methods to VLMs yields poor performances, sometimes underperforming the inductive zero-shot predictions. This might explain why the transductive paradigm has been overlooked in zero-shot and few-shot learning for VLMs so far.

The low performance of current transductive few-shot methods in the context of VLMs could be explained by the fact that the underlying objective functions do not account for the text knowledge. In this new multimodal paradigm, additional supervision could be leveraged from the textual descriptions of the classes (prompts) [16], e.g., `a photo of a` [$\texttt{class}_k$], along with their corresponding normalized representation $\mathbf{t}_k$ derived from the language encoder. We utilize the interleaved representation of text prompts and images with their cosine similarity $\mathbf{f}^\top \mathbf{t}_k$, which yields text-based prediction $\hat{\mathbf{s}}_k$, thereby guiding our transductive optimization procedure with text-encoder knowledge. This is illustrated in Figure 5.1. Our method optimizes a novel objective function integrating a text-driven penalty. Optimization is carried out efficiently w.r.t the assignment variables associated with the unlabeled samples, which are then used as final predictions.

Adapting VLMs has recently attracted wide attention in the literature, predominantly focusing on inductive methods. As seen in Chapter 3, prompt tuning has become the favorite strategy for adapting VLMs in a variety of contexts, including unsupervised [46–48, 133, 134] and few-shot [1, 25,26,40–45,58,135,136] learning. Meanwhile, there have been a few efforts towards computationally more efficient adapters [27,50,59]. Our transduction formulation aligns with this initiative. By operating solely on the output embeddings (i.e., in a black-box setting), TransCLIP is computationally efficient and does not make assumptions on the underlying encoder architectures. Still, our method is orthogonal to these design choices and could be applied atop any of the above-mentioned inductive approaches.

5.1.2 Summary of Contributions

In this chapter, we introduce **TransCLIP**, which enhances the zero- and few-shot generalization of CLIP models, leveraging the structure of unlabeled data and textual priors through a computationally efficient, plug-and-play formulation.

(i) We introduce a transductive formulation that enhances the zero-shot and few-shot generalization capabilities of VLMs by leveraging the structure of unlabeled data. Our new objective function can be viewed as a regularized maximum-likelihood estimation, constrained by a Kullback-Leibler (KL) divergence penalty integrating the text-encoder knowledge

and guiding the transductive learning process. We further derive an iterative Block Majorize-Minimize (BMM) procedure for optimizing our objective, with guaranteed convergence and decoupled sample-assignment updates, yielding computationally efficient transduction.

(ii) Our method can be used as a plug-and-play module on top of current inductive zero-shot models and few-shot learning methods, consistently boosting their performance.

(iii) Our approach substantially outperforms recent transductive few-shot methods in the literature, notably due to the KL-based language supervision as a critical success factor.

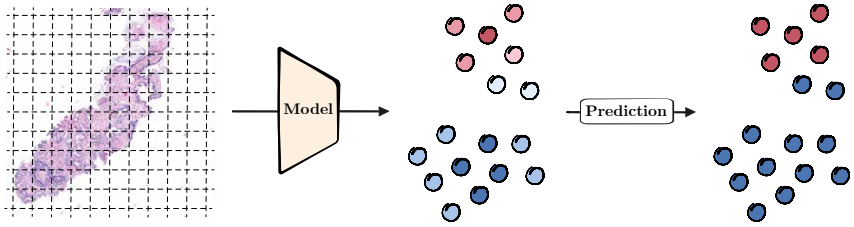
5.1.3 Chapter Outline

The remainder of this chapter is structured as follows. Section 5.2 reviews the paradigm of transductive learning, focusing on its application in traditional vision-only methods. Next, Section 5.3 introduces our novel procedure, TransCLIP. This section details the core objective function which integrates a text-driven Kullback-Leibler (KL) divergence penalty and derives the Block Majorize-Minimize (BMM) optimization procedure. We then present the experimental validations in Sections 5.4 and 5.5, deploying TransCLIP as a plug-and-play module atop inductive zero-shot and few-shot models, respectively. To demonstrate the framework's versatility, Sections 5.7 and 5.8 extend the evaluation to the specialized domains of satellite imagery and histopathology. Finally, Section 5.9 summarizes the findings of the chapter, while the Appendix (Section 5.10) provides the detailed mathematical derivations.

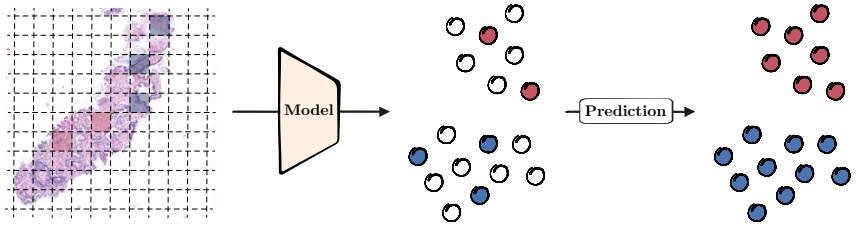
5.2 Evolution of Transductive Inference in the VLM Era

Transduction for vision-only classifiers

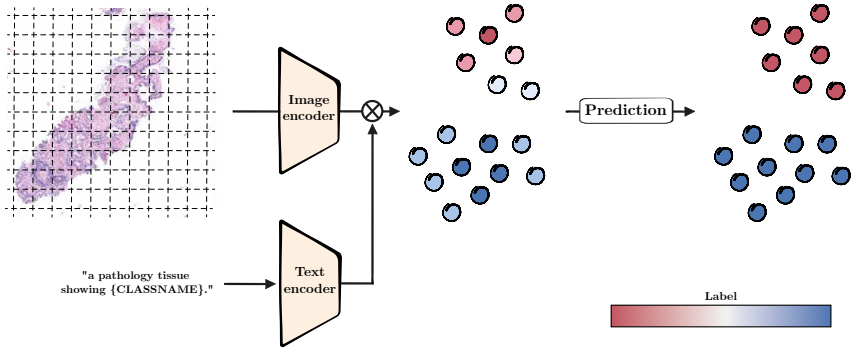
The use of unlabeled test data at inference time has received attention lately in rapidly emerging subjects, such as few-shot learning and unsupervised test-time adaptation. Examples include adjusting batch normalization layer statistics [137] and minimizing the entropy of predictions [138], which can be supplemented by pseudo-labeling strategies [139]. In the few-shot literature solely based on vision models, transduction leverages



(a) In the typical **inductive setting**, a model is trained and then used to perform inference on each patch separately. This approach can be effective when large annotated datasets for each task are available. This procedure often involves predicting the most probable class (argmax).



(b) In the traditional **transductive few-shot setting**, a pre-trained encoder (e.g., on ImageNet or a large-scale histology dataset) requires manual annotations for the new task to propagate information from labeled to unlabeled samples. This process often involves measuring affinities or distances between encoded samples.



(c) In our **zero-shot transductive setting**, VLMs leverage textual descriptions of each class to generate pseudo-labels without any manual annotation. These initial predictions can then be refined, for example, by leveraging the data structure.

Fig. 5.1 Illustration depicting histopathology classification in the inductive setting (a), the commonly used few-shot transductive setting (b), and the zero-shot transductive setting enabled by VLMs (c). This highly practical case will be studied later in this chapter (see Section 5.8).

both the few labeled samples and unlabeled test data, outperforming inductive methods [125, 128, 130–132]. One of the first works introducing transduction in vision-based few-shot learning proposes propagating the labels from the support (labeled) to the query (unlabeled) set with a meta-learned graph [140]. Building on this idea, another work proposes to iteratively augment the support set to improve label propagation [127]. LaplacianShot [130] also exploits the inherent structure of the data through a graph-Laplacian clustering, which discourages disparate class predictions for samples with close features, while matching each query set point to the nearest support prototype.

Alternative approaches propose directly learning the class prototypes. For instance, Transductive Fine-Tuning (TF) [126] uses the prediction entropy on the query samples as a regularization term, while TIM and its variants [125, 141] employ the mutual information between the query samples and their predictions. BD-CSPN [128] refines the class prototypes by reducing the feature biases between the support set and the most confident query samples.

An other group of methods performs clustering in the feature space, for instance, by solving an optimal transport problem like PT-MAP [131], by projecting features into sub-spaces to facilitate clustering [132], or by revisiting the standard K-means with an additional partition-complexity regularizer to control the number of predicted classes [129].

Transductive inference in VLMs

Despite the growing interest in unsupervised, zero-shot and few-shot learning for VLMs, the transductive-inference paradigm has not been explored so far in this new multi-modal context, except for the very recent work in [142], which was deployed for small-scale tasks ($\approx 10^2$ test samples). However, the method in [142] may not be computationally tractable for large-scale query sets, due to expensive inner loops for estimating the Dirichlet distribution's parameters. We provide a computationally efficient solution, which can scale up to large target datasets (such as ImageNet), while being easily amenable as a plug-and-play module on top of state-of-the-art inductive methods.

It is worth mentioning that test-time adaptation methods also employ the transduction paradigm, but their settings are very different from those

studied in this work. For instance, SwapPrompt [133] has been designed to make batch predictions on-the-fly, and has continual-learning mechanisms such as an exponential moving average prompt across batches. TPT [47] works on a single sample with many data augmentations to train one prompt per image. Both methods require access to model weights for training (i.e., do not operate in a black-box setting) and an expensive training procedure. We also note that prompt tuning does not scale well with the model size and is even impractical on very large models such as EVA-CLIP-8B [143].

5.3 TransCLIP: Transduction for Vision-Language Models

In this section, we describe our objective function for transductive inference and derive a Block Majorize-Minimize (BMM) algorithm for minimizing it. This algorithm features guaranteed convergence and decoupled sample-assignment updates. We adopt the zero-shot classification framework and notation established in Section 2.3.5; we encourage the reader to review that section to gain familiarity with the terminology and concepts.

Recall that for a test image $\mathbf{x}_i$ and a set of K classes, we obtain a visual embedding $\mathbf{f}_i$ and a set of text embeddings $\{\mathbf{t}_k\}_{k=1}^K$ derived from class-specific prompts (e.g., "`a photo of a` [$\texttt{class}_k$]"). Building on this, we compute the initial zero-shot pseudo-labels $\hat{\mathbf{s}}_i$ by applying the softmax function to the cosine similarities, scaled by the temperature τ:

$$\hat{\mathbf{s}}_i = (\hat{s}_{i,k})_{1\leq k\leq K} \in \Delta_K; \quad \hat{s}_{i,k} = \frac{\exp(\mathbf{f}_i^\top \mathbf{t}_k/\tau)}{\sum_j \exp(\mathbf{f}_i^\top \mathbf{t}_j/\tau)} \tag{5.1}$$

where Δ_K denotes the probability simplex. Let $\mathcal{D} = \mathcal{S} \cup \mathcal{Q}$ denote the set of all sample indices in the target dataset. Accordingly, $\mathcal{Q}$ represents the set of indices for the unlabeled *query* samples, and $\mathcal{S}$ denotes the set of indices for the labeled *support* samples.

Note that, in the zero-shot setting, $\mathcal{S} = \emptyset$. We define a Gaussian Mixture Model-clustering (GMM) term in our objective function by modeling the likelihood of these target data as a balanced mixture of multivariate Gaussian distributions, each representing a class k and parameterized by

mean vector $\boldsymbol{\mu}_k$ and a diagonal covariance matrix $\boldsymbol{\Sigma}$:

$$p_{i,k} = \Pr(\mathbf{f}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \propto \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{f}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{f}_i - \boldsymbol{\mu}_k)\right)$$

Notice that, unlike standard GMMs, we deploy a common diagonal covariance matrix $\boldsymbol{\Sigma}$ across all classes. Interestingly, in our experiments, we found this simplifying choice improves the performance while reducing the computational load as there are substantially fewer parameters to learn. This is particularly the case when dealing with large numbers of classes as in large-scale target datasets such as ImageNet. This choice is further motivated by a strong few-shot baseline based on Gaussian discriminant analysis clustering [144] demonstrates VLMs' adaptation abilities with a Gaussian hypothesis on the embedding space.

5.3.1 Proposed Objective Function

Our objective function depends on two types of variables: (i) Sample-to-class assignment variables within probability simplex: $\mathbf{z}_i = (z_{i,k})_{1 \le k \le K} \in \Delta_K$, $i \in \mathcal{Q}$; and (ii) GMM parameters $\boldsymbol{\mu} = (\boldsymbol{\mu}_k)_{1 \le k \le K}$ and $\boldsymbol{\Sigma}$. We propose to minimize the following objective, which integrates a GMM-clustering term, a Laplacian regularizer and a Kullback-Leibler (KL) divergence penalty encoding the text-encoder knowledge and guiding the transductive learning process:

$$\mathcal{L}_{\text{ZERO-SHOT}}(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \underbrace{-\sum_{i \in \mathcal{Q}} \mathbf{z}_i^\top \log(\mathbf{p}_i)}_{\text{GMM clustering}} \underbrace{- \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{D}} w_{ij} \mathbf{z}_i^\top \mathbf{z}_j}_{\text{Laplacian reg.}} + \underbrace{\sum_{i \in \mathcal{Q}} \text{KL}_\lambda(\mathbf{z}_i || \hat{\mathbf{s}}_i)}_{\text{Text knowledge}} \tag{5.2}$$

where $\mathbf{p}_i = (p_{i,k})_{1 \le k \le K}$ vectorizes the GMM component densities, w_{ij} denotes some measure of affinity between visual embeddings $\mathbf{f}_i$ and $\mathbf{f}_j$, and the sample-wise parameterized[1] KL terms are given by:

$$\text{KL}_\lambda(\mathbf{z}_i || \hat{\mathbf{s}}_i) = \mathbf{z}_i^\top \log \mathbf{z}_i - \lambda \mathbf{z}_i^\top \log \hat{\mathbf{s}}_i, \quad i \in \mathcal{Q}; \quad \lambda > 0 \tag{5.3}$$

In the following, we describe the effect of each term in our objective function in (5.2):

[1] Notice that, for $\lambda = 1$, the expression in (5.3) corresponds to the KL divergence.

- **GMM-based clustering**: This unsupervised-learning term is akin to the GMM-based maximum-likelihood estimation objective in the standard EM algorithm [145]. By taking the negative logarithm, its minimization corresponds to maximizing the likelihood of the data. It can also be viewed as a probabilistic generalization of the K-means clustering objective [146]. Indeed, assuming $\mathbf{\Sigma}$ is the identity matrix reduces the first term in (5.2) to the K-means objective.

- **Laplacian regularization**: The second term in (5.2) is the Laplacian regularizer, widely used in the context of graph/spectral clustering [147] and semi-supervised learning [148]. This term encourages nearby samples in the visual-embedding space (i.e., pairs of samples with high affinity $w_{i,j}$) to have similar $\mathbf{z}$ assignments. In our case, we propose to build a positive semi-definite (PSD) affinity matrix based on the cosine similarities as $w_{ij} = \mathbf{f}_i^\top \mathbf{f}_j$ (Gram matrix). As we see below, this PSD condition is important to obtain a convergent Majorize-Minimize optimizer with decoupled (parallel) sample-wise updates for the $\mathbf{z}$-assignments, yielding a highly efficient transduction for large-scale target datasets (such as ImageNet).

- **Text-guided KL divergence:** This term is dedicated to vision-language models and, as we will see in our experiments (ablation studies in Tables 5.4 and 5.5), has a substantial effect on performance. It encourages the prediction not to deviate significantly from the zero-shot predictions, thereby providing text supervision to the other two unsupervised-learning terms. Furthermore, being convex over $\mathbf{z}_i$, $i \in \mathcal{Q}$, this term facilitates the optimization of the objective w.r.t the assignment variables.

5.3.2 Extension to the Few-Shot Setting

Our zero-shot formulation naturally extends to the few-shot setting. We integrate supervision from the labeled-support samples, in the form of a cross-entropy, which corresponds to minimizing the following overall loss:

$$\mathcal{L}_{\text{Few-Shot}}(\mathbf{z}, \boldsymbol{\mu}, \mathbf{\Sigma}) = -\frac{\gamma}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \mathbf{z}_i^\top \log(\mathbf{p}_i) + \frac{1}{|\mathcal{Q}|} \mathcal{L}_{\text{Zero-Shot}}(\mathbf{z}, \boldsymbol{\mu}, \mathbf{\Sigma}) \tag{5.4}$$

Note that, in the first term, the $\mathbf{z}_i$ are fixed, with $\mathbf{z}_i = \mathbf{y}_i$, $i \in \mathcal{S}$ and $\mathbf{y}_i$ the one-hot ground-truth label associated with the corresponding shot.

5.3.3 Block Majorize-Minimize (BMM) Optimization

As our objective depends on three types of variables ($\mathbf{z}$, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$), we proceed with a BMM procedure, alternating three sub-step optimizers. Each sub-step optimizes over one block of variables while the other two are fixed, ensuring the overall objective does not increase. Importantly, the obtained $\mathbf{z}$-updates (Eq. (5.5)) are decoupled, yielding computationally efficient transduction for large-scale datasets. Also, our overall procedure is guaranteed to converge (Theorem 5.1).

Majorize-Minimize (MM) with respect to the z-block

When $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are fixed, both the GMM- and KL-based terms are convex w.r.t $\mathbf{z}_i$. However, the Laplacian term is concave[2] (for PSD matrix $\boldsymbol{W}$). Therefore, we proceed with inner iterations, each minimizing a linear and tight upper bound, the so-called majorizing function in the MM-optimization literature [149–151], which guarantees the overall objective does not increase. To obtain the tight linear bound, let us write the Laplacian term conveniently in the following matrix form: $\mathbf{z}^\top \boldsymbol{\Psi} \mathbf{z}$, with $\boldsymbol{\Psi} = -\boldsymbol{W} \otimes \boldsymbol{I}$, where $\otimes$ denotes the Kronecker product and $\boldsymbol{I}$ is the $|\mathcal{D}| \times |\mathcal{D}|$ identity matrix. Note that $\boldsymbol{\Psi}$ is negative semi-definite for a positive semi-definite $\boldsymbol{W}$. Therefore, $\mathbf{z}^\top \boldsymbol{\Psi} \mathbf{z}$ is a concave function with respect to $\mathbf{z}$, and its first-order approximation at current solution $\mathbf{z}^l$ (l being the iteration index) gives the following tight[3] upper bound on the Laplacian term:

$$\mathbf{z}^\top \boldsymbol{\Psi} \mathbf{z} \leq (\mathbf{z}^l)^\top \boldsymbol{\Psi} \mathbf{z}^l + (\boldsymbol{\Psi} \mathbf{z}^l)^\top (\mathbf{z} - \mathbf{z}^l)$$

Replacing the quadratic Laplacian term by this linear bound yields a majorizing function on our overall objective. Importantly, this majorizing function is a sum of decoupled objectives, each corresponding to one assignment variable $\mathbf{z}_i$, yielding a highly efficient optimizer for large-scale target datasets. Indeed, using simplex constraints $\mathbf{z}_i \in \Delta_K, i \in \mathcal{Q}$, and solving the Karush-Kuhn-Tucker (KKT) conditions independently for each $\mathbf{z}_i$, we obtain the following decoupled update rules for the $\mathbf{z}$-block:

$$\mathbf{z}_i^{(l+1)} = \frac{\hat{\mathbf{s}}_i^\lambda \odot \exp(\log(\mathbf{p}_i) + \sum_{j \in \mathcal{D}} w_{ij} \mathbf{z}_j^{(l)})}{(\hat{\mathbf{s}}_i^\lambda \odot \exp(\log(\mathbf{p}_i) + \sum_{j \in \mathcal{D}} w_{ij} \mathbf{z}_j^{(l)}))^\top \mathbf{1}_K} \tag{5.5}$$

[2] This makes the overall sub-problem non-convex and there is no closed-form solution.

[3] "Tight" means that the upper bound is equal to the original objective at the current solution $\mathbf{z}^l$.

Closed-form updates of μ and Σ

When both $\mathbf{z}$ and $\mathbf{\Sigma}$ are fixed, our objective in (5.4) is convex. It can be minimized by setting its gradient w.r.t each $\boldsymbol{\mu}_k$ to zero, which yields the following closed-form updates:

$$\boldsymbol{\mu}_k = \frac{\frac{\gamma}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} z_{i,k}\mathbf{f}_i + \frac{1}{|\mathcal{Q}|}\sum_{i\in\mathcal{Q}} z_{i,k}\mathbf{f}_i}{\frac{\gamma}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} z_{i,k} + \frac{1}{|\mathcal{Q}|}\sum_{i\in\mathcal{Q}} z_{i,k}} \tag{5.6}$$

Similarly, when both $\mathbf{z}$ and $\boldsymbol{\mu}$ are fixed, the following closed-form updates minimize the overall objective w.r.t $\mathbf{\Sigma}$:

$$\text{diag}(\mathbf{\Sigma}) = \frac{\frac{\gamma}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\sum_k z_{i,k}(\mathbf{f}_i - \boldsymbol{\mu}_k)^2 + \frac{1}{|\mathcal{Q}|}\sum_{i\in\mathcal{Q}}\sum_k z_{i,k}(\mathbf{f}_i - \boldsymbol{\mu}_k)^2}{\gamma + 1} \tag{5.7}$$

Note that, after convergence, we use the sample-to-class assignment variables $\mathbf{z}_i$ as predictions for each sample i of the query set $\mathcal{Q}$ using the argmax operation for conventional classification.

Implementation details

The main component of our transductive formulation is the text-guided KL divergence penalty. We fix $\lambda = 1$ for all our zero-shot experiments (see ablation study in Table 5.5), and $\lambda = 0.5$ in all the few-shot experiments to reduce the impact of the text-driven regularization. Another component of our optimization problem is the Laplacian regularization, which enforces consistent predictions for close instances. We truncate the affinity matrix to the 3 nearest-neighbors, making it sparse. $\boldsymbol{\mu}$ is initialized as the average embedding of the top-8 most confident samples of each class for the zero-shot setting. For the few-shot setting, we use the class-wise average over the shot embeddings.

5.3.4 Convergence

Our optimizer can be viewed as an instance of the general Block Majorize-Minimize paradigm for optimization [152], which optimizes a majorizing function for each block of variables. The convergence of general BMM procedures is well studied in the optimization community [152]. Indeed, under certain conditions (such as the strong convexity of the block-wise majorizing functions), we can establish convergence of our procedure using the following result (more details in Appendix 5.10.1):

Theorem 5.1 (Convergence of BMM [152]). *Assume that, for each block, the majorizing function is quasi-convex, and its first-order behavior is the same as the original objective locally. Furthermore, assume that the sub-problem solved for each block has a unique solution. Then, every limit point of the iterates generated by BMM is a coordinate-wise minimum of the overall objective.*

5.4 Experiment 1: Zero-Shot TransCLIP on Natural Images

5.4.1 Experimental Setup

Datasets
We follow conventional benchmarks composed of 11 datasets for fine-grained classification, namely ImageNet (I.) [17], SUN397 (SUN) [69], Aircraft (Air.) [70], EuroSAT (SAT) [68], StanfordCars (Cars), Food101 (Food) [71], OxfordPets (Pets) [72], Flowers102 (Flow.) [73], Caltech101 (Calt.) [74], DTD (DTD) [75], UCF101 (UCF) [76]. We also report results on four ImageNet variants: Adversarial (Adv.) [117], V2 (V2) [118], Rendition (Rend.) [111] and Sketch (Sketch) [119].

Models
We primarily report results for the CLIP ViT-B/16 encoder but complementary results on other backbones can be found in the TransCLIP paper [5].

Evaluation protocol
We aim to show the breadth of potential applications of transduction in the context of VLMs. Notably, employing supervised fine-tuning, followed by transduction with TransCLIP on the unlabeled test samples, emerges as a powerful and efficient solution. This is particularly convenient when the labeled samples (the support set) and/or computational power are not accessible at inference (i.e., test) time. To this end, we first study the applicability of our zero-shot formulation TransCLIP (Eq. (5.2)) across three settings: (i) on top of inductive *zero-shot* learning and popular *few-shot* learning methods; (ii) on top of 16-shot ImageNet pretraining for *cross-dataset* transferability, and (iii) on top of 16-shot ImageNet pretraining for *domain generalization* on the four ImageNet variants. Each reported performance metric represents the top-1 accuracy on the whole test set, averaged over three random seeds.

Comparative methods
We slightly modify UPL to apply it to the test set in a transductive manner (transductive UPL is denoted UPL*). Indeed, UPL relies on the generation of $N = 16$ hard pseudo-labels per class from a training set, after which a cross-entropy loss function on soft tokens is minimized. Instead, UPL* generates the pseudo-labels directly from the test set. For fairness, we reevaluated the number of pseudo-labels to select and still found that 16 per class yields the best results on average. For conciseness, we only report UPL* performance in Table 5.6. We refer to Chapter 3 for details on few-shot methods used in this experimental section.

5.4.2 Results

Boosting inductive baselines
Tables 5.1 and 5.2 demonstrate the consistent advantages of our transductive approach. TransCLIP enhances the zero-shot top-1 accuracy by over 5% and improves popular few-shot methods by 4% (in the 1-shot setting) on average, all without requiring additional labeled data. We observe that while transductive gains are significant in low-shot regimes, they tend to decrease as the number of shots increases (e.g., 16-shot), presumably because the few-shot training already captures much of the data structure.

Domain generalization
Table 5.3 highlights that TransCLIP effectively enhances robustness even when applied atop fine-tuned methods such as CoOp or TaskRes. This confirms that leveraging the structure of unlabeled test samples provides complementary information to supervised fine-tuning.

Robustness and efficiency
With *its hyperparameters unchanged*, TransCLIP exhibits strong generalization across diverse architectures, from convolutional networks to transformers. More detailed results for five different backbone architectures and comparisons with unsupervised non-transductive methods can be found in the Appendix of the TransCLIP paper [5]. Furthermore, as shown in Table 5.6, TransCLIP significantly outperforms the transductive prompt learning baseline (UPL*) while being two to three orders of magnitude faster, confirming the efficiency of operating directly in the embedding space.

Table 5.1 TransCLIP atop inductive vision-language zero-shot and popular few-shot methods. We indicate improvement or degradation compared to zero-shot.

	Method	I.	SUN	Air.	SAT	Cars	Food	Pets	Flow.	Calt.	DTD	UCF	Average
0-shot	**CLIP-ViT-B/16**	66.6	62.5	24.7	48.3	65.6	85.9	89.1	70.7	93.2	43.5	67.5	65.3
	+ TransCLIP	+3.7	+6.3	+2.2	+16.8	+3.8	+1.2	+3.5	+5.9	-0.5	+6.0	+6.9	+5.1
1-shot	CoOp	65.7	66.9	20.7	56.4	67.6	84.3	90.2	78.2	92.5	50.1	71.2	67.6
	+ TransCLIP	+3.6	+4.6	+3.1	+8.9	+4.3	+2.0	+1.8	+11.5	+1.3	+5.4	+6.5	+4.8
	TIP-Adapter-F	69.5	67.2	28.8	67.8	67.1	85.8	90.6	83.7	94.0	51.6	73.4	70.9
	+ TransCLIP	+2.5	+4.6	+1.9	+9.1	+3.9	+1.1	+2.4	+9.1	-0.5	+6.1	+6.7	+4.3
	PLOT	66.9	67.0	28.9	72.8	68.5	84.9	91.9	81.8	94.0	52.8	74.7	71.3
	+ TransCLIP	+8.9	+3.3	-0.8	+6.0	+1.6	+0.4	-0.8	+11.4	-0.0	+3.9	+6.7	+3.7
	TaskRes	69.6	68.1	31.2	65.6	69.1	84.5	90.2	81.6	93.6	53.4	71.8	70.8
	+ TransCLIP	+2.5	+4.4	+0.2	+8.1	+2.4	+2.0	+1.5	+9.1	+0.4	+6.0	+4.6	+3.7
	ProGrad	67.0	67.0	28.7	57.0	68.2	84.9	91.4	80.8	93.5	52.8	73.3	69.5
	+ TransCLIP	+3.1	+4.6	+1.8	+13.9	+4.1	+1.6	+1.4	+10.7	+0.7	+5.1	+6.1	+4.8
4-shot	CoOp	68.8	69.7	30.8	69.6	74.4	84.5	92.5	92.2	94.5	59.4	77.5	74.0
	+ TransCLIP	+2.6	+3.5	+2.3	+7.5	+3.2	+1.9	+1.1	+3.1	+0.6	+3.6	+4.3	+3.1
	TIP-Adapter-F	70.7	70.8	35.7	76.8	74.1	86.5	91.9	92.1	94.8	59.8	78.1	75.6
	+ TransCLIP	+1.9	+3.5	+0.5	+2.9	+1.8	+0.9	+1.3	+3.3	+0.3	+4.2	+5.2	+2.3
	PLOT	70.0	71.8	34.8	84.7	76.6	83.5	92.8	93.2	94.9	61.0	79.7	76.6
	+ TransCLIP	+7.2	+1.7	-0.9	-2.9	-0.8	+2.2	-0.3	+2.6	-0.1	+2.6	+3.6	+1.4
	TaskRes	71.0	72.8	33.3	73.8	76.1	86.1	91.9	85.0	94.9	59.7	75.5	74.6
	+ TransCLIP	+2.0	+2.5	+1.1	+4.4	+1.1	+1.2	+1.1	+7.4	+0.2	+4.6	+3.7	+2.7
	ProGrad	70.2	71.7	34.0	69.5	75.0	85.4	92.0	91.1	94.4	59.8	77.9	74.6
	+ TransCLIP	+2.1	+3.3	+1.6	+5.3	+2.9	+1.5	+1.7	+4.2	+0.8	+5.1	+5.4	+3.1
16-shot	CoOp	71.9	74.9	43.3	85.0	82.8	84.2	91.9	96.8	95.8	69.7	83.1	79.9
	+ TransCLIP	+1.4	+1.8	-0.4	+1.0	+0.2	+2.1	+1.2	+0.8	+0.1	+1.7	+2.3	+1.1
	TIP-Adapter-F	73.3	76.0	44.6	85.9	82.3	86.8	92.6	96.2	95.7	70.8	83.9	80.7
	+ TransCLIP	+0.9	+0.8	+0.3	-0.7	+0.4	+0.6	+0.9	+0.7	-0.1	-1.5	+1.7	+0.4
	PLOT	72.5	76.0	46.8	92.1	84.6	85.6	92.5	97.1	96.0	71.1	84.8	81.7
	+ TransCLIP	+5.3	-1.0	-4.9	-7.5	-4.9	+0.2	-0.4	+0.1	-1.0	-2.4	+0.9	-1.4
	TaskRes	73.0	76.0	44.8	80.7	83.5	86.9	92.5	97.3	95.9	70.9	83.4	80.5
	+ TransCLIP	+1.0	+0.8	-1.2	-0.3	-0.7	+0.6	+0.4	+0.3	+0.1	-0.7	+2.8	+0.3
	ProGrad	72.1	75.1	42.8	83.6	82.9	85.8	92.9	96.6	95.9	68.9	82.6	79.9
	+ TransCLIP	+1.4	+1.7	-0.0	+0.2	+0.2	+1.3	+0.8	+0.8	+0.1	+2.5	+3.4	+1.1

Table 5.2 Cross-dataset transferability evaluation. Few-shot learning methods are trained on 16-shot ImageNet and evaluate on the ten other fine-grained datasets. Average excludes ImageNet. We indicate improvement or degradation compared to zero-shot.

		Source	Target										
	Method	I.	SUN	Air.	SAT	Cars	Food	Pets	Flow.	Calt.	DTD	UCF	Average
Cross-Dataset	CoOp	71.9	62.0	15.7	44.6	62.1	84.3	88.3	67.1	92.7	39.5	64.1	62.0
	+ TransCLIP	+1.4	+5.4	+1.4	+9.9	+4.8	+2.0	+1.1	+7.2	+0.7	+2.6	+5.7	+4.1
	CoCoOp	71.1	67.0	22.7	44.6	64.9	86.2	90.7	71.6	93.9	45.2	68.8	65.6
	+ TransCLIP	+5.7	+2.7	-0.1	+14.6	+2.1	-0.8	-0.9	+7.4	+0.3	+5.4	+5.7	+3.6
	MaPLE	70.5	67.3	24.4	45.8	65.7	86.4	90.4	72.0	93.7	46.3	68.7	66.1
	+ TransCLIP	+6.1	+2.5	+0.2	+13.7	+1.2	-1.0	-0.7	+6.0	+0.6	+3.1	+5.6	+3.1
	ProGrad	72.1	63.9	21.6	38.9	64.0	85.9	90.2	67.8	92.9	43.2	65.9	63.4
	+ TransCLIP	+1.4	+4.7	+1.1	+16.4	+3.8	+1.2	+1.1	+6.1	+1.1	+3.4	+7.6	+4.6
	PromptSRC	71.4	67.3	24.1	45.0	65.6	86.5	90.1	70.5	93.8	46.2	68.9	65.8
	+ TransCLIP	+5.5	+2.6	+0.8	+14.4	+2.0	-1.2	-0.7	+6.2	+0.4	+5.0	+7.0	+3.7

Table 5.3 Domain generalization evaluation with improved manual prompting strategy (custom templates are given in Table 8.1b), 16-shot prompt-tuning and 16-shot adapter. We indicate improvement or degradation compared to zero-shot.

		Source	Target					
	Method	I.	-Adv.	-V2	-Rend.	-Sketch	Average	Average OOD
0-shot	**CLIP-ViT-B/16** *w/* `a photo of a`	66.6	47.9	60.6	73.8	46.0	59.0	57.1
	+ TransCLIP	**+3.7**	**+1.7**	**+1.7**	**+1.3**	**+3.7**	**+2.4**	**+2.1**
	CLIP-ViT-B/16 *w/ custom templates*	68.8	50.6	62.3	77.8	48.4	61.6	59.8
	+ TransCLIP	**+2.7**	**+1.4**	**+1.1**	**+0.2**	**+2.7**	**+1.6**	**+1.3**
Domain G.	**CLIP-ViT-B/16** *w/ prompt tuning (CoOp)*	71.9	49.4	64.1	75.1	47.2	61.5	59.0
	+ TransCLIP	**+1.4**	**+1.4**	**+0.5**	**+0.7**	**+3.1**	**+1.5**	**+1.4**
	CLIP-ViT-B/16 *w/ adapter (TaskRes)*	73.0	50.3	65.6	77.8	49.2	63.2	60.7
	+ TransCLIP	**+1.1**	**+1.6**	**-0.2**	**+0.6**	**+2.4**	**+1.1**	**+1.1**

5.5 Experiment 2: Transductive Few-Shot TransCLIP

5.5.1 Experimental Setup

Datasets and models

See Experiment 1 (Section 5.4).

Evaluation protocol

We compare our few-shot extension TransCLIP-FS (Eq. (5.4)) to transductive few-shot learning methods. As for the zero-shot version, we also operate in a black-box setting (i.e., using only the output embeddings, without training the model parameters). We vary the number of labeled support examples per class, denoted as $N_{\text{shots}} \in \{1, 2, 4, 8, 16\}$. For each N_{shots} configuration, we randomly sample the support set $\mathcal{S}$ from the training data. The models have then access to the samples indexed by $\mathcal{S}$ and all the samples from the test set indexed by $\mathcal{Q}$. Performance is then evaluated on the whole test set. To account for the variability inherent in few-shot sampling, we report the top-1 classification accuracy averaged over three random seeds.

Comparative methods

We report several state-of-the-art transductive few-shot learning methods. (i) Transductive Fine-Tuning [126]. We search for the optimal temperature value by validation in $\{0.25, 0.5, 1, 2, 4\}$ and the number of iterations in $\{10, 15, 20, 25, 30, 35, 40\}$. (ii) BD-CSPN [128]: regarding the hyperparameters, this method generates Z pseudo-labels per class from the query

set to augment the support set and to build the K prototype vectors. They also introduce a temperature scaling parameter ε for the computation of the prototype vectors. The authors set Z to 8 and the temperature scaling ε to 10. We search for the value of Z in $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and ε in $\{2.5, 5, 10, 20, 40\}$ by validation. (iii) LaplacianShot [130]: the authors balance the Laplacian regularization term with a factor λ and used k-nearest neighbors consistency. We follow the proposed ranges to find the hyperparameter values by validation, with λ in $\{0.1, 0.3, 0.5, 0.7, 0.8, 1, 1.2, 1.5\}$ and the number of neighbors to consider k in $\{3, 5, 10\}$. (iv) PT-MAP [131]: the authors show a small performance sensitivity to the learning rate α used to update the class prototypes through iterative adaptation. Following their discussion, we search α in $\{0.2, 0.4\}$. (v) TIM [125]: the authors set the weighting factors of the cross-entropy, the marginal entropy, and the conditional entropy terms to 0.1, 1 and 0.1, respectively. They also introduced a temperature parameter τ in their classifier and set it to 15. We search for the values of the cross-entropy and the conditional entropy factors in $\{0.05, 0.1, 0.4, 0.7, 1\}$ and the temperature in $\{5, 10, 15, 30, 60\}$ by validation. (vi) CoOp [25] +UPL [46]: We implement a natural extension of CoOp to include the pseudo-labels proposed by UPL. As in UPL, $N = 16$ hard pseudo-labels per class are generated according to the prediction's confidence. Pseudo-labels from the query set $\mathcal{P} \subseteq \mathcal{Q}$ and labeled shots from $\mathcal{S}$ are unified into a single learning set $\mathcal{S} \cup \mathcal{P}$. To separate the contribution of the pseudo-labels from the labeled shots, we split the cross-entropy loss function into two terms:

$$\mathcal{L}_{\mathcal{S}\cup\mathcal{P}}(\overline{\mathrm{V}}|\{\mathbf{x}_i\}_{i=1}^{|\mathcal{S}\cup\mathcal{P}|}) = \beta\frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}}\mathcal{L}_{\text{CoOp}}(\overline{\mathrm{V}}|\mathbf{x}_j) + (1-\beta)\frac{1}{|\mathcal{P}|}\sum_{j\in\mathcal{P}}\mathcal{L}_{\text{UPL}}(\overline{\mathrm{V}}|\mathbf{x}_j)\,, \quad \beta \in [0,1] \tag{5.8}$$

Where $\overline{\mathrm{V}}$ denotes the vector of learnable context token embeddings. Despite increased computational needs, we search for the value of β in {0.1, 0.3,0.5,0.7,0.9} by validation. The number of epochs, the learning rate and its schedule, the optimizer and the context tokens initialization follow exactly the CoOp implementation.

Table 5.4 Transductive few-shot learning evaluation. *w/o text* denotes $\lambda = 0$ in Eq. (5.3).

	Method	I.	SUN	Air.	SAT	Cars	Food	Pets	Flow.	Calt.	DTD	UCF	Average
	CLIP-ViT-B/16	66.6	62.5	24.7	48.3	65.6	85.9	89.1	70.7	93.2	43.5	67.5	65.3
1-shot	TF	29.7	38.1	19.2	46.0	32.5	43.5	38.2	67.8	75.5	31.6	48.8	42.8
	BD-CSPN	35.4	45.7	22.0	45.7	42.0	54.2	52.9	82.9	83.5	34.7	58.0	50.6
	LaplacianShot	34.9	44.5	22.1	52.1	41.1	53.0	52.2	83.1	83.4	35.8	57.3	50.9
	PT-MAP	40.1	52.6	23.8	59.7	48.4	64.4	61.8	69.4	54.1	41.8	63.5	52.7
	TIM	37.5	48.3	22.8	48.2	44.8	65.7	53.9	**86.4**	75.1	35.8	62.7	52.8
	TransCLIP-FS *w/o text*	30.2	43.4	23.7	56.6	41.0	50.9	54.3	83.5	77.7	36.9	54.5	50.2
	TransCLIP-FS	**69.8**	**70.6**	**29.9**	**72.5**	**70.9**	**87.9**	**93.8**	84.8	**93.1**	**53.3**	**78.4**	**73.2**
4-shot	TF	51.1	61.0	30.3	64.9	56.8	71.0	65.9	90.9	91.5	53.7	67.9	64.1
	BD-CSPN	53.8	62.5	30.5	64.8	58.5	75.3	72.0	92.5	92.0	52.1	70.9	65.9
	LaplacianShot	53.5	62.5	29.6	74.3	58.5	75.7	73.4	92.8	92.0	52.7	71.7	67.0
	PT-MAP	57.6	68.1	31.2	74.9	63.1	81.1	79.5	76.2	60.2	58.4	73.9	65.8
	TIM	57.4	67.0	32.8	79.3	65.8	83.5	82.3	93.4	88.5	58.1	76.5	71.3
	TransCLIP-FS *w/o text*	53.9	63.8	**34.2**	**79.4**	63.5	76.7	76.7	93.3	92.8	57.0	74.8	69.6
	TransCLIP-FS	**70.3**	**71.9**	34.0	**79.4**	**74.0**	**86.4**	**91.6**	**93.6**	**94.0**	**61.1**	**79.1**	**75.9**
16-shot	TF	61.8	70.1	38.3	74.3	71.2	80.7	79.5	**95.4**	93.6	62.9	76.0	73.1
	BD-CSPN	61.7	69.4	37.7	73.4	70.7	80.2	81.2	94.8	93.3	61.3	76.0	72.7
	LaplacianShot	60.9	68.3	36.1	78.1	69.2	81.2	81.7	94.8	93.1	58.6	76.3	72.6
	PT-MAP	64.0	72.0	37.4	75.6	72.0	82.7	86.1	78.5	63.7	63.7	76.3	70.2
	TIM	67.8	73.6	40.6	**83.6**	79.5	84.9	88.7	**95.4**	92.4	**67.5**	**82.1**	77.8
	TransCLIP-FS *w/o text*	65.9	72.6	**41.9**	81.1	77.0	83.2	86.1	95.2	**94.6**	65.3	80.0	76.6
	TransCLIP-FS	**71.8**	**74.7**	38.6	83.0	**79.8**	**86.9**	**92.4**	94.4	94.0	65.1	**82.1**	**78.4**

5.5.2 Results

Transductive few-shot learning with VLMs

We compare TransCLIP-FS, TransCLIP-FS without text regularization (i.e., $\lambda = 0$) and state-of-the-art transductive few-shot methods. It is important to note that these few-shot methods were primarily developed for vision-centric tasks. Hence, they rely on visual information, omitting the textual elements. This allows us to study the impact of our text-based regularization term. Table 5.4 shows that incorporating language in the transductive paradigm boosts the performance over vision-only methods. Especially for the 1- to 4-shot settings, our language-driven KL penalty enhances the performance by a large margin on many tasks (e.g., ImageNet, SUN397, StanfordCars, DTD). As the number of shots increases, the text-driven penalty becomes less useful, especially for the datasets capitalizing on the visual shots rather than the text-encoder knowledge (e.g., EuroSAT and Flowers). This points to promising future directions involving more flexible text regularization (e.g., an adaptable λ taking into account the number of shots and the quality of the text embeddings). Detailed results for five different encoder architectures can be found in the Appendix of the TransCLIP paper [5], consistently showing similar conclusions.

Table 5.5 Analysis on the components and sensitivity to hyperparameters of TransCLIP (zero-shot).

(a) Components of the procedure.

Update μ	Update Σ	Lapl. w	ImageNet	SUN397	Aircraft	EuroSAT
✗	✓	✓	69.7	67.5	25.5	63.9
✓	✗	✓	68.7	66.0	25.1	51.6
✓	✓	✗	69.9	68.8	**27.0**	64.5
✗	✗	✓	68.6	65.9	25.2	61.8
✓	✓	✓	**70.3**	**68.9**	26.9	**65.1**

(b) Text regularization hyperparameter λ.

λ	ImageNet	SUN397	Aircraft	EuroSAT
0.1	56.3	58.6	26.0	65.5
0.5	69.8	**69.3**	26.6	**65.6**
1	**70.3**	68.9	**26.9**	65.1
2	69.5	67.6	26.2	64.1
5	68.2	65.2	25.2	51.2

(c) Number of nearest-neighbors.

# neighbors	ImageNet	SUN397	Aircraft	EuroSAT
3	70.3	68.9	26.9	65.1
5	70.3	68.9	26.8	65.1
10	70.2	68.8	26.9	65.2

(d) Impact of an isotropic $\boldsymbol{\Sigma}$.

	ImageNet	SUN397	Aircraft	EuroSAT
Σ (ours)	70.3	68.9	26.9	65.1
Σ isotropic	69.4	68.0	26.4	64.1
Δ	-0.9	-0.9	-0.5	-1.0

Table 5.6 Performance and runtime comparison between TransCLIP and prompt learning solutions on average over ImageNet and the 10 fine-grained classification datasets.

(a) Zero-shot setting.

	Performance	Runtime
UPL*	69.8	>150 min
TransCLIP	**70.3**	**14.4 sec**

(b) Few-shot setting (4-shot).

	Performance	Runtime
CoOp+UPL*	74.4	>12h
TransCLIP-FS	**75.9**	**35.3 sec**

5.6 Ablation Studies

Component analysis

We study the impact of the principal components involved in the TransCLIP procedure on four diverse datasets. Table 5.5a shows that updating $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ significantly boosts TransCLIP's performance. This indicates the importance of having a dynamic parametric model instead of a fixed one. Table 5.5b demonstrates the critical role of text-driven penalty for TransCLIP in the zero-shot setting. Alongside the prior findings from Table 5.4, it is evident that incorporating text information is key to the success of TransCLIP and its wide applicability across the zero- and few-shot learning scenarios. The number of nearest-neighbors considered in the Laplacian term (Eq. (5.2)) does not make a significant difference in TransCLIP's performance as suggested by Table 5.5c. However, removing the Laplacian regularization (Table 5.5a) leads to inferior results on some datasets such

Table 5.7 Performance of TransCLIP (zero-shot) for increasingly large VLMs. Relative Δ is the improvement normalized by the zero-shot error: ($\text{ACC}_{\text{TransCLIP}}$ - $\text{ACC}_{\text{Zero-Shot}}$) / (100 - $\text{ACC}_{\text{Zero-Shot}}$).

		ImageNet			Average (11 datasets)		
	#Params	Zero-shot	w/ TransCLIP	relative Δ	Zero-shot	w/ TransCLIP	relative Δ
CLIP-ViT-B/16	177M	66.6	**+3.7**	**+11 %**	65.3	**+5.0**	**+14 %**
CLIP-ViT-L/14	427M	72.9	**+4.3**	**+16 %**	72.5	**+4.9**	**+18 %**
EVA-CLIP-8B	7.5B	82.5	**+2.1**	**+12 %**	81.5	**+4.3**	**+23 %**

as ImageNet and EuroSAT. We choose to consider 3 nearest-neighbors to make the affinity matrix W sparse and reduce memory consumption. We also investigate the diagonal covariance matrix design by restricting it to be isotropic (i.e., $\mathbf{\Sigma} = \sigma^2 \mathbf{I}_d$ with $\mathbf{I}_d$ the identity matrix). Table 5.5d shows that a non-isotropic $\mathbf{\Sigma}$ performs better without significantly increasing the number of trainable parameters.

Scaling to larger models

We report TransCLIP zero-shot performance on the EVA-CLIP 8 billion parameter version [143] (approximately 42 times larger than CLIP-ViT-B/16). It is worth mentioning that TransCLIP is easily applicable to multi-billion parameter models since it does not necessitate gradient computation or model parameter training (i.e., it only requires the memory needed for single-sample inference because the whole dataset processing can be performed one sample at a time). Table 5.7 shows that transduction can also bring significant improvements to larger models (detailed results are provided in the Appendix of the TransCLIP paper [5]).

5.7 Additional Zero-Shot Results on Satellite Imagery

We extend our evaluation to the specialized domain of remote sensing. In this section, we leverage the same diverse suite of 10 benchmarks introduced in Chapter 3 for inductive few-shot learning. However, we now evaluate them in the transductive zero-shot setting, testing the ability of TransCLIP to adapt specialized vision-language models without any labeled support samples.

5.7.1 Experimental Setup

Datasets
The zero-shot scene classification performance is evaluated on 10 RS benchmark datasets: AID (AID) [85], EuroSAT (SAT.) [68], MLRSNet (MLRS) [86], OPTIMAL31 (OPT.) [87], PatternNet (Pat.) [88], RESISC45 (RES.) [89], RSC11 (RSC.) [90], RSICB128 (R128) [91], RSICB256 (R256) [91], and WHURS19 (WHU.) [92]. These datasets cover a wide range of sizes, with sample counts ranging from $\sim 10^3$ to $\sim 10^5$ per dataset. They also include varying numbers of classes, from coarse-grained classification with 10 classes to fine-grained scene classification with up to 46 classes. Note that none of the chosen VLMs was fine-tuned on any of the listed datasets.

Models
We test TransCLIP on four VLMs: CLIP [16], RemoteCLIP [93], SkyCLIP [95], and GeoRSCLIP [94], all with various model architectures to generate their respective image embeddings. Using RS text-prompt templates from [94], 106 individual text embeddings were averaged to get a single textual embedding per class.

Evaluation protocol
The goal of this experiment is to evaluate the transferability of the TransCLIP zero-shot procedure to specialized contrastive VLMs with fixed hyperparameters, which are taken from Experiment 1. Each reported performance is the top-1 accuracy on the test set, averaged over 3 random seeds.

5.7.2 Results

Zero-shot improvement
We observe the top-1 accuracy with TransCLIP. We can clearly see a massive performance improvement across all benchmarks and models. We find that the addition of TransCLIP provides average gains ranging from 9.9% to 17.1% across all benchmarks and models.

Model size
Interestingly, TransCLIP produces notable improvements even when the inductive model's top-1 accuracy performance is already high. For example, when GeoRSCLIP ViT-H/14 is applied to WHURS19, the top-1 accuracy increases from 90.4% to 99.7%. Similarly, for the same model applied

Table 5.8 Top-1 accuracy for zero-shot scene classification without (white) and with (blue) TransCLIP on 10 RS datasets.

	Model	AID	SAT	MLRS	OPT.	Pat.	RES.	RSC11	R128	R256	WHU.	Average
ResNet-50	CLIP	55.4	28.3	45.0	64.5	46.4	52.8	56.7	23.4	30.4	71.3	47.4
	+ TransCLIP	**+14.2**	**+19.8**	**+9.3**	**+15.2**	**+22.6**	**+16.7**	**+21.0**	**+10.8**	**+16.4**	**+24.6**	**+17.1**
	RemoteCLIP	89.1	26.7	43.0	64.0	43.6	51.6	67.0	15.0	36.4	95.4	53.2
	+ TransCLIP	**+4.2**	**+7.8**	**+15.0**	**+21.0**	**+10.0**	**+21.2**	**+20.2**	**+4.1**	**+11.8**	**+3.0**	**+11.8**
ViT-B/32	CLIP	66.4	45.3	51.2	73.0	59.6	60.7	55.5	27.7	40.3	81.1	56.1
	+ TransCLIP	**+14.3**	**+3.6**	**+13.0**	**+9.9**	**+16.9**	**+13.4**	**+11.5**	**+5.6**	**+6.0**	**+9.3**	**+10.4**
	GeoRSCLIP	70.3	53.4	65.0	79.6	75.8	68.8	68.3	29.0	46.5	88.8	64.5
	+ TransCLIP	**+7.9**	**+15.5**	**+6.9**	**+7.7**	**+18.6**	**+10.7**	**+10.3**	**+13.8**	**+15.3**	**+10.0**	**+11.7**
	RemoteCLIP	91.7	35.5	56.3	77.6	55.9	68.1	61.8	26.0	41.5	95.2	61.0
	+ TransCLIP	**+3.9**	**+15.5**	**+9.5**	**+10.3**	**+14.8**	**+11.2**	**+17.9**	**+5.1**	**+7.7**	**+2.7**	**+9.9**
	SkyCLIP50	70.3	52.6	63.2	79.5	73.8	66.7	61.2	39.0	47.1	91.0	64.5
	+ TransCLIP	**+8.3**	**+11.9**	**+10.1**	**+5.8**	**+13.8**	**+10.6**	**+15.9**	**+10.4**	**+11.9**	**+6.8**	**+10.5**
ViT-L/14	CLIP	69.7	60.1	64.1	80.6	74.7	71.3	67.3	37.9	47.2	85.5	65.8
	+ TransCLIP	**+14.4**	**+11.9**	**+10.4**	**+11.7**	**+17.1**	**+10.9**	**+13.2**	**+5.9**	**+3.3**	**+13.6**	**+11.3**
	GeoRSCLIP	74.4	59.9	66.7	83.7	77.4	73.8	75.0	33.7	52.2	88.5	68.5
	+ TransCLIP	**+6.0**	**+12.8**	**+7.3**	**+9.9**	**+15.7**	**+12.9**	**+6.9**	**+19.9**	**+12.4**	**+10.4**	**+11.4**
	RemoteCLIP	84.1	43.6	62.2	83.8	61.4	76.0	67.8	34.8	50.7	93.5	65.8
	+ TransCLIP	**+8.6**	**+8.7**	**+9.3**	**+5.5**	**+20.7**	**+8.1**	**+20.3**	**+10.1**	**+9.8**	**+3.7**	**+10.5**
	SkyCLIP50	72.1	51.5	64.0	80.9	75.3	70.5	66.8	38.0	46.6	87.5	65.3
	+ TransCLIP	**+18.7**	**+17.0**	**+9.2**	**+11.9**	**+18.3**	**+11.1**	**+13.7**	**+13.4**	**+15.4**	**+11.7**	**+14.0**
H/14	GeoRSCLIP	76.3	68.3	67.4	84.8	82.7	73.8	77.4	43.1	56.5	90.4	72.1
	+ TransCLIP	**+7.5**	**+22.9**	**+10.7**	**+9.7**	**+13.5**	**+14.2**	**+5.9**	**+11.7**	**+16.3**	**+9.3**	**+12.1**

to PatternNet, the top-1 accuracy improves from 82.7% to 96.2%. This shows TransCLIP's applicability for tasks where these VLMs are already effective, without any labels. We also notice that, for the ViT-L/14 backbone, TransCLIP offers slightly higher gains to SkyCLIP50 compared to CLIP and RemoteCLIP, allowing it to outperform them when combined with transduction. TransCLIP also demonstrates its applicability to more robust models, bringing an average gain of 12.1% on GeoRSCLIP ViT-H/14.

Computational cost

We evaluate the computational cost of TransCLIP using three datasets of varying sizes. As shown in Table 5.9, the feature extraction time increases with the number of image patches while the additional load from TransCLIP remains minimal. This can be of particular interest when image analysis of remote sensing images is subject to strict time constraints, for example in case of emergency disaster response. Thus, by not requiring optimization of model parameters or input prompts, our transductive method ensures fast inference all while boosting model accuracy.

Table 5.9 TransCLIP run time on top of CLIP ViT-L/14, evaluated with 24GB NVIDIA GeForce RTX 4090 GPU.

Dataset	Nb of images	Encoding time	TransCLIP time
WHU.	$\sim 10^3$	$\sim$ 8 seconds	$\sim$ 0.3 seconds
AID	$\sim 10^4$	$\sim$ 40 seconds	$\sim$ 2 seconds
MLRS	$\sim 10^5$	$\sim$ 6 minutes	$\sim$ 25 seconds

5.8 Extension to a New Domain: Computational Pathology

Histology slides obtained from Whole Slide Image (WSI) [153] scanners play a crucial role in cancer diagnosis and staging [154]. These slides offer a detailed view of diseased tissues, aiding in the determination of treatment options. Pathologists primarily diagnose cancers by examining WSIs to identify different tissue types. However, manually analyzing these WSIs imposes a significant workload, leading to substantial delays in reporting time. Moreover, in real clinical environments, the classification of cancer-related tissues is highly diverse, encompassing various cancer sites. Even within a single cancer site, tasks can vary in their levels of class granularity. Therefore, automating tissue-type classification in histology images ([155–159] to list a few) holds significant clinical value but is hindered by the difficulty of collecting large labeled datasets and the variability of fine-grained labels.

The advent of multi-modal learning methods, in particular VLMs, that process and integrate information from diverse modalities has alleviated some issues of training fine-grained classifiers and collecting costly labeled data. This new multi-modal paradigm can naturally be extended to clinical scenarios, where combinations of multiple data modalities (mainly texts and images) are often adopted to obtain more accurate and comprehensive diagnosis. For example, clinical notes and pathology reports, alongside histopathology slides, are commonly used for thorough analysis [160]. However, the direct application of deep learning techniques, more specifically vision-language pre-training strategies, to medical imaging is complex, due to the lack of fine-grained expert medical knowledge, which is required to capture specialized information [161]. This issue has been partly addressed for histopathology slides by collecting diverse data from scientific publications, Twitter, or even YouTube videos [162–164].

With the ongoing development of foundation models in medical imaging and specifically histopathology, and the potential application of transductive inference, our objective is to improve zero-shot predictions of VLMs within this framework.

5.8.1 Experimental Setup

Datasets
We study different histopathology classification tasks on various organs and cancer types. Specifically, *NCT-CRC* [165] comprises patches of colorectal adenocarcinoma categorized into 9 classes, *SICAP-MIL* [166] includes 4 prostate cancer grading, *SKINCANCER* [167] is annotated with 9 anatomical tissue structures, and *LC25000(Lung)* [168] focuses on 3 classes of lung cancer. These diverse benchmarks enable the study of the generalization capability of VLMs pretrained on histology images.

Models
We compare several vision-language models pretrained on histology images, namely PLIP [162], QUILT [163] and CONCH [164].

Evaluation protocol
As for the experiment on satellite imagery, the goal is to evaluate the transferability of the TransCLIP zero-shot procedure to specialized contrastive VLMs with fixed hyperparameters, which are taken from Experiment 1. Each reported performance is the top-1 accuracy on the test set, averaged over 3 random seeds. Text embeddings for each category are obtained following the specific 22 prompts used for CONCH (only one name is assigned to each target class), which are then averaged to get a single textual embedding per class.

5.8.2 Results

Transductive performance gains
Applying TransCLIP consistently boosts performance across models and datasets. We observe average accuracy gains of +7% for CONCH and up to +15.6% for Quilt-B16. This confirms that the transductive alignment of test batches effectively mitigates the domain shift even in specialized

Table 5.10 TransCLIP performance on top of various VLMs. Best values are highlighted in **bold**. $\Delta_{\text{transductive}}$ is the average accuracy gain brought by our transductive approach.

Dataset	Method	Model				
		CLIP	Quilt-B16	Quilt-B32	PLIP	CONCH
SICAP-MIL	zero-shot	**29.9**	40.4	**35.0**	46.8	27.7
	+TransCLIP	24.7	**58.5**	28.2	**53.2**	**32.6**
LC(Lung)	zero-shot	**31.5**	43.0	76.2	85.0	84.8
	+TransCLIP	25.6	**50.5**	**93.9**	**93.80**	**96.3**
SKINCANCER	zero-shot	4.2	15.4	39.7	22.9	58.5
	+TransCLIP	**11.5**	**33.3**	**48.8**	**36.7**	**66.2**
NCT-CRC	zero-shot	25.4	29.6	53.7	63.2	66.3
	+TransCLIP	**39.6**	**48.4**	**58.1**	**77.5**	**70.4**
Average	zero-shot	22.7	32.1	51.2	54.5	59.3
	+TransCLIP	**25.4**	**47.7**	**57.3**	**65.3**	**66.4**
	$\Delta_{\text{transductive}}$	**+2.6**	**+15.6**	**+6.1**	**+10.9**	**+7.0**

medical tasks. TransCLIP demonstrates a strong ability to refine text-based predictions, effectively improving the initial zero-shot prototypes.

Robustness and failure modes

In the vast majority of cases, TransCLIP enhances performance. However, we observe marginal drops in rare cases where the initial zero-shot performance is extremely low. This is an expected behavior of the text-driven KL regularization: if the textual priors ($\hat{\mathbf{s}}_i$) are too noisy or misaligned, they may provide a misleading signal to the transductive optimization.

5.9 Conclusion of Chapter 5

In this chapter, we investigated the transductive paradigm within the context of Vision-Language Models, introducing **TransCLIP**. Our algorithm is highly efficient, as it operates solely in the output embedding space (i.e., black-box setting), making it suitable for a wide range of models, including very large ones. This also enables TransCLIP to be compatible with models that are accessible only through APIs. We first showed how TransCLIP can bring transduction to the zero-shot setting, achieving consistent gains without additional supervision. Subsequently, we proposed a new setting

that applies transduction on top of popular few-shot methods, offering a convenient strategy to combine computationally intensive supervised fine-tuning with efficient test-time transduction. Finally, we proposed a simple extension of TransCLIP to incorporate labeled samples. We showed that transductive learning can bring substantial improvement in a large variety of settings and domains, including natural images, satellite imagery, and computational pathology. Alongside our open-sourced codebase[4], we provide a strong baseline for future research in this direction. As in Chapter 4, we demonstrate again that high-performing methods can be designed without heavy training procedures, and that a simple prior (here, the zero-shot prediction) can be integrated smartly into a single objective function. In all our experiments, TransCLIP's text-guided KL divergence term appears as a key factor in its success. Lastly, we advocate for fixed hyperparameters across experiments as already discussed in Chapter 3 or, at the very least, transparency regarding their existence and guidelines on how to tune them for each specific task.

[4] https://github.com/MaxZanella/transduction-for-vlms

5.10 Appendix of Chapter 5

5.10.1 More Details on Convergence

As mentioned in the main paper, our derived block-wise optimization procedure in Eqs. (5.5), (5.6) and (5.7) can be viewed as an instance of the the general Block Majorize-Minimize paradigm for non-convex optimization, also referred to as the Block Successive Minimization (BSUM) method [152]. We update each block of variables, with the other blocks fixed, by minimizing a tight upper bound (majorizing function), thereby guaranteeing the overall objective does not increase at each step. In the steps with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, we optimize directly the objective in closed-form, which could be also viewed as a particular case of optimizing a tight upper bound. The convergence of the general BSUM procedure is well studied in the optimization community [152]. Indeed, under the following assumptions for each block of variables, one can establish convergence results for the application of BSUM to non-convex problems [152]:

- A1: The majorizing function is a tight upper bound, i.e., equal to the objective at the current solution.
- A2: The first-order behavior of the majorizing function is the same as the original objective locally.

Indeed, when assumptions A1 and A2 are verified for each block, we have the result in Theorem 5.1 [152].

As for our case of alternating Eqs. (5.5), (5.6) and (5.7), it is straightforward to verify that Assumptions A1 and A2 are satisfied for each block of variables. Furthermore, the majorizing functions are convex and thus quasi-convex. Also, the sub-problem solved for each block has a unique solution. In particular, for the $\mathbf{z}$-updates, the majorizing function is the sum of a linear and a strongly convex function (the negative entropy). Therefore, it is strongly convex. As for the $\boldsymbol{\mu}$- and $\boldsymbol{\Sigma}$-updates, the solutions are obtained in closed form (hence unique).

6

Transductive Learning in the Wild

Preliminary Note:

This chapter expands on conference proceedings [8], introducing **Stat**$\mathcal{A}$, a robust transductive method designed for *realistic* deployment conditions. It builds upon the TransCLIP procedure to handle challenging scenarios, such as **varying class distributions** and **correlated data streams**, by incorporating a novel *statistical anchor regularization*.

Before starting the chapter, readers are advised to review the fundamentals of CLIP in Chapter 2. Additionally, a good understanding of the TransCLIP methodology (specifically Section 5.3) could be useful.

6.1 Introduction

The previous chapter investigated adaptation strategies within somewhat idealized settings, where experimental conditions are favorable. While these standardized benchmarks facilitate rapid prototyping and direct comparison to previous methods, they often lack the complexity of *realistic* deployment scenarios. In real-world applications, perfect conditions are rarely met; batch sizes vary, data streams may be correlated, and class distributions are often imbalanced. Although no benchmark can perfectly mirror reality, it is crucial to rigorously stress-test the reliability of current state-of-the-art methods. We aim to push the boundaries of standard evaluation protocols by challenging the validity of results obtained under simplified assumptions. To this end, this chapter explores several critical edge cases, moving towards *transductive learning in the wild*.

6.1.1 Context and Research Positioning

As demonstrated multiple times in the previous chapters of this thesis, Vision-Language Models (VLMs) have introduced a powerful framework that connects images and texts, enabling models to adapt to new concepts without costly, human-labeled training data. This is achieved through a pre-training phase during which an image and its associated text description are aligned through contrastive learning. This enables zero-shot recognition, where novel images could be categorized by matching them to textual class descriptions, supporting tasks beyond traditional supervised learning.

VLMs such as CLIP [16] have triggered wide interest, particularly in the few-shot adaptation setting [1,25,27,58,59] (Chapter 3), in which they have shown promising generalization capabilities using limited labeled data for each downstream task. More recently, the focus has shifted towards even more challenging unsupervised scenarios, where models must adapt on-the-fly to the test data. Test-time adaptation (TTA) approaches have thus emerged, covering a spectrum of data availability regimes: from test-time augmentation on a single image [4, 47, 48, 169] (Chapter 4), to transductive inference leveraging global statistics on batches [5, 142, 170] (Chapter 5 and this chapter), and finally to continuous adaptation on incoming data streams [171, 172] (developed in the following sections).

TTA is very popular in the vision community [138, 173–177], and this interest is now extending to VLMs [5, 142, 170–172], where specific methods are required to fully exploit their open-vocabulary pre-training and zero-shot capabilities. However, while real-world scenarios frequently involve highly correlated incoming samples—such as patches from a satellite image or video recordings, as shown in Figure 6.1—previous studies still lack a *realistic* evaluation, i.e., one that envisions test-time class-distribution sampling to mimic what is encountered in real-life deployments.

By considering a breadth of realistic scenarios, we found that existing TTA or transductive methods for VLMs are often highly biased toward settings with a small number of effective classes (i.e., the actual number of classes in the batch) [142], where the classes are uniformly distributed [5, 170], or where the streams of samples are i.i.d. [171, 172]. In contrast, our method makes no assumptions about the number of effective classes in the batch, and can handle a much wider range of scenarios.

More specifically, we develop two *realistic* TTA evaluation settings for VLMs. First, we argue that adaptation methods operating on a single batch must demonstrate robustness when dealing with varying numbers of effective classes. While this aspect has been discussed recently in the context of VLMs [142], current methods are effective only within a narrow and specific range of class numbers. In contrast, we perform a comprehensive evaluation of methods over a broader range of effective class numbers. Secondly, online adaptation should be resilient to correlated batches, as already discussed in TTA for vision models [175–177].

Nevertheless, we show that current TTA and transductive methods designed for VLMs frequently compromise the models' initial zero-shot robustness across all these scenarios in exchange for performance gains in well-specified conditions. In response to these limitations, we introduce Stat.$\mathcal{A}$, a more resilient transductive method. Stat.$\mathcal{A}$ integrates a novel regularization term specifically designed for VLMs, serving as a statistical anchor to preserve the initial textual class representations, particularly in low-data regimes. Stat.$\mathcal{A}$ is highly efficient, processing thousands of samples within seconds.

6.1.2 Summary of Contributions

In this chapter, we introduce **Stat.$\mathcal{A}$**, a method designed to enhance the robustness of Vision-Language Models in realistic deployment conditions. Our core contributions are threefold:

(i) We analyze and rigorously challenge the limitations of current transductive and test-time adaptation techniques. We demonstrate their systematic failure modes in *realistic* scenarios characterized by a varying number of present classes, and temporal correlations.

(ii) To address these shortcomings, we introduce a novel evaluation framework comprising two demanding settings: adaptation within a single batch containing a variable number of effective classes, and online adaptation on non-i.i.d. sample streams.

(iii) We propose a versatile transductive algorithm featuring a novel regularization term dubbed the Statistical Anchor, which allows our method to robustly handle an arbitrary number of effective classes while maintaining high performance across these challenging distributions.

6.1.3 Chapter Outline

The remainder of this chapter is structured as follows. Section 6.2 defines the realistic evaluation protocols and motivates the need for more rigorous testing scenarios. Section 6.3 details our core contribution, Stat.$\mathcal{A}$, introducing the realistic transductive learning formulation and deriving the regularized updates for the GMM parameters. Next, Sections 6.4 and 6.5 present a comprehensive experimental validation, systematically benchmarking our approach against state-of-the-art methods in challenging batch and online test-time adaptation scenarios, respectively. Section 6.6 analyzes the impact of key components through ablation studies. We conclude with the main insights of this chapter in Section 6.7. Finally, mathematical derivations regarding the Kullback-Leibler divergence and the regularized parameter updates are provided in the Appendix (Section 6.8).

6.2 Towards More *Realistic* Scenarios

The experimental settings currently envisioned to validate transduction or TTA methods for VLMs rely on unrealistic assumptions about the data dis-

tribution. In contrast, our work aims to tackle real-world deployment conditions, which we refer to as *realistic* scenarios. Figure 6.1 illustrates two of those scenarios of interest for TTA deployment. In our experiments, we have divided the study of these *realistic* scenarios into two perspectives:

Batch *realistic* scenarios

In practical applications, batches often contain a limited number of effective classes, denoted as K_{eff}. For instance, a user will define a large panel of classes of interest whereas only a subset is actually present in the data at hand. In this first setup of interest, each batch is processed independently, for example when processing large satellite images decomposed into patches (Figure 6.1a). In this case, we vary the number of effective classes, simulating situations where a batch might not contain all the classes of interest, considering six scenarios: `Very Low` contains from 1 to 4 classes; `Low` contains from 2 to 10 classes; `Medium` contains from 5 to 25 classes; `High` contains from 25 to 50 classes; `Very High` contains from 50 to 100 classes; `All` includes all classes.

Online *realistic* scenarios

In our second perspective on the notion of realistic scenario, we examine streams of batches, akin to processing a sequence of images. In this case, it is known that real-world data distributions, such as those encountered in autonomous driving and human activity recognition (Figure 6.1b), tend to exhibit temporal correlations. Following previous works, we control this correlation using a Dirichlet distribution [175–177], denoted by γ.

Hyperparameters

Generalization to unseen cases is crucial for TTA methods. Therefore, optimizing hyperparameters for each scenario would require access to labels and prior knowledge of the specific scenario encountered during testing, which goes against the core purpose of the TTA approach. We found that some methods largely rely on dataset-specific hyperparameters without clear guidance on how to tune them for a new dataset, or even resort to hyperparameter search using knowledge from ground truth labels. To ensure fairness in comparison, we use hyperparameters optimized on a single dataset for all reported experiments. This is further discussed in the experimental sections.

(a) Satellite images of an area may not contain certain classes. Images are taken from the EuroSAT [68] dataset.

(b) Video recordings may contain multiple consecutive frames of the same class. Images are taken from the UCF101 [76] dataset.

Fig. 6.1 Illustration of two *realistic* scenarios: (a) batch adaptation with limited number of effective classes and (b) online test-time adaptation with a correlated, non-i.i.d. data stream.

6.3 *Realistic* Transductive Learning

Current TTA or transductive methods based on probabilistic clustering often hold implicit assumptions about the statistics of the testing batch, such as uniform class distributions and low effective numbers of classes. As a result, they may have difficulty generalizing to diverse scenarios. We introduce a versatile method that can handle a breadth of scenarios, with varying batch sizes and numbers of effective classes, without any hyperparameter tuning, while maintaining a strong overall performance. We derive our method as Maximum Likelihood Estimation (MLE) tailored to VLMs, with a novel Statistical Anchor (Stat.$\mathcal{A}$) term that leverages the text-encoder knowledge, acting as a regularizer on the statistical parameters of the vision features (i.e., the mean vectors and covariance matrices).

6.3.1 Formulation

We build directly upon the CLIP framework of Chapter 2 and the TransCLIP methodology (Chapter 5). To ensure this section remains self-contained, we briefly recall the essential notations and definitions below, while encouraging readers to refer to the aforementioned chapters for a more in-depth background.

We seek to predict a label k for each image in the *query* set, $\mathbf{x}_i$ $i \in \mathcal{Q}$, given pre-trained vision-encoded features $\mathbf{f}_i = \theta_v(\mathbf{x}_i) \in \mathbb{R}^d$ and text embeddings $\mathbf{t}_k = \theta_t(\mathbf{c}_k) \in \mathbb{R}^d$, where θ_v denotes the vision encoder, θ_t the text encoder, and $\mathbf{c}_k$ the text prompt representing class k, $k = 1, \ldots, K$. The total number of all possible classes is K, while the effective number of classes occurring in a given query set is K_{eff}, which might be lower but no greater than K.

Regularized Maximum Likelihood Estimation

The proposed method belongs to the broad family of soft probabilistic clustering approaches, which jointly estimate:

- Assignment vectors $\mathbf{z} = (\mathbf{z}_i)_{1 \leq i \leq N_\mathcal{Q}}$ within the probability simplex $(\Delta_K)^{N_\mathcal{Q}}$. The k-th component $z_{i,k}$ of vector $\mathbf{z}_i$ yields the probability that the i-th sample is in class k.
- Statistical models $\mathbf{M} = (\mathbf{M}_k)_{1 \leq k \leq K}$. Each $\mathbf{M}_k$ contains the set of parameters of a distribution modeling the features within class k, e.g., a mean vector and a covariance matrix in the case of multivariate Gaussian distributions.

In the context of transductive few-shot methods and TTA, several recent methods could be included in this family of clustering-based approaches, e.g., PADDLE [129], Dirichlet [142], LaplacianShot [130] and TransCLIP [5], among others. These methods minimize Maximum Likelihood Estimation (MLE) objective functions of the following general form, alternating optimization updates w.r.t both the assignment variables in $\mathbf{z}$ and the model parameters in $\mathbf{M}$:

$$\mathcal{L}_{\text{MLE}}(\mathbf{z}; \mathbf{M}) = -\sum_{i \in \mathcal{Q}} \mathbf{z}_i^\top \log(\mathbf{p}_i) + \mathcal{R}(\mathbf{z}) \tag{6.1}$$

where $\mathbf{p}_i = (p_{i,k})_{1 \leq k \leq K} = (\Pr(\mathbf{f}_i | \mathbf{M}_k))_{1 \leq k \leq K}$ contains likelihood probability scores evaluating how likely is feature vector $\mathbf{f}_i$, given the k-th class

parametric density $\mathbf{M}_k$, and $\mathcal{R}$ is a regularization term, characteristic of the method. These methods may also differ in how they model likelihoods $p_{i,k}$. For instance, TransCLIP [5] is based on multivariate Gaussian models, whereas the authors of [142] used Dirichlet distributions. In fact, the first term in Eq. (6.1) is the general log-likelihood model fitting objective, well established in the clustering literature as a probabilistic generalization of K-means[1] [129,146]. This general term has an inherent bias to class-balanced clusters, a well-known fact in the clustering literature [146,178].

As for regularization term $\mathcal{R}(\mathbf{z})$, there are several choices in the recent transductive learning literature, some of which also embed priors on class statistics, inducing biases towards specific scenarios. For instance, PADDLE [129] and Dirichlet [142] explicitly penalize the number of non-empty clusters via a minimum description length (MDL) regularizer. While this regularizer could counter the class-balance bias in the log-likelihood term in Eq. (6.1), we found that these methods are strongly biased towards small numbers of effective classes; see Table 6.1b and 6.1c. LaplacianShot [130] uses a Laplacian regularizer, encouraging samples with nearby representations (in the feature space) to have similar predictions. TransCLIP [5] adopts a combination of this Laplacian term and a Kullback-Leibler divergence regularizer, which penalizes the deviation of each assignment variable $\mathbf{p}_i$ from the zero-shot text-driven softmax prediction of the i-th sample.

Proposed Statistical Anchor (Stat$\mathcal{A}$) term

It is notable that all the state-of-the-art transductive learning methods mentioned in the previous section regularize the assignment variables in $\mathbf{z}$, via the regularizer $\mathcal{R}(\mathbf{z})$, but not statistical model parameters $\mathbf{M}$ that appear in the log-likelihood term in Eq. (6.1). However, in the context of VLMs, and as these learn from aligning vision and language representations, we argue that useful priors about those statistical parameters could be leveraged from the text-encoder knowledge. For instance, one could derive prototypes from the textual prompts. In the following, we introduce a Statistical Anchor (Stat$\mathcal{A}$) term, which acts as a regularizer on the model parameters using text-driven priors. As will be shown in our experiments, Stat$\mathcal{A}$ brings important gains in robustness, yielding strong performance across all settings.

[1] The K-means objective corresponds to multivariate Gaussian models, but with the simplifying assumption fixing all the covariance matrices to the identity matrix.

In the following, we develop Stat.$\mathcal{A}$ under the multivariate Gaussian assumption for models $(\mathbf{M}_k)_{1\leq k\leq K}$, but the concept could be extended to other density functions in the exponential family. Assume that the feature vectors $\mathbf{f}_i$ within a class k are random variables following a multivariate Gaussian distribution $\mathcal{N}_k = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, with a mean vector $\boldsymbol{\mu}_k$ and a diagonal covariance matrix $\boldsymbol{\Sigma}_k$, i.e., $\mathbf{M}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Thus, the likelihood probabilities appearing in the general problem in Eq. (6.1) read as follows:

$$p_{i,k} \propto \frac{1}{\sqrt{|\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\mathbf{f}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{f}_i - \boldsymbol{\mu}_k)\right) \tag{6.2}$$

We introduce an additional penalty, which regularizes the statistical parameters of multivariate Gaussian $\mathcal{N}_k$. Specifically, we exploit the textual prompts to build an initial estimation of the mean vectors and covariance matrices, i.e., a fixed multivariate Gaussian distribution $\mathcal{N'}_k = \mathcal{N}(\boldsymbol{\mu'}_k, \boldsymbol{\Sigma'}_k)$, which serves as a statistical anchor (Stat.$\mathcal{A}$). Then, we add a Kullback-Leibler term that penalizes the overall objective when the multivariate Gaussian of class k deviates from these initial mean vectors $\boldsymbol{\mu'}_k$ and covariance matrices $\boldsymbol{\Sigma'}_k$, which are kept fixed and not optimized upon:

$$\begin{aligned}\mathrm{KL}\left(\mathcal{N}'_k||\mathcal{N}_k\right) = \frac{1}{2}((\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k) \\ + \mathrm{Tr}(\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\Sigma'}_k) + \log\frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma'}_k|} - d).\end{aligned} \tag{6.3}$$

Our statistical anchor term could be added to the generic MLE clustering objective defined in Eq. (6.1) and, hence, used in conjunction with any method befitting this general setting, provided that the likelihood probabilities are assumed to be multivariate Gaussian:

$$\mathcal{L}_{\mathcal{A}}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{L}_{\mathrm{MLE}}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \alpha\mathcal{A}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{6.4}$$

where $\boldsymbol{\mu} = (\boldsymbol{\mu}_k)_{1\leq k\leq K}$, $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_k)_{1\leq k\leq K}$, α is a non-negative constant, and our statistical anchor is given by:

$$\mathcal{A}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \mathrm{KL}\left(\mathcal{N}'_k||\mathcal{N}_k\right) \tag{6.5}$$

We compute mean vectors and a shared diagonal covariance matrix of the anchor (fixed) distribution $\mathcal{N}'_k$ from the text-encoder knowledge and zero-

shot predictions as follows:

$$\boldsymbol{\mu}'_k = \mathbf{t}_k;\ \boldsymbol{\Sigma}' = \text{Diag}\left(\frac{\sum_{i,k} \hat{s}_{i,k}(\mathbf{f}_i - \boldsymbol{\mu}_k)(\mathbf{f}_i - \boldsymbol{\mu}_k)^\top}{\sum_{i,k} \hat{s}_{i,k}}\right) \tag{6.6}$$

where $\hat{s}_{i,k}$ denotes the zero-shot, text-driven prediction:

$$\hat{s}_{i,k} = \frac{\exp(\mathbf{f}_i^\top \mathbf{t}_k / \tau)}{\sum_j \exp(\mathbf{f}_i^\top \mathbf{t}_j / \tau)} \tag{6.7}$$

Our overall objective in Eq. (6.4) depends on two types of variables, statistical parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and the assignments in $\mathbf{z}$. Therefore, we proceed with a block-coordinate descent scheme, which alternates two update steps, one fixing $\mathbf{z}$ and updating $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ in closed-form, and the other fixing the parameters and optimizing over $\mathbf{z}$.

6.3.2 Regularized updates of the parameters

When the variables in $\mathbf{z}$ are fixed, notice that the objective function in Eq. (6.4) is strictly convex w.r.t each $\boldsymbol{\mu}_k$ and each $\boldsymbol{\Sigma}_k$, $k = 1, \ldots, K$, for any non-negative α. Therefore, by setting the gradient with respect to each of these statistical variables equal to zero (details in Appendix 6.8.2), we obtain the following closed-form updates:

$$\begin{aligned}
\boldsymbol{\mu}_k &= \beta_k \hat{\boldsymbol{\mu}}_k + (1 - \beta_k) \boldsymbol{\mu'}_k \\
\boldsymbol{\Sigma}_k &= \beta_k \hat{\boldsymbol{\Sigma}}_k + (1 - \beta_k)(\boldsymbol{\Sigma'} + \text{Diag}((\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)^2)
\end{aligned} \tag{6.8}$$

where scalar $\beta_k \in [0, 1]$ is given by: $\beta_k = \frac{\sum_{i=1}^N z_{i,k}}{\sum_{i=1}^N z_{i,k} + \alpha}$, and $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\Sigma}}_k$ are, respectively, the sample mean vector and diagonal covariance matrix of class k:

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^N z_{i,k} \mathbf{f}_i}{\sum_{i=1}^N z_{i,k}};\quad \hat{\boldsymbol{\Sigma}}_k = \frac{\sum_{i=1}^N z_{i,k} \text{Diag}((\mathbf{f}_i - \boldsymbol{\mu}_k)^2)}{\sum_{i=1}^N z_{i,k}} \tag{6.9}$$

Interpretation

Updates in Eq. (6.8) allows for a clear interpretation. Indeed, the sample mean and covariance updates in Eq. (6.9) are the standard MLE estimates optimizing Eq. (6.1), which corresponds to the case $\alpha = 0$ in our objective in Eq. (6.4) (i.e., no anchor term). The updates in Eq. (6.8), $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, are a convex combination of these MLE estimates and a term dependent on the

statistical anchor, with scalar $\beta_k \in [0,1]$ controlling the combination.

Notice that the larger the number of samples predicted to belong to class k, the larger the value of this scalar. Therefore, the convex combination in Eq. (6.8) gets closer to the statistical anchor when few or no samples are predicted as belonging to class K, and closer to the standard MLE estimates in Eq. (6.9) otherwise. This makes sense as a small number of samples may not be sufficient to estimate reliably the mean and covariance statistics. Weighting factor β_k depends on α, which is a hyperparameter. While we have found that the straightforward choice $\alpha = 1$ (used in all experiments) works well across various settings, one could further tune it depending, for example, on the quality of the chosen anchor (see Figure 6.3 of the ablation studies).

Implementation of the weighting factor

In our numerical implementation of the algorithm, we modified the weighting factor β_k by replacing the soft assignment predictions with hard ones on the vertices of the simplex, for a better estimation of the predicted cardinality (i.e., number of samples) of each class k:

$$\beta_k = \frac{\sum_{i=1}^{N} \mathbf{1}\left[k = \arg\max_r z_{i,r}\right]}{\sum_{i=1}^{N} \mathbf{1}\left[k = \arg\max_r z_{i,r}\right] + \alpha} \tag{6.10}$$

We found Eq. (6.10) to be more robust to the noise induced by residuals in components other than the one with the maximum probability in $\mathbf{z}_i$ (see Table 6.3 of the ablation study).

6.3.3 Complete formulation

Assignments' regularization

In addition to the proposed regularization of statistical parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, we follow recent works, regularizing assignments $\mathbf{z}$. We adopt the ubiquitous Laplacian regularizer [5, 130, 179], as well as the text-supervision term from [5] designed for VLMs. The former encourages samples with close visual features to have the same predictions, whereas the latter penalizes the deviation of assignment variable $\mathbf{z}_i$ from zero-shot predictions $\hat{\mathbf{s}}_i = (s_{i,k})_{1 \le k \le K}$. More precisely, we set regularizer $\mathcal{R}(\mathbf{z})$, which appears in

Eqs. (6.1) and (6.4), as follows:

$$\mathcal{R}(\mathbf{z}) = \underbrace{-\sum_{i,j} w_{ij}\mathbf{z}_i^\top \mathbf{z}_j}_{\text{Laplacian reg.}} + \underbrace{\sum_{i\in\mathcal{Q}} \mathrm{KL}(\mathbf{z}_i||\hat{\mathbf{s}}_i)}_{\text{text supervision}} \tag{6.11}$$

where $w_{ij} = \mathbf{f}_i^\top \mathbf{f}_j$. With parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ fixed, and using the regularizer in Eq. (6.11), the overall objective in Eq. (6.4) becomes the sum of concave and convex functions of $\mathbf{z}$. Therefore, we could deploy a Concave-Convex Procedure (CCCP) minimization procedure [96], which, at each update step, minimizes a tight convex upper bound on the objective. This guarantees that each update does not increase the objective. CCCP replaces the concave part with its linear first-order approximation at the current step, which is a tight upper bound, while keeping the convex part unchanged. The Laplacian term is concave whereas the remaining part of the objective is convex. Solving the necessary conditions for minimizing the convex upper bound gives the following decoupled updates of the assignment variables (the proof is similar to the one in Appendix of the previous chapter, Section 5.10):

$$\mathbf{z}_i^{(l+1)} = \frac{\hat{\mathbf{s}}_i \odot \exp(\log(\mathbf{p}_i) + \sum_j w_{ij}\mathbf{z}_j^{(l)})}{(\hat{\mathbf{s}}_i \odot \exp(\log(\mathbf{p}_i) + \sum_j w_{ij}\mathbf{z}_j^{(l)}))^\top \mathbf{1}_K} \tag{6.12}$$

6.4 Experiment 1: Batch Test-Time Adaptation

6.4.1 Experimental Setup

Datasets

We use conventional benchmarks composed of 11 datasets for fine-grained classification, namely Imagenet (I.) [17], SUN397 (SUN) [69], Aircraft (Air.) [70], EuroSAT (SAT) [68], StanfordCars (Cars) [67], Food101 (Food) [71], OxfordPets (Pets) [72], Flowers102 (Flow.) [73], Caltech101 (Calt.) [74], DTD (DTD) [75], UCF101 (UCF) [76]. These datasets offer a thorough benchmarking framework for evaluating zero-shot adaptation on visual classification tasks. See Table 8.3 for details on the number of samples and classes.

Models
We use the ViT-B/16 CLIP model, while additional results with four other backbones can be found in the Appendix of the Stat$\mathcal{A}$ paper [8].

Evaluation protocol
Each task corresponds to a batch from the test set, which is processed independently. To reduce variability in results, 1,000 tasks are generated for each scenario, and the accuracies are averaged across tasks. We report results for batch sizes of 64 (with `Very Low`, `Low`, `Medium` number of effective classes) and 1,000 (with `Medium`, `High`, `Very High` number of effective classes), and the whole dataset (with `All` classes). See Section 6.2 for more details. Each reported performance metric represents the averaged top-1 accuracy.

Comparative methods
We use the same handcrafted prompts for all methods, which are listed in Table 8.1a. Due to the more versatile scenarios studied in this paper, we find that our centroid initialization based on text embeddings is much more robust, especially when the number of effective classes is reduced. Therefore, we implement it for TransCLIP instead of their original initialization based on the top-confident samples for each class. For ZLaP and Dirichlet, we follow the hyperparameters of their official implementation.

6.4.2 Results

Table 6.1a presents results for small batches containing relatively few effective classes; Table 6.1b shows results for larger batches; and Table 6.1c reports results for large-scale, whole-dataset processing. Stat$\mathcal{A}$ demonstrates robustness across all scenarios, whereas other transductive methods exhibit strong performance only within specific, narrow application ranges.

Dirichlet performs well in the `Low` setting but rapidly declines as conditions deviate, which aligns closely with the setting emphasized in their paper (3 to 10 effective classes per batch of size 75). TransCLIP excels in the `All` scenario but quickly underperforms when the number of effective classes decreases, even with relatively large batch sizes, consistent with their experimental focus on large-scale transduction with all classes present. Similarly, ZLaP shows small performance improvements when all classes are present, but struggles in more diverse settings. Stat$\mathcal{A}$ emerges as a well-balanced solution, providing stable gains across a wide range

Table 6.1 Comparison of different batch sizes and ranges of effective class numbers. The best average accuracy for each configuration is highlighted in **bold**, while the second-best is indicated with underline. Subscripts indicate improvement or degradation compared to zero-shot. Each reported performance is averaged over 1,000 tasks.

(a) Three scenarios with a batch size of 64: `Very Low` (1–4 effective classes per task), `Low` (2–10 classes), and `Medium` (5–25 classes).

	Method	AVERAGE	I.	SUN	Air.	SAT	**Cars**	**Food**	Pets	Flow.	Calt.	DTD	UCF
	CLIP	65.2	66.6	62.5	24.7	48.3	65.6	85.9	89.1	70.7	93.2	43.5	67.5
	MTA	+1.3	+2.7	+2.3	+2.7	-1.4	+2.4	+1.3	+0.3	+1.0	+0.8	+0.9	+1.5
`Very Low` (1–4)	Dirichlet	+3.3	+12.6	+13.2	+3.5	-1.1	+2.6	+2.2	-1.6	+0.5	-4.4	+6.8	+1.5
	ZLaP	-37.8	-52.1	-49.5	-16.3	-11.7	-41.9	-54.0	-32.1	-48.3	-40.8	-30.5	-38.3
	TransCLIP	-26.3	-45.0	-41.4	-13.1	-3.2	-30.9	-26.7	-16.7	-34.3	-30.9	-17.4	-29.8
	Stat$\mathcal{A}$	+5.1	+6.3	+3.5	+4.6	+8.5	+10.6	+4.4	+6.4	+6.9	-0.2	+2.6	+2.7
`Low` (2–10)	Dirichlet	+5.1	+13.5	+15.5	+3.4	-4.8	+5.9	+6.4	+3.6	+4.0	-0.2	+5.4	+3.4
	ZLaP	-30.0	-47.5	-43.5	-12.7	-1.9	-37.7	-42.4	-22.5	-39.4	-32.4	-21.1	-28.8
	TransCLIP	-24.8	-46.3	-40.1	-10.4	+5.6	-34.8	-30.3	-19.7	-29.8	-28.6	-11.9	-26.6
	Stat$\mathcal{A}$	+4.1	+6.2	+4.4	+3.0	+3.0	+7.9	+3.6	+4.6	+5.9	+0.4	+3.4	+2.1
`Medium` (5–25)	Dirichlet	+2.2	+11.1	+10.4	+1.4	-9.7	+6.0	+4.9	-0.7	+0.8	+0.5	-0.6	+0.3
	ZLaP	-20.6	-37.6	-34.6	-8.2	+0.7	-29.6	-26.8	-12.7	-27.8	-21.2	-11.5	-17.2
	TransCLIP	-22.5	-51.1	-39.7	-7.7	+9.9	-32.7	-29.6	-16.5	-25.7	-27.6	-6.0	-21.0
	Stat$\mathcal{A}$	+2.2	+4.1	+2.8	+1.3	-3.3	+5.5	+2.3	+1.7	+3.0	+0.7	+4.0	+1.6

(b) Three scenarios with a batch size of 1000: `Medium` (5–25 effective classes per task), `High` (25–50 classes), and `Very High` (50-100 classes).

	Method	AVERAGE	I.	SUN	Air.	SAT	Cars	Food	Pets	Flow.	Calt.	DTD	UCF
	CLIP	65.2	66.6	62.5	24.7	48.3	65.6	85.9	89.1	70.7	93.2	43.5	67.5
	MTA	+1.3	+2.7	+2.3	+2.7	-1.4	+2.4	+1.3	+0.3	+1.0	+0.8	+0.9	+1.5
`Medium` (5–25)	Dirichlet	-0.8	-5.7	+12.9	+2.0	-9.5	+8.5	-9.7	+1.9	+0.9	-0.8	-7.3	-2.1
	ZLaP	-23.7	-50.0	-42.4	-8.3	+0.7	-33.4	-30.4	-12.7	-30.1	-25.5	-9.3	-19.4
	TransCLIP	-8.7	-26.7	-19.8	-2.7	+14.8	-15.7	-5.3	-1.2	-12.0	-14.1	-0.6	-12.5
	Stat$\mathcal{A}$	+4.4	+4.2	+2.0	+3.7	+12.1	+8.4	+1.6	+4.0	+6.8	-0.4	+3.6	+2.7
`High` (25–50)	Dirichlet	-20.0	-49.3	-25.2	-3.7	-10.4	-0.2	-39.6	-7.8	-24.4	-12.7	-22.4	-23.9
	ZLaP	-13.0	-42.8	-30.3	-2.5	+1.0	-20.2	-11.0	-2.6	-14.5	-13.5	+0.1	-6.7
	TransCLIP	-3.3	-22.7	-12.9	+0.1	+15.7	-8.3	-2.9	+2.3	-1.6	-7.7	+4.0	-2.1
	Stat$\mathcal{A}$	+4.5	+5.3	+3.9	+1.2	+12.4	+8.0	+2.1	+2.3	+6.0	0.0	+4.4	+4.0
`Very High` (50–100)	Dirichlet	-31.6	-55.8	-46.8	-7.2	-10.5	-14.4	-56.8	-9.8	-46.4	-34.1	-24.5	-41.4
	ZLaP	-6.8	-33.9	-18.5	+0.7	+1.0	-10.4	-2.6	-1.8	-5.9	-5.3	+1.7	+0.3
	TransCLIP	-0.8	-22.1	-9.5	+0.9	+15.8	-4.7	-0.7	+2.8	+3.6	-2.7	+4.6	+3.2
	Stat$\mathcal{A}$	+3.7	+5.2	+4.6	-0.8	+12.4	+4.6	+1.2	+2.0	+3.6	+0.5	+4.5	+3.2

(c) Scenario with full dataset and all classes. Note that our hardware couldn't run Dirichlet on Imagenet due to excessive memory consumption.

	Method	AVERAGE	I.	SUN	Air.	SAT	Cars	Food	Pets	Flow.	Calt.	DTD	UCF
	CLIP	65.2	66.6	62.5	24.7	48.3	65.6	85.9	89.1	70.7	93.2	43.5	67.5
	MTA	+1.3	+2.7	+2.3	+2.7	-1.4	+2.4	+1.3	+0.3	+1.0	+0.8	+0.9	+1.5
`All`	Dirichlet	-35.7	N/A	-61.8	-9.2	-8.4	-44.4	-77.4	-12.8	-56.4	-49.2	-24.1	-49.1
	ZLaP	+1.1	+2.2	+4.7	+2.2	+0.8	+1.5	+1.0	-1.3	-2.0	-2.4	+2.0	+3.9
	TransCLIP	+5.1	+3.8	+6.4	+2.2	+17.8	+3.9	+1.2	+3.4	+5.8	-0.5	+5.1	+6.6
	Stat$\mathcal{A}$	+4.7	+3.3	+6.2	0.0	+19.0	+2.4	+1.2	+3.3	+4.5	+1.0	+4.9	+6.0

of scenarios. Results with four other backbones are presented in the Appendix of the Stat.$\mathcal{A}$ paper [8].

6.5 Experiment 2: Online Test-Time Adaptation

6.5.1 Experimental Setup

Datasets and models
The same datasets and models are used from experiment 1 (Section 6.4).

Evaluation protocol
Each task corresponds to a full pass through a generated (non i.i.d.) data stream. To reduce variability in results, 100 tasks are generated for each configuration and accuracy is averaged. We report results using a batch size of 128. Each reported performance metric represents the top-1 accuracy, the whole test set is used to generate each stream. For generating non i.i.d. data streams, we follow the setup of recent works [177] and adopt a framework based on Dirichlet distributions. Namely, we distribute each class over a fixed number of slots according to proportions drawn following a Dirichlet distribution parametrized by a single scalar parameter γ. Therefore, for large values of γ each class is evenly distributed among slots (i.i.d. data stream) while for small values each class is distributed in a single slot (highly correlated data stream). Then, samples are randomly shuffled within each slot. For every dataset and a given batch size, the number of slots is $\min\{K, \left\lfloor \frac{|Q|}{\text{batch size}} \right\rfloor\}$. This is illustrated in Figure 6.2.

Comparative methods
We use the same handcrafted prompts for all methods, which are listed in Table 8.1a. We found that TDA largely relies on dataset-specific hyperparameters without clear guidance on how to tune them for a new dataset. Similarly, DMN conducts a hyperparameter search in order to find the optimal balance between the logits obtained from the text prompts and the logits from their model's memory, using knowledge from ground truth labels. Hence, following our discussion, we use the hyperparameters optimized for ImageNet. For TDA, this means the positive logits mixing coefficient is set to 2, while the negative logits mixing coefficient is set to 0.117. For DMN, since we only consider zero-shot scenarios, we only need to set the

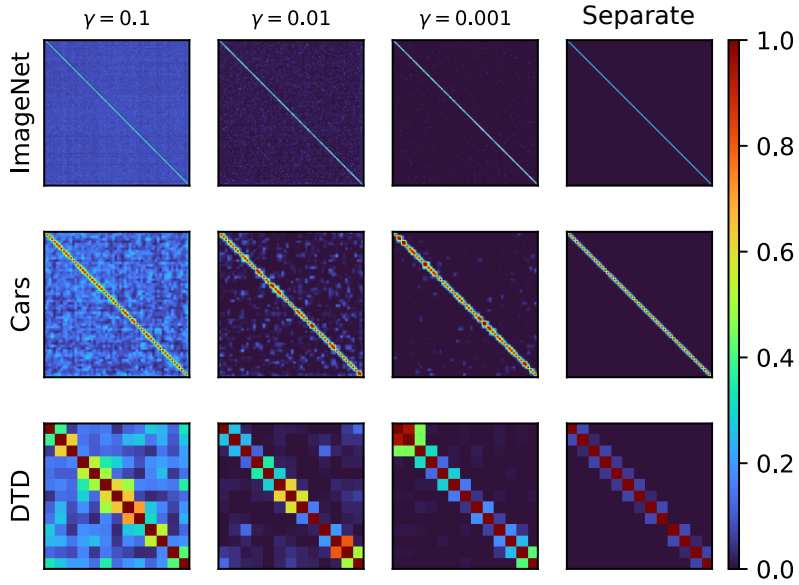


Fig. 6.2 Correlation matrix of per-batch ℓ_2 normalized vectors of class proportions for batch size 128. x and y axis of each plot is the batch index corresponding to the order in which the batches are processed. This illustrates the inter-batch correlation increasing as the Dirichlet parameter γ decreases.

coefficient relative to the dynamic memory, which is set to 1. For TENT, we use a learning rate of 0.001 and 10 steps of gradient descent per batch.

6.5.2 Results

Table 6.2 presents the results for various correlated streams. Stat.$\mathcal{A}$ demonstrates robustness across all scenarios. In contrast, while TDA and DMN do not experience as sharp a performance drop as some transductive methods from the previous sections, they still struggle to deliver significant performance gains, demonstrating that advances are still to be made for online test-time adaptation of VLMs. Moreover, TENT does not provide consistent improvements, which might be due to the absence of batch normalization layers in the CLIP encoder. In contrast, Stat.$\mathcal{A}$ benefits from more correlated streams and still emerges as a good compromise across a wide range of scenarios. Results with 4 other backbones (CNNs and ViTs) are presented in the Appendix of the Stat.$\mathcal{A}$ paper [8].

Table 6.2 Comparison of methods for online TTA. The best average accuracy for each configuration is highlighted in **bold**, while the second-best is indicated with underline. Subscripts indicate improvement or degradation compared to zero-shot. Each reported performance is averaged over 100 tasks.

(a) Four scenarios with a batch size of 128: `Low`, `Medium`, and `High` correlation. For `Separate`, classes are seen sequentially.

	Method	AVERAGE	I.	SUN	Air.	SAT	Cars	Food	Pets	Flow.	Calt.	DTD	UCF
	CLIP	65.2	66.6	62.5	24.7	48.3	65.6	85.9	89.1	70.7	93.2	43.5	67.5
	MTA	+1.3	+2.7	+2.3	+2.7	-1.4	+2.4	+1.3	+0.3	+1.0	+0.8	+0.9	+1.5
Low($\gamma = 0.1$)	TENT	+0.6	0.0	+2.0	-0.1	+3.5	+0.1	0.0	+0.2	-0.1	+0.2	+0.5	+0.3
	TDA	+2.5	+1.7	+3.5	+0.7	+12.3	+1.3	+0.2	+0.5	+1.8	+0.2	+2.0	+3.5
	DMN	+2.0	+1.4	+2.3	+0.2	+11.5	+1.4	-1.7	+0.8	+2.6	-1.0	+1.3	+2.8
	Stat.$\mathcal{A}$	+1.7	-0.4	+1.1	-0.4	+4.0	+1.8	+2.1	+3.4	+2.0	+1.0	+3.3	+1.3
Medium($\gamma = 0.01$)	TENT	+0.2	+0.1	+1.8	-0.1	-0.4	0.0	0.0	+0.3	-0.1	+0.1	+0.5	+0.3
	TDA	+1.9	+1.6	+3.1	+0.5	+8.2	+0.9	-0.1	+0.2	+1.9	+0.3	+1.7	+2.6
	DMN	+1.2	+1.4	+2.3	+0.2	+7.9	+1.2	-4.0	-0.1	+2.3	-1.1	+1.4	+2.1
	Stat.$\mathcal{A}$	+3.7	+3.0	+3.4	+2.6	+4.0	+7.6	+3.2	+5.5	+4.9	+1.1	+3.3	+2.2
High($\gamma = 0.001$)	TENT	+0.1	+0.2	+1.8	+0.1	-2.7	0.0	+0.2	+0.3	-0.2	+0.2	+0.3	+0.4
	TDA	+1.6	+1.3	+2.6	+0.4	+7.0	+0.7	-0.4	-0.1	+1.8	+0.4	+1.6	+2.2
	DMN	+1.0	+1.3	+2.3	+0.2	+8.0	+1.2	-6.0	-0.2	+2.2	-1.1	+1.3	+1.9
	Stat.$\mathcal{A}$	+4.2	+5.3	+3.5	+3.2	+3.5	+9.1	+3.4	+5.7	+5.7	+1.2	+3.5	+2.3
Separate	TENT	-0.7	+0.1	+1.7	0.0	-11.3	0.0	+0.2	+0.2	+0.1	+0.2	+0.4	+0.4
	TDA	+1.4	+0.8	+2.1	+0.2	+7.0	+0.3	-0.7	-0.2	+1.6	+0.4	+1.5	+2.1
	DMN	+0.6	+1.1	+2.2	+0.2	+6.8	+1.1	-7.4	-1.1	+2.1	-1.3	+1.3	+1.5
	Stat.$\mathcal{A}$	+3.8	+5.1	+2.4	+4.2	-0.1	+9.6	+3.0	+6.1	+6.9	+1.1	+2.3	+1.5

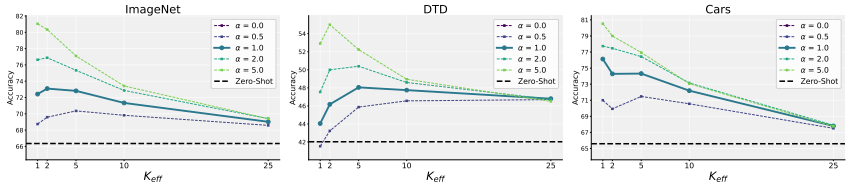
6.6 Ablation Studies

Gaussian anchor term weighting factor

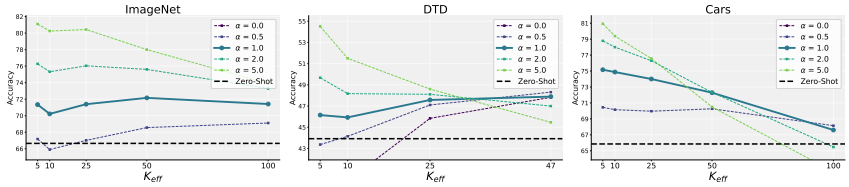
We study the effect of the Gaussian anchor term weighting factor in Figure 6.3. Our observations show that setting α to relatively large values can be highly beneficial when the ratio between the batch size and the number of effective classes is significantly greater than 1. For instance, Figure 6.3a demonstrates a +10% improvement on the ImageNet dataset when $K_{\text{eff}} \leq 5$ with a batch size of 64, and a +13% improvement when $5 \leq K_{\text{eff}} \leq 25$. In contrast, smaller α values can be more beneficial when the K_{eff} increases, as depicted for DTD and Cars in Figure 6.3b. However, over-reliance on tuning α would contradict our goal of maintaining a strong and versatile method across scenarios. While addressing this question lies beyond the scope of this paper, it presents an interesting avenue for future work.

β_k computation

We study the impact of our implementation choice of Section 6.3.2 in Table 6.3. Hard assignment proves to be more beneficial for large batches with



(a) Effect of α with a batch size of 64 and 1 to 25 effective classes on ImageNet, DTD, and Cars dataset.



(b) With a batch size of 1,000 and 5 to 100 effective classes on ImageNet, DTD, and Cars dataset.

Fig. 6.3 Ablation study on the impact of the anchor weighting α across various numbers of effective classes (K_{eff}). The line corresponding to $\alpha = 1$ (used in all our experiments) is highlighted with a wider stroke. Each reported performance is averaged over 1,000 tasks.

very few classes. This can be attributed to the small residuals of the soft assignment, as predictions for a given class are never exactly zero (see the softmax operator in Eq. (6.12)). Soft β_k yields greater improvement for smaller batches with relatively more classes. These insights suggest potential avenues for refining the anchor strategy, either by smoothing the small residuals of the prediction vectors or by better estimating the presence of a given class, in line with the observations made on the impact of the α weighting factor.

Runtime

Table 6.4 highlights the efficiency of Stat.$\mathcal{A}$, which is capable of processing 50,000 samples in just a few seconds. Note that CLIP inference encompasses both image features computation and text prompts encoding. Stat.$\mathcal{A}$ acts as a robust additional processing, seamlessly applied atop the model after inference to enhance predictions. Its computational cost is minimal, adding negligible overhead compared to the initial inference.

Table 6.3 Anchor strategies for various numbers of effective classes (K_{eff}) on the ImageNet dataset. Hard β_k refers to Eq. (6.10).

(a) Batch size of 1,000.

K_{eff}	5	10	25	50	100
Stat.$\mathcal{A}$ w/ soft β_k	70.3	69.2	70.7	**72.7**	**72.8**
Stat.$\mathcal{A}$ w/ hard β_k	**71.3**	**70.2**	**71.4**	72.2	71.4

(b) Batch size of 5,000.

K_{eff}	50	100	250	500	1,000
Stat.$\mathcal{A}$ w/ soft β_k	69.7	69.2	70.8	69.4	66.1
Stat.$\mathcal{A}$ w/ hard β_k	**70.8**	**70.4**	**71.1**	**69.5**	**66.5**

Table 6.4 Runtime on a single Nvidia RTX 3090 GPU with 24 GB of memory for the ImageNet dataset.

Batch size	CLIP inference	Stat.$\mathcal{A}$	Total runtime
1,000	6 s	0.1 s	6.1 s
50,000	5 min	15 s	$\sim$ 5 min

6.7 Conclusion of Chapter 6

The purpose of this work was twofold: first, to challenge the current conventions for benchmarking transductive and test-time adaptation methods; and second, to propose a step towards a more robust adaptation that provides consistent gains adhering to strict realistic constraints, such as the use of fixed hyperparameters across tasks. Our analysis revealed that while state-of-the-art methods excel in the specific scenarios of their original benchmarks, they fail in at least one of the more realistic scenarios introduced in this chapter. Consistent with the philosophy of previous chapters, we aimed to show that a simple, well-designed prior can yield a highly effective solution without incurring significant computational overhead. We introduced a novel parameter regularization within the TransCLIP objective, termed the Statistical Anchor (Stat.$\mathcal{A}$). This addition provides an interpretable mechanism that dynamically balances the data-driven statistics with the text-based priors. By making our codebase[2] publicly available, we

[2] https://github.com/MaxZanella/StatA

hope to encourage the community to move beyond simplified settings and adopt more diverse and rigorous benchmarking standards.

6.8 Appendix of Chapter 6

6.8.1 Kullback-Leibler divergence between two multivariate Gaussian distributions

Let $\mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and $\mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ two multivariate Gaussian distributions with respective probability density functions p and q. Then, we have

$$\begin{aligned}
\mathrm{KL}(p||q) &= \int_x p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x && (6.13)\\
&= \mathbb{E}_p[\log(p) - \log(q)] && (6.14)\\
&= \mathbb{E}_p[\frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_p|} - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_p)^\top \boldsymbol{\Sigma}_p^{-1}(\mathbf{x} - \boldsymbol{\mu}_p) \\
&+ \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_q)^\top \boldsymbol{\Sigma}_q^{-1}(\mathbf{x} - \boldsymbol{\mu}_q)] && (6.15)\\
&= \frac{1}{2}\mathbb{E}_p[\log \frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_p|}] - \frac{1}{2}\mathbb{E}_p[(\mathbf{x} - \boldsymbol{\mu}_p)^\top \boldsymbol{\Sigma}_p^{-1}(\mathbf{x} - \boldsymbol{\mu}_p)] \\
&+ \frac{1}{2}\mathbb{E}_p[(\mathbf{x} - \boldsymbol{\mu}_q)^\top \boldsymbol{\Sigma}_q^{-1}(\mathbf{x} - \boldsymbol{\mu}_q)] && (6.16)\\
&= \frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_p|} - \frac{1}{2}\mathbb{E}_p[(\mathbf{x} - \boldsymbol{\mu}_p)^\top \boldsymbol{\Sigma}_p^{-1}(\mathbf{x} - \boldsymbol{\mu}_p)] \\
&+ \frac{1}{2}\mathbb{E}_p[(\mathbf{x} - \boldsymbol{\mu}_q)^\top \boldsymbol{\Sigma}_q^{-1}(\mathbf{x} - \boldsymbol{\mu}_q)]. && (6.17)
\end{aligned}$$

We can rewrite the second term as

$$\begin{aligned}
(\mathbf{x} - \boldsymbol{\mu}_p)^\top \boldsymbol{\Sigma}_p^{-1}(\mathbf{x} - \boldsymbol{\mu}_p) &= \mathrm{Tr}\left((\mathbf{x} - \boldsymbol{\mu}_p)^\top \boldsymbol{\Sigma}_p^{-1}(\mathbf{x} - \boldsymbol{\mu}_p)\right) \\
&= \mathrm{Tr}\left((\mathbf{x} - \boldsymbol{\mu}_p)(\mathbf{x} - \boldsymbol{\mu}_p)^\top \boldsymbol{\Sigma}_p^{-1}\right) && (6.18)
\end{aligned}$$

by using

$$\mathrm{Tr}(ABC) = \mathrm{Tr}(BCA) = \mathrm{Tr}(CAB). \tag{6.19}$$

For the third term, since we assume $\mathbf{x}$ follows a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$, we have (see Matrix cookbook [180] Eq. 380 of Section 8.2)

$$\mathbb{E}[(\mathbf{x} - \boldsymbol{\mu}_q)^\top \boldsymbol{\Sigma}_q^{-1}(\mathbf{x} - \boldsymbol{\mu}_q)] = (\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^\top \boldsymbol{\Sigma}_q^{-1}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q) + \mathrm{Tr}(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p) \tag{6.20}$$

And therefore

$$\begin{aligned}\mathrm{KL}(p||q) &= \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_p|} - \frac{1}{2}\mathbb{E}_p[\mathrm{Tr}\left((\mathbf{x}-\boldsymbol{\mu}_p)(\mathbf{x}-\boldsymbol{\mu}_p)^\top\boldsymbol{\Sigma}_p^{-1}\right)] \\ &+ \frac{1}{2}((\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^\top\boldsymbol{\Sigma}_q^{-1}(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q) + \mathrm{Tr}\{\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p\}) \qquad (6.21)\\ &= \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_p|} - \frac{1}{2}(\mathrm{Tr}\left(\mathbb{E}_p[(\mathbf{x}-\boldsymbol{\mu}_p)(\mathbf{x}-\boldsymbol{\mu}_p)^\top]\boldsymbol{\Sigma}_p^{-1}\right) \\ &+ \frac{1}{2}((\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^\top\boldsymbol{\Sigma}_q^{-1}(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q) + \mathrm{Tr}(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p)) \qquad (6.22)\end{aligned}$$

by using the fact that trace and expectation can be interchanged. Moreover,

$$\mathbb{E}_p[(\mathbf{x}-\boldsymbol{\mu}_p)(\mathbf{x}-\boldsymbol{\mu}_p)^\top] = \boldsymbol{\Sigma}_p \qquad (6.23)$$

which simplifies further the second term of the sum and gives

$$\begin{aligned}\mathrm{KL}(p||q) &= \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_p|} - \frac{1}{2}(\mathrm{Tr}(\boldsymbol{I}_k) + \frac{1}{2}((\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^\top\boldsymbol{\Sigma}_q^{-1}(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q) \\ &+ \mathrm{Tr}(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p)) \qquad (6.24)\\ &= \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_p|} - \frac{d}{2} + \frac{1}{2}((\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^\top\boldsymbol{\Sigma}_q^{-1}(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q) + \mathrm{Tr}(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p)) \qquad (6.25)\\ &= \frac{1}{2}\left(\log\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_p|} - d + (\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^\top\boldsymbol{\Sigma}_q^{-1}(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q) + \mathrm{Tr}(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p)\right) \qquad (6.26)\end{aligned}$$

with d the number of dimensions.

6.8.2 Detailed derivations of the regularized updates of the parameters

We write p_k and q_k the respective probability density functions of the distributions $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ and $\mathcal{N}(\boldsymbol{\mu'}_k, \boldsymbol{\Sigma'}_k)$. With previous results, we have

$$\begin{aligned}\mathcal{A}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \sum_k \mathrm{KL}(q_k||p_k) = \frac{1}{2}\sum_k(\boldsymbol{\mu'}_k-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu'}_k-\boldsymbol{\mu}_k) \\ &+ \mathrm{Tr}(\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\Sigma'}_k) + \log\frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma'}_k|} - d. \qquad (6.27)\end{aligned}$$

With respect to μ_k

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \frac{\partial}{\partial \boldsymbol{\mu}_k}\Bigg(- \sum_{i \in \mathcal{Q}} \mathbf{z}_i^\top \log(\mathbf{p}_i) - \sum_{i \in \mathcal{Q}} \sum_{j \in \mathcal{Q}} w_{ij} \mathbf{z}_i^\top \mathbf{z}_j + \sum_{i \in \mathcal{Q}} \mathrm{KL}(\mathbf{z}_i || \hat{\mathbf{y}}_i) + \alpha . \mathcal{A}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \Bigg) \tag{6.28}$$

$$= \frac{\partial}{\partial \boldsymbol{\mu}_k}\Bigg(- \sum_{i \in \mathcal{Q}} \mathbf{z}_i^\top \log(\mathbf{p}_i) - \sum_{i \in \mathcal{Q}} \sum_{j \in \mathcal{Q}} w_{ij} \mathbf{z}_i^\top \mathbf{z}_j + \sum_{i \in \mathcal{Q}} \mathrm{KL}(\mathbf{z}_i || \hat{\mathbf{y}}_i) + \alpha \sum_{l=1}^{K} \mathrm{KL}(\mathbf{q}_l || \mathbf{p}_l) \Bigg) \tag{6.29}$$

$$= \frac{\partial}{\partial \boldsymbol{\mu}_k}\Bigg(- \sum_{i \in \mathcal{Q}} z_{i,k} \left(-\frac{1}{2} \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} (\mathbf{f}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{f}_i - \boldsymbol{\mu}_k) \right) + \frac{\alpha}{2} (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k) \Bigg) \tag{6.30}$$

$$= - \sum_{i \in \mathcal{Q}} z_{i,k} \left(\boldsymbol{\Sigma}_k^{-1} (\mathbf{f}_i - \boldsymbol{\mu}_k) \right) + \alpha \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k) \tag{6.31}$$

Observe that the term $\alpha \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)$ directly comes from the derivative of our statistical anchor $\mathcal{A}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with regard to $\boldsymbol{\mu}_k$. By setting the derivative to 0

$$- \sum_{i \in \mathcal{Q}} z_{i,k} (\mathbf{f}_i - \boldsymbol{\mu}_k) - \alpha (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k) = 0 \tag{6.32}$$

$$\sum_{i \in \mathcal{Q}} z_{i,k} \mathbf{f}_i + \alpha \boldsymbol{\mu'}_k = \sum_{i \in \mathcal{Q}} z_{i,k} \boldsymbol{\mu}_k + \alpha \boldsymbol{\mu}_k \tag{6.33}$$

$$\sum_{i \in \mathcal{Q}} z_{i,k} \mathbf{f}_i + \alpha \boldsymbol{\mu'}_k = \left(\sum_{i \in \mathcal{Q}} z_{i,k} + \alpha \right) \boldsymbol{\mu}_k \tag{6.34}$$

We then obtain the centroid update

$$\boldsymbol{\mu}_k = \frac{\sum_{i \in \mathcal{Q}} z_{i,k} \mathbf{f}_i + \alpha \boldsymbol{\mu'}_k}{\sum_{i \in \mathcal{Q}} z_{i,k} + \alpha}. \tag{6.35}$$

If we write

$$\beta_k = \frac{\sum_{i \in \mathcal{Q}} z_{i,k}}{\sum_{i \in \mathcal{Q}} z_{i,k} + \alpha} \tag{6.36}$$

and

$$\boldsymbol{v}_k = \frac{\sum_{i \in \mathcal{Q}} z_{i,k} \mathbf{f}_i}{\sum_{i \in \mathcal{Q}} z_{i,k}} \tag{6.37}$$

we get the new centroid update

$$\boldsymbol{\mu}_k = \beta_k \boldsymbol{v}_k + (1 - \beta_k) \boldsymbol{\mu'}_k \tag{6.38}$$

With respect to $\boldsymbol{\Sigma}_k^{-1}$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k^{-1}} = \frac{\partial}{\partial \boldsymbol{\Sigma}_k^{-1}} \Bigg(- \sum_{i \in \mathcal{Q}} \mathbf{z}_i^\top \log(\mathbf{p}_i) - \sum_{i \in \mathcal{Q}} \sum_{j \in \mathcal{Q}} w_{ij} \mathbf{z}_i^\top \mathbf{z}_j + \sum_{i \in \mathcal{Q}} \mathrm{KL}(\mathbf{z}_i || \hat{\mathbf{y}}_i) + \alpha \mathcal{A}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \Bigg) \tag{6.39}$$

$$= \frac{\partial}{\partial \boldsymbol{\Sigma}_k^{-1}} \Bigg(- \sum_{i \in \mathcal{Q}} \mathbf{z}_i^\top \log(\mathbf{p}_i) - \sum_{i \in \mathcal{Q}} \sum_{j \in \mathcal{Q}} w_{ij} \mathbf{z}_i^\top \mathbf{z}_j + \sum_{i \in \mathcal{Q}} \mathrm{KL}(\mathbf{z}_i || \hat{\mathbf{y}}_i) + \alpha \sum_{l=1}^{K} \mathrm{KL}(\mathbf{q}_l || \mathbf{p}_l) \Bigg) \tag{6.40}$$

$$= \frac{\partial}{\partial \boldsymbol{\Sigma}_k^{-1}} \Bigg(- \sum_{i \in \mathcal{Q}} z_{i,k} \left(-\frac{1}{2} \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} (\mathbf{f}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{f}_i - \boldsymbol{\mu}_k) \right) + \frac{\alpha}{2} \left(\log \frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma'}_k|} + (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k) + \mathrm{Tr}(\boldsymbol{\Sigma}_k^{-1} \boldsymbol{\Sigma'}_k) \right) \Bigg) \tag{6.41}$$

Note that the term $\log \frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma'}_k|} + (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k) + \mathrm{Tr}(\boldsymbol{\Sigma}_k^{-1} \boldsymbol{\Sigma'}_k)$ directly comes from the derivative of our statistical anchor $\mathcal{A}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with regard to $\boldsymbol{\Sigma}_k^{-1}$. Using the formulas (from Matrix cookbook [180])

$$\frac{\partial}{\partial X} (\log |X|) = (X^{-1})^\top \tag{6.42}$$

$$\frac{\partial}{\partial X^{-1}} (\log |X|) = -X^\top \tag{6.43}$$

and

$$\frac{\partial}{\partial X}(\mathrm{Tr}(AXB)) = A^\top B^\top \tag{6.44}$$

$$\frac{\partial}{\partial X}(\mathrm{Tr}(AX^{-1}B)) = -(X^{-1}BAX^{-1})^\top \tag{6.45}$$

as well as the fact that covariances are symmetric ($\mathbf{\Sigma}^\top = \mathbf{\Sigma}$), setting the derivative to 0 yields

$$\begin{aligned} &-\sum_{i\in\mathcal{Q}} z_{i,k}\left(\mathbf{\Sigma}_k - (\mathbf{f}_i - \boldsymbol{\mu}_k)(\mathbf{f}_i - \boldsymbol{\mu}_k)^\top\right) \\ &+\alpha(-\mathbf{\Sigma}_k^\top + (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)(\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)^\top + \mathbf{\Sigma'}_k^\top) = 0 \end{aligned} \tag{6.46}$$

$$\begin{aligned} &-\sum_{i\in\mathcal{Q}} z_{i,k}\left(\mathbf{\Sigma}_k - (\mathbf{f}_i - \boldsymbol{\mu}_k)(\mathbf{f}_i - \boldsymbol{\mu}_k)^\top\right) \\ &+\alpha(-\mathbf{\Sigma}_k + (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)(\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)^\top + \mathbf{\Sigma'}_k) = 0 \end{aligned} \tag{6.47}$$

$$\begin{aligned} \left(\sum_{i\in\mathcal{Q}} z_{i,k} + \alpha\right)\mathbf{\Sigma}_k = &\sum_{i\in\mathcal{Q}} z_{i,k}(\mathbf{f}_i - \boldsymbol{\mu}_k)(\mathbf{f}_i - \boldsymbol{\mu}_k)^\top \\ &+ \alpha(\mathbf{\Sigma'}_k + (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)(\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)^\top). \end{aligned} \tag{6.48}$$

We get

$$\mathbf{\Sigma}_k = \frac{\sum_{i\in\mathcal{Q}} z_{i,k}(\mathbf{f}_i - \boldsymbol{\mu}_k)(\mathbf{f}_i - \boldsymbol{\mu}_k)^\top + \alpha(\mathbf{\Sigma'}_k + (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)(\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)^\top)}{\sum_{i\in\mathcal{Q}} z_{i,k} + \alpha}. \tag{6.49}$$

By writing the old $\mathbf{\Sigma}_k$-update

$$\boldsymbol{T}_k = \frac{\sum_{i\in\mathcal{Q}} z_{i,k}(\mathbf{f}_i - \boldsymbol{\mu}_k)(\mathbf{f}_i - \boldsymbol{\mu}_k)^\top}{\sum_{i\in\mathcal{Q}} z_{i,k}}, \tag{6.50}$$

we obtain the new covariance update

$$\mathbf{\Sigma}_k = \beta_k \boldsymbol{T}_k + (1-\beta_k)(\mathbf{\Sigma'}_k + (\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)(\boldsymbol{\mu'}_k - \boldsymbol{\mu}_k)^\top). \tag{6.51}$$

6.8.3 Algorithm

Algorithm 1: Stat$\mathcal{A}$ procedure

Input: $\mathbf{f}_i, i = 1, \ldots, N$, $\mathbf{t}_k, k = 1, \ldots, K$, $\boldsymbol{\varnothing}$

$\mathbf{z}_i \leftarrow \hat{\mathbf{s}}_i \quad \forall i$; ▷ See Eq. (6.7)

$\boldsymbol{\mu}_k = \boldsymbol{\mu'}_k$; $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma'} \quad \forall k$; ▷ See Eq. (6.6)

while *not converged* **do**

  // Iterative decoupled updates

  for $l = 1 : \ldots$ **do**

    Update $\mathbf{z}_i^{(l+1)} \quad \forall i$; ▷ See Eq. (6.12)

  end

  // Closed-form updates

  Update $\beta_k \quad \forall k$; ▷ See Eq. (6.10)

  Update $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k \quad \forall k$; ▷ See Eq. (6.8)

end

return $\mathbf{z}$

7

Unifying View, Conclusion, and Perspectives

Preliminary Note:

Before starting the chapter, readers should be familiar with the methods and concepts introduced in Chapters 3 through 6. While previous chapters focused on proposing specific solutions for distinct problem settings (few-shot learning, test-time adaptation, and transduction), this chapter takes a step back to explore the connections between them. We aim to provide a unified perspective that links our contributions and the current state-of-the-art, highlighting common structures. This will serve as a discussion on the perspectives of the field and conclude the technical contributions of this thesis.

7.1 Introduction

In the previous chapters, we investigated various adaptation strategies for VLMs, ranging from inductive fine-tuning with limited supervision (Chapter 3) to training-free test-time adaptation on single samples (Chapter 4) and transductive inference on batches (Chapters 5 and 6). While these contributions addressed distinct experimental constraints (varying from the availability of a labeled support set to the statistical structure of the query data), they share a common objective: bridging the domain gap between pre-training and deployment by refining the model's alignment with the target task. This shared goal allows us to take a step back and examine the broader picture of this literature. In this somewhat more exploratory final chapter, we synthesize these distinct approaches under a unified framework. We argue that few-shot learning and test-time adaptation, often treated as separate paradigms in the literature and in this thesis, can be viewed as two instances of a broad adaptation problem. We formalize this perspective through a general objective function formulation, which allows us to categorize our contributions (namely CLIP-LoRA, MTA, TransCLIP, and Stat.$\mathcal{A}$) based on how they regularize the latent assignment variables or the model parameters. We also integrate popular adaptation techniques from state-of-the-art methods into this unified view. Finally, we conclude this thesis by summarizing our core findings and outlining future perspectives on inductive and transductive learning.

7.2 A Unified View of Few-Shot and Test-Time Adaptation

Few-shot learning (FSL) and test-time adaptation (TTA) are traditionally studied in separate communities, yet they address the same fundamental challenge: improving model generalization under changes in data conditions with minimal to no supervision. FSL focuses on generalizing to novel concepts using a small support set of labeled examples. In contrast, TTA aims to adapt a pre-trained model to unlabeled data drawn from a shifted distribution directly at inference time, without access to the source training data. TTA can thus be viewed as an extreme case of unsupervised domain adaptation, particularly relevant for foundation models where source data are often private or prohibitively large.

Despite their different origins, these fields have converged on similar algorithmic principles, particularly the use of unsupervised regularization techniques such as entropy minimization or pseudo-labeling. In practice, the distinction between these scenarios is often blurred. Real-world applications frequently involve varying degrees of supervision, ranging from fully unlabeled test batches to partially annotated streams. This continuity motivates the need for a flexible adaptation framework capable of interpolating between these extremes.

Furthermore, with the emergence of foundation models and large-scale pretraining, the traditional dichotomies on which few-shot learning and test-time adaptation were initially founded are being challenged. In this new context, the boundaries between base and novel classes, or between source and target domains, become less well-defined; not because distribution shifts disappear, but because the model already possesses broad, generic capabilities across many domains. As a result, the key challenge is no longer to transfer knowledge between clearly separated tasks, but rather to refine or specialize the model's general-purpose capabilities so they align more closely with the structure of the current data while relying on minimal supervision (as in few-shot learning) or incurring minimal computational overhead (as required for test-time applications).

Motivated by these observations, we provide a unified theoretical framework that encompasses the proposed methods of previous chapters under a general objective function. Furthermore, we integrate some key learning strategies of the FSL and TTA literature by matching them with the key terms of our formulation.

7.3 General Objective Function

7.3.1 Formulation

Structured objective formulation
Let $Z = \{\mathbf{z}_i\}_{i=1}^{N}$ be the set of soft assignments. The model is parameterized

by θ, which encapsulates the learnable weights[1]. In the previous chapters, we aimed to minimize an objective function of the general form:

$$\mathcal{J}(Z,\theta) = \underbrace{\mathcal{L}_{\text{fit}}(Z,\theta)}_{\text{compatibility}} + \underbrace{\mathcal{R}_{\text{E}}(Z)}_{\text{assignment reg.}} + \underbrace{\mathcal{R}_{\text{M}}(\theta)}_{\text{parameter reg.}}, \tag{7.1}$$

where $\mathcal{L}_{\text{fit}}$ measures the compatibility between the assignments and the model parameters. When the model defines likelihood scores such as the GMM used in TransCLIP, this term reduces to the standard log-likelihood weighted by Z:

$$\mathcal{L}_{\text{fit}}(Z,\theta) = -\sum_{i=1}^{N} \mathbf{z}_i^\top \log \mathbf{p}_i.$$

Here $\mathbf{p}_i = (p_{i,k})_{k=1}^K$ is the vector of likelihood scores. The regularizers $\mathcal{R}_E$ and $\mathcal{R}_M$ allow us to encode task priors on the assignments (e.g., entropy minimization) and the parameters (e.g., weight constraints), respectively.

7.3.2 Regularized EM algorithm

Since the joint optimization of Z and θ is typically non-convex, we iteratively minimize the objective with respect to one set of variables while keeping the other fixed. This alternating procedure guarantees that the objective value is non-increasing at each step, ensuring convergence to a local minimum (or a stationary point).

The algorithm alternates between the following two steps until convergence:

$$\textbf{(E-step)} \quad Z^{(l+1)} \leftarrow \arg\min_{Z} \left\{ \mathcal{L}_{\text{fit}}(Z,\theta^{(l)}) + \mathcal{R}_{\text{E}}(Z) \right\}, \tag{7.2}$$

$$\textbf{(M-step)} \quad \theta^{(l+1)} \leftarrow \arg\min_{\theta} \left\{ \mathcal{L}_{\text{fit}}(Z^{(l+1)},\theta) + \mathcal{R}_{\text{M}}(\theta) \right\}. \tag{7.3}$$

This formulation provides a modular framework where $\mathcal{R}_{\text{E}}$ and $\mathcal{R}_{\text{M}}$ can be tailored to implement a wide range of learning strategies commonly found in few-shot learning and test-time adaptation. This can take the form of *assignment regularization* (e.g., forcing consistencies through pairwise affinities as in TransCLIP) and *parameter regularization* (e.g., avoiding catastrophic forgetting with constraints on the weights as in Stat$\mathcal{A}$).

[1] In some cases, θ may contain both the original model parameters and additional ones, such as the centroids and covariance matrices in TransCLIP.

The case of the cross-entropy

It is instructive to analyze where the fully inductive few-shot learning paradigm fits within this framework. Consider the case where we have access to a labeled support set $\mathcal{S}$. The standard training objective is to minimize the cross-entropy loss between the model predictions and the ground-truth labels:

$$\mathcal{R}_{\mathrm{M}}(\theta) = -\frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{k=1}^{K} y_{i,k} \log s_{i,k}, \tag{7.4}$$

where $\mathbf{s}_i$ represents the posterior probability distribution $P(Y \mid X = \mathbf{x}_i; \theta)$. In this expression, the targets $y_{i,k}$ are fixed observations. Consequently, minimizing this term affects only the model parameters θ and does not involve the latent assignments Z. Therefore, the cross-entropy acts strictly as a parameter regularizer $\mathcal{R}_{\mathrm{M}}$: it constrains the optimization of θ to the manifold of solutions that are consistent with the labeled support data.

7.3.3 Interpretation through inductive and transductive learning

The presented framework provides a natural way to decompose learning into two complementary processes: estimating assignments and updating parameters.

E-step

The E-step can often be interpreted as the fully *transductive* component of the algorithm. It focuses on estimating the latent structure of the data; specifically, the assignment variables $\mathbf{z}_i$ given the current model state. In a purely inductive supervised setting (without assignment regularizer $\mathcal{R}_E$ and L_{fit}), this step effectively disappears. Conversely, in transductive settings, the E-step can be critical to infer soft labels for the unlabeled query set (e.g., maximizing confidence or enforcing consistency between similar pairs).

M-step

The M-step, on the other hand, often corresponds to the fully *inductive* learning phase. It updates the model parameters θ to better explain the estimated assignments/pseudo-labels (fixed or not). In typical inductive few-shot learning, the M-step adapts the model weights or prototypes to minimize $\mathcal{R}_M$ (often a cross-entropy loss), potentially constrained by other parameter regularization to prevent overfitting. Nevertheless, some transductive regularization can also be integrated into $\mathcal{R}_M$, for example by en-

forcing the output of the model to be consistent for similar query samples.

Fitting term
Finally, the fitting term $\mathcal{L}_{\text{fit}}$ serves as the bridge between these two processes. It links the estimated assignments from the E-step with the learnable parameters of the M-step, ensuring that the model's representations align with the inferred structure of the data. By selectively enabling or regularizing these steps, this framework allows us to recover a wide spectrum of methods from standard supervised training (fixed Z, update θ) to pure transductive inference (update Z, fixed θ) and fully adaptive approaches (update both, often alternating between generating pseudo-labels and updating some parameters). Under this interpretation, TransCLIP and Stat.$\mathcal{A}$ contain both a transductive and an inductive components that work in synergy.

7.4 Methods

7.4.1 Connection to Previous Chapters

By decomposing the learning objective into a data-fitting term ($\mathcal{L}_{\text{fit}}$), an assignment regularizer ($\mathcal{R}_{\text{E}}$), and a parameter regularizer ($\mathcal{R}_{\text{M}}$), we can categorize each method of the previous chapters based on which components it utilizes to constrain the optimization. Table 7.1 summarizes these relationships.

CLIP-LoRA as fine-tuning with pure parameter regularization
CLIP-LoRA (Chapter 3) operates in a strictly inductive few-shot setting. It does not infer latent assignments for the query set during training (no E-step). Instead, it optimizes the model parameters θ (specifically, some low-rank matrices) to minimize the cross-entropy loss on the support set. In our framework, this supervision acts entirely as a parameter regularizer $\mathcal{R}_{\text{M}}$, constraining the hypothesis space based on labeled data.

MTA as mode seeking with assignment regularization
MTA (Chapter 4) can also be fit into our R-EM framework. Unlike the other methods where the latent variables represent class probabilities, here $\mathbf{u}$ represents the *inlierness scores* of augmented views, and the parameter θ is the density mode $\mathbf{m}$ of the visual features. The fitting term $\mathcal{L}_{\text{fit}}$ is a

Table 7.1 Interpretation of the methods presented in previous chapters within the R-EM framework. Weighting coefficients are omitted for clarity.

Method	Fit term ($\mathcal{L}_{\text{fit}}$)	Assignment reg. ($\mathcal{R}_{\text{E}}$)	Parameter reg. ($\mathcal{R}_{\text{M}}$)
CLIP-LoRA	—	—	$\underbrace{-\frac{1}{\lvert\mathcal{S}\rvert}\sum_{i\in\mathcal{S}}\mathbf{y}_i^\top \log \mathbf{s}_i}_{\text{Cross-Entropy}}$
MTA	$\underbrace{-\sum_{p=1}^{N}\mathbf{u}_p K(\mathbf{f}_p-\mathbf{m})}_{\text{Kernel density est.}}$	$\underbrace{-\sum_{p,q} w_{p,q}u_p u_q}_{\text{Pairwise affinities}}\underbrace{-\sum_{p=1}^{N} u_p \ln u_p}_{\text{Entropy}}$	—
TransCLIP	$\underbrace{-\sum_{i\in\mathcal{Q}}\mathbf{z}_i^\top \log \mathbf{p}_i}_{\text{GMM log-likelihood}}$	$\underbrace{-\sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{D}} w_{ij}\mathbf{z}_i^\top\mathbf{z}_j}_{\text{Laplacian reg.}}+\underbrace{\sum_{i\in\mathcal{Q}}\mathbf{z}_i^\top \log\frac{\mathbf{z}_i}{\hat{\mathbf{s}}_i}}_{\text{KL divergence}}$	—
TransCLIP-FS	$\underbrace{-\sum_{i=1}^{N}\mathbf{z}_i^\top \log \mathbf{p}_i}_{\text{GMM log-likelihood}}$	$\underbrace{-\sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{D}} w_{ij}\mathbf{z}_i^\top\mathbf{z}_j}_{\text{Laplacian reg.}}+\underbrace{\sum_{i\in\mathcal{Q}}\mathbf{z}_i^\top \log\frac{\mathbf{z}_i}{\hat{\mathbf{s}}_i}}_{\text{KL divergence}}$	$\underbrace{-\sum_{i\in\mathcal{S}}\mathbf{y}_i^\top \log \mathbf{p}_i}_{\text{Supervision}}$
Stat$\mathcal{A}$	$\underbrace{-\sum_{i\in\mathcal{Q}}\mathbf{z}_i^\top \log \mathbf{p}_i}_{\text{GMM log-likelihood}}$	$\underbrace{-\sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{D}} w_{ij}\mathbf{z}_i^\top\mathbf{z}_j}_{\text{Laplacian reg.}}+\underbrace{\sum_{i\in\mathcal{Q}}\mathbf{z}_i^\top \log\frac{\mathbf{z}_i}{\hat{\mathbf{s}}_i}}_{\text{KL divergence}}$	$\underbrace{\sum_{k=1}^{K}\text{KL}\left(\mathcal{N}_k'\lvert\rvert\mathcal{N}_k\right)}_{\text{Statistical anchor}}$

robust Kernel Density Estimator that pulls the mode towards high-density regions weighted by $\mathbf{u}$. The assignment regularizer $\mathcal{R}_{\text{E}}$ enforces structural consistency by penalizing high scores for views that are dissimilar to others, so that the E-step can be interpreted as an outlier filtering. We can observe that there is no parameter regularization (e.g., enforcing the mode to stay close to its initialization).

TransCLIP as GMM fitting with assignment regularization

TransCLIP (Chapter 5) introduces transduction by modeling the query data distribution with a GMM with a $\mathcal{L}_{\text{fit}}$ term (GMM log-likelihood) and alternating between E-steps and M-steps. Crucially, TransCLIP imposes structure on the latent assignments Z via $\mathcal{R}_{\text{E}}$: it enforces smoothness (Laplacian regularization) and alignment with the zero-shot text predictions (KL divergence). It does not explicitly regularize the updated parameters μ and Σ (i.e., no $\mathcal{R}_{\text{M}}$), allowing them to drift freely to maximize the likelihood. This explains why TransCLIP is powerful in standard balanced scenarios but with failure modes when classes are missing or batches are small.

Stat$\mathcal{A}$ as GMM fitting with full regularization

Stat$\mathcal{A}$ (Chapter 6) addresses the robustness issues of TransCLIP in *realis-*

tic scenarios by activating the final component of the framework: $\mathcal{R}_{\mathrm{M}}$. It introduces a Statistical Anchor that penalizes deviations of the estimated parameters θ from a reliable prior derived from the text encoder. Stat$\mathcal{A}$ thus represents a complete instantiation of the R-EM framework, balancing data fitting ($\mathcal{L}_{\mathrm{fit}}$), assignment structure ($\mathcal{R}_{\mathrm{E}}$), and parameter regularization ($\mathcal{R}_{\mathrm{M}}$).

7.4.2 Connection to the Literature

We now extend our analysis to established methods from the broader adaptation literature. Table 7.2 in the Appendix (Section 7.7) encompasses both vision-only transductive few-shot methods, vision-only test-time adaptation, as well as previously presented adaptations for VLMs. By mapping their main components to our general objective function $\mathcal{J}(Z, \boldsymbol{\theta})$, we aim to provide a structured perspective on some key advancements of the literature. Note that not every method can be matched perfectly into our objective function (e.g., the hard filtering of TPT [47] or the SHOT that alternates optimizing centroids and backbone updates [139]).

7.5 Limitations and Future Directions

Towards realistic deployment

Throughout this thesis, our evaluation primarily relied on publicly available and curated benchmarks. While these datasets are essential for reproducibility and fair comparison with the state-of-the-art, they do not fully capture the complexity of real-world deployment. As discussed in Chapter 6 with Stat$\mathcal{A}$, real-world data streams often violate standard assumptions. Therefore, a critical trajectory for future research involves assessing these approaches under organic rather than artificially induced shifts. This includes handling heterogeneous data sources, signal degradation due to sensor failures, or highly noisy web-scraped data, among others.

Generative models

Recent years have witnessed the rise of generative models, a trend extending to VLMs with architectures such as LLaVA [181]. Generative models fundamentally rely on the structure of their latent spaces to process and generate meaningful tokens. We hypothesize that robust optimization al-

gorithms, such as the Gaussian Mixture Model fitting employed in TransCLIP or the mode-seeking mechanism in MTA, remain highly relevant. An important avenue for future research is therefore to extend our current algorithms beyond classification. Our methods may require specific tuning to accommodate these architectures. For instance, handling sequences of tokens instead of single embeddings per modality can be challenging due to the high dimensionality of the embedding space.

Model scaling vs. adaptation

Another critical open question concerns the relevance of adaptation methods in an era where foundation models are scaling rapidly in both parameter count and training data size. While scaling has been a primary driver of recent AI successes with new models beating previous ones by a large margin even when utilizing state-of-the-art adaptation techniques. It raises questions regarding the impact of research conducted under limited computational power that precludes training such models. One might argue that adaptation methods developed for current architectures risk obsolescence, as newer, larger models could outperform them by significant margins solely through raw capacity. However, the methods developed in this thesis rely on minimal assumptions regarding the underlying architecture. For instance, TransCLIP requires only a feature representation and an initial prediction guess. Consequently, as we demonstrated with EVA-CLIP-8B, a more powerful underlying representation naturally yields superior adaptation results.

Implications of extensive pre-training

Beyond model parameters, the scaling of data size presents both fundamental challenges and opportunities. As foundational models are trained on increasingly exhaustive datasets, potentially approaching internet-scale coverage, the distinction between training and out-of-distribution data begins to blur. One might argue that if a model has effectively "seen everything" during pre-training, the domain gaps that necessitate alignment may diminish, potentially reducing the utility of transductive adaptation. Future research must therefore determine where adaptation remains critical: likely not in standard scenarios covered by web-scale data, but in the *long tail* of rare events, specific sensor degradations, or highly specialized domains where even massive pre-training cannot guarantee coverage.

Test-time compute and transduction

Transductive approaches are uniquely positioned to manage vast amounts

of unlabeled test data, dynamically leveraging massive, uncurated streams to refine predictions. This resonates with the recent surge of interest in *test-time compute*, [182] where allocating computational resources at inference drives performance gains. For instance, transductive optimization could serve as a mechanism to trade inference latency for accuracy: by allowing the model to reason over the structure of the incoming batch, or even query vast external databases of unlabeled images, we can correct errors that an inductive forward pass could miss. This opens the door to optimization processes which are not limited to the current batch but are guided by retrieving relevant, structurally similar samples from a massive external corpus.

7.6 Final Thoughts

In this thesis, we went through a comprehensive landscape of adaptation strategies, addressing the critical challenges of learning with limited supervision. Our investigation has spanned a wide spectrum of limited data availability, from few-shot labeled scenarios to unsupervised settings involving either single-sample or batch inference.

Collectively, our work contributes to the field on three complementary levels: (i) a practical contribution: with **CLIP-LoRA**, we provided an efficient and accessible codebase for Low-Rank Adaptation, lowering the barrier to entry for fine-tuning contrastive VLMs; (ii) a theoretical contribution: through **MTA**, we developped a new more robust version of the MeanShift algorithm and proved its convergence; (iii) a methodological contribution: with **TransCLIP** and **Stat**$\mathcal{A}$, we pioneered the application of transductive inference in the VLM domain and introduced rigorous new evaluation protocols.

Beyond these specific algorithmic advances, we believe this body of work offers a broader contribution to the research community regarding scientific rigor and efficiency. First, we have consistently advocated for evaluation integrity. By questioning standard protocols, such as the reliance on oracle hyperparameter tuning on test sets, we highlighted the fragility of current state-of-the-art leaderboards. In addition, our work demonstrates that true progress requires benchmarking in "wild" scenar-

ios where data distributions are unknown and validation sets could be unavailable. Second, we have championed the philosophy of algorithmic efficiency over brute-force scaling. In an era dominated by massive compute, our findings with MTA, TransCLIP and Stat.$\mathcal{A}$ prove that well-designed, lightweight inference algorithms can achieve competitive performance without the need for prohibitive retraining.

Ultimately, we hope to have provided a modest but meaningful contribution to the field, encouraging future research towards more robust, transparent, and computationally efficient systems capable of navigating real-world constraints.

7.7 Appendix of Chapter 7

Table 7.2 Interpretation of (transductive) few-shot learning and test-time adaptation of the literature within our general objective function. Weighting coefficients are omitted for clarity. We match as close as possible the notations of each of the paper to the one used through this thesis but some conflicts might appear. $\bar{s}_k = \sum_{i \in \mathcal{Q}} s_{i,k} / |\mathcal{Q}|$

Method	Fit term ($\mathcal{L}_{\text{fit}}$)	Assignment reg. ($\mathcal{R}_{\text{E}}$)	Parameter reg. ($\mathcal{R}_{\text{M}}$)
CoOp [25]	—	—	$\underbrace{-\frac{1}{\lvert\mathcal{S}\rvert}\sum_{i\in\mathcal{S}}\mathbf{y}_i^\top \log \mathbf{s}_i}_{\text{Cross-Entropy}}$
Baseline [126]	—	—	$\underbrace{-\frac{1}{\lvert\mathcal{S}\rvert}\sum_{i\in\mathcal{S}}\mathbf{y}_i\top \log \mathbf{s}_i}_{\text{Cross-Entropy}} \underbrace{-\frac{1}{\lvert\mathcal{Q}\rvert}\sum_{i\in\mathcal{Q}}\mathbf{s}_i^\top \ln \mathbf{s}_i}_{\text{Cond. Entropy}}$
TIM [125]	—	—	$\underbrace{-\frac{1}{\lvert\mathcal{S}\rvert}\sum_{i\in\mathcal{S}}\mathbf{y}_i^\top \log \mathbf{s}_i}_{\text{Cross-Entropy}} \underbrace{\sum_{k=1}^{K}\bar{s}_k \log \bar{s}_k}_{\text{-Marginal Entropy}} \underbrace{-\sum_{i\in\mathcal{Q}}\mathbf{s}_i^\top \log \mathbf{s}_i}_{\text{Cond. Entropy}}$
LaplacianShot [130]	$\underbrace{\sum_{i\in\mathcal{Q}}\mathbf{z}_i^\top d(\mathbf{f}_i, \mathbf{m})}_{\text{Nearest Prototype}}$	$\underbrace{\sum_{i,j} w_{ij}\lVert\mathbf{z}_i - \mathbf{z}_j\rVert^2}_{\text{Laplacian Reg.}}$	—
PADDLE [129]	$\underbrace{\sum_{i\in\mathcal{Q}}\mathbf{z}_i^\top d(\mathbf{f}_i, \mathbf{m})}_{\text{Data fit}}$	$\underbrace{-\sum_{k=1}^{K}\bar{z}_k \log \bar{z}_k}_{\text{partition complexity}}$	$\underbrace{\sum_{i\in\mathcal{S}}\mathbf{y}_i^\top d(\mathbf{f}_i, \mathbf{m})}_{\text{Supervision}}$
Tent [138]	—	—	$\underbrace{-\frac{1}{\lvert\mathcal{Q}\rvert}\sum_{i\in\mathcal{Q}}\mathbf{s}_i^\top \ln \mathbf{s}_i}_{\text{Cond. Entropy}}$
SHOT [139]	$\underbrace{-\sum_{i\in\mathcal{Q}}\mathbf{z}_i^\top \log \mathbf{s}_i}_{\text{Pseudo-labeling (hard)}}$	—	$\underbrace{-\frac{1}{\lvert\mathcal{Q}\rvert}\sum_{i\in\mathcal{Q}}\mathbf{s}_i^\top \ln \mathbf{s}_i}_{\text{Cond. Entropy}} \underbrace{\sum_{k=1}^{K}\bar{s}_k \log \bar{s}_k}_{\text{-Marginal entropy}}$
T3A [173]	$\underbrace{-\sum_{i\in\mathcal{Q}}\mathbf{z}_i^\top \log \mathbf{s}_i}_{\text{Pseudo-labeling (hard)}}$	—	—
MEMO [103]	—	—	$\underbrace{-\sum_{k=1}^{K}\bar{\mathbf{s}}_k \log \bar{\mathbf{s}}_k}_{\text{Marginal entropy}}$
TPT [47]	—	—	$\underbrace{-\sum_{k=1}^{K}\bar{\mathbf{s}}_k \log \bar{\mathbf{s}}_k}_{\text{Marginal entropy}}$
ZLaP [170]	—	$\frac{\alpha}{2}\sum_{i,j=1}^{N} w_{ij}\left\lVert\frac{\mathbf{z}_i}{\sqrt{d_i}} - \frac{\mathbf{z}_j}{\sqrt{d_j}}\right\rVert^2$ $+(1-\alpha)\sum_{i=1}^{N}\lVert\mathbf{z}_i - \mathbf{y}_i\rVert_F^2$	—

8

Annex

Table 8.1 Prompt templates for each dataset.

(a) Prompt templates used in the experiments unless otherwise specified.

Dataset	Prompt template
ImageNet	"a photo of a []."
SUN397	"a photo of a []."
Aircraft	"a photo of a [], a type of aircraft.",
EuroSAT	"a centered satellite photo of [].",
Cars	"a photo of a [].",
Food101	"a photo of [], a type of food.",
Pets	"a photo of [], a type of pet.",
Flower102	"a photo of a [], a type of flower.",
Caltech101	"a photo of a [].",
DTD	"[] texture.",
UCF101	"a photo of a person doing [].",

(b) Custom prompt templates for ImageNet dataset [16].

"itap of a []."
"a bad photo of the []."
"a origami []."
"a photo of the large []."
"a [] in a video game."
"art of the []."
"a photo of the small []."

Table 8.2 The 80 handcrafted prompts used for majority vote.

"a photo of a [].", "a bad photo of a [].", "a photo of many [].", "a sculpture of a []., "a photo of the hard to see []., "a low resolution photo of the []., "a rendering of a []., "graffiti of a []., "a bad photo of the []., "a cropped photo of the [].",, "a tattoo of a []., "the embroidered []., "a photo of a hard to see []., "a bright photo of a []., "a photo of a clean []., "a photo of a dirty []., "a dark photo of the []., "a drawing of a []., "a photo of my []., "the plastic []., "a photo of the cool []., "a close-up photo of a []., "a black and white photo of the []., "a painting of the []., "a painting of a []., "a pixelated photo of the []., "a sculpture of the []., "a bright photo of the []., "a cropped photo of a []., "a plastic []., "a photo of the dirty []., "a jpeg corrupted photo of a []., "a blurry photo of the []., "a photo of the []., "a good photo of the []., "a rendering of the []., "a [] in a video game.", "a photo of one []., "a doodle of a []., "a close-up photo of the []., "the origami []., "the [] in a video game.", "a sketch of a []., "a doodle of the []., "a origami []., "a low resolution photo of a []., "the toy []., "a rendition of the []., "a photo of the clean []., "a photo of a large []., "a rendition of a []., "a photo of a nice []., "a photo of a weird []., "a blurry photo of a []., "a cartoon []., "art of a []., "a sketch of the []., "a embroidered []., "a pixelated photo of a []., "itap of the []., "a jpeg corrupted photo of the []., "a good photo of a []., "a plushie []., "a photo of the nice []., "a photo of the small []., "a photo of the weird []., "the cartoon []., "art of the []., "a drawing of the []., "a photo of the large []., "a black and white photo of a []., "the plushie []., "a dark photo of a []., "itap of a []., "graffiti of the []., "a toy []., "itap of my []., "a photo of a cool []., "a photo of a small []., "a tattoo of the []."

Table 8.3 Comprehensive overview of the 14 natural images datasets.

Dataset Properties	ImageNet [17]	ImageNet-A [117]	ImageNet-V2 [118]	ImageNet-R [111]	ImageNet-Sketch [119]	SUN397 [69]	Aircraft [70]	StanfordCars [67]	Food101 [71]	OxfordPets [72]	Flowers102 [73]	Caltech101 [74]	DTD [75]	UCF101 [76]
# Test Samples	50,000	7,500	10,000	30,000	50,889	19,850	3,333	8,041	30,300	3,669	2,463	2,465	1,692	3,783
# Classes	1,000	200	1,000	200	1,000	397	100	196	101	37	102	101	47	101

Table 8.4 Comprehensive overview of the 10 remote sensing scene classification datasets used in the few-shot benchmark.

Dataset Properties	AID [85]	EuroSAT [68]	MLRSNet [86]	OPTIMAL31 [87]	PatternNet [88]	RESISC45 [89]	RSC11 [90]	RSICB128 [91]	RSICB256 [91]	WHURS19 [92]
Total Samples	10,000	27,000	109,161	1,860	30,400	31,500	1,232	24,747	36,700	1,000
# Classes	30	10	46	31	38	45	11	45	35	19

Unrelated Works

I present peripheral contributions that still played an important role in my scientific journey. A key driver for these works was my active involvement in the TRAIL environment, which facilitated rich collaborations with researchers from other universities, for example during the TRAIL workshops (Paris, Berlin, Nantes and Lisbon).

- Benoît Gérin, Maxime Zanella, Maxence Wynen, Saïd Mahmoudi, Benoît Macq, Christophe De Vleeschouwer, *Exploring viability of Test-Time Training: Application to 3D segmentation in Multiple Sclerosis*, IEEE Conference on Artificial Intelligence (CAI) [13]

  Summary: Test-Time Training (TTT) is an unsupervised domain adaptation technique employing a self-supervised task performed by an attached branch model. While justification of key design choices are often neglected in the literature, we explore the viability of TTT in a real-case scenario and conduct an extensive evaluation of key hyperparameters in TTT, including the choice and the placement of the auxiliary task, the type of normalization layer, and the model parameters to adapt. We carry out this study in Multiple Sclerosis (MS) diagnosis relying on focal lesions visible in conventional MRI. Manual lesion segmentation based on automated methods face significant challenges in clinical integration due to MRI domain shifts, i.e. discrepancies between training and deployment data, notably due to different acquisition settings. Finally, we propose general guidelines to apply TTT in practice.

- Sébastien Piérard, Anthony Cioppa, Anaïs Halin, Renaud Vandeghen, Maxime Zanella, Benoît Macq and , Saïd Mahmoudi, Marc Van Droo-

genbroeck, *Mixture Domain Adaptation To Improve Semantic Segmentation in Real-World Surveillance*, Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops [14]

Summary: Various tasks encountered in real-world surveillance can be addressed by determining posteriors (e.g. by Bayesian inference or machine learning), based on which critical decisions must be taken. However, the surveillance domain (acquisition device, operating conditions, etc.) is often unknown, which prevents any possibility of scene-specific optimization. In this work, we define a probabilistic framework and present a formal proof of an algorithm for the unsupervised many-to-infinity domain adaptation of posteriors. Our proposed algorithm is applicable when the probability measure associated with the target domain is a convex combination of the probability measures of the source domains. It makes use of source models and a domain discriminator model trained off-line to compute posteriors adapted on the fly to the target domain. Finally, we show the effectiveness of our algorithm for the task of semantic segmentation in real-world surveillance.

- Anaïs Halin, Sébastien Piérard, , Renaud Vandeghen , BenoîtGérin, Maxime Zanella, Martin Colot, Jan Held,Anthony Cioppa, Emmanuel Jean, Gianluca Bontempi, Saïd Mahmoudi, Benoît Macq, Marc, Van Droogenbroeck, *Physically Interpretable Probabilistic Domain Characterization*, Proceedings of the Asian Conference on Computer Vision (ACCV) Workshops [15]

  Summary: Characterizing domains is essential for models analyzing dynamic environments, as it allows them to adapt to evolving conditions or to hand the task over to backup systems when facing conditions outside their operational domain. Existing solutions typically characterize a domain by solving a regression or classification problem, which limits their applicability as they only provide a limited summarized description of the domain. In this work, we present a novel approach to domain characterization by characterizing domains as probability distributions. Particularly, we develop a method to predict the likelihood of different weather conditions from images captured by vehicle-mounted cameras by estimating distributions of physical parameters using normalizing flows.

Bibliography

[1] Maxime Zanella and Ismail Ben Ayed. Low-rank few-shot adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1593–1603, June 2024.

[2] Manon Dausort, Tiffanie Godelaine, Maxime Zanella, Karim El Khoury, Isabelle Salmon, and Benoît Macq. Exploring foundation models fine-tuning for cytology classification. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1–5, 2025.

[3] Karim El Khoury, Maxime Zanella, Christophe De Vleeschouwer, and Benoit Macq. Few-shot adaptation benchmark for remote sensing vision-language models. *arXiv preprint arXiv:2510.07135*, 2025.

[4] Maxime Zanella and Ismail Ben Ayed. On the test-time zero-shot generalization of vision-language models: Do we really need prompt learning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23783–23793, June 2024.

[5] Maxime Zanella, Benoît Gérin, and Ismail Ben Ayed. Boosting vision-language models with transduction. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 62223–62256. Curran Associates, Inc., 2024.

[6] Maxime Zanella, Fereshteh Shakeri, Yunshi Huang, Houda Bahig, and Ismail Ben Ayed. Boosting vision-language models for histopathology classification: Predict all at once. In Zhongying Deng, Yiqing Shen, Hyunwoo J. Kim, Won-Ki Jeong, Angelica I.

Aviles-Rivero, Junjun He, and Shaoting Zhang, editors, *Foundation Models for General Medical AI*, pages 153–162, Cham, 2025. Springer Nature Switzerland.

[7] Karim El Khoury, Maxime Zanella, Benoît Gérin, Tiffanie Godelaine, Benoît Macq, Saïd Mahmoudi, Christophe De Vleeschouwer, and Ismail Ben Ayed. Enhancing remote sensing vision-language models for zero-shot scene classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2025.

[8] Maxime Zanella, Clément Fuchs, Christophe De Vleeschouwer, and Ismail Ben Ayed. Realistic test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25103–25112, June 2025.

[9] Maxime Zanella, Clément Fuchs, Ismail Ben Ayed, and Christophe De Vleeschouwer. Vocabulary-free few-shot learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 149–158, June 2025.

[10] Ghassen Baklouti, Maxime Zanella, and Ismail Ben Ayed. Language-aware information maximization for transductive few-shot clip. *arXiv preprint arXiv:2509.00305*, 2025.

[11] Fereshteh Shakeri, Ghassen Baklouti, Julio Silva-Rodríguez, Maxime Zanella, Houda Bahig, Jose Dolz, and Ismail Ben Ayed. Test-time adaptation of medical vision-language models. In Won-Ki Jeong, Hyunwoo J. Kim, Zhongying Deng, Yiqing Shen, Angelica I. Aviles-Rivero, and Shaoting Zhang, editors, *Foundation Models for General Medical AI*, pages 181–191, Cham, 2026. Springer Nature Switzerland.

[12] Clément Fuchs, Maxime Zanella, and Christophe De Vleeschouwer. Online gaussian test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 128–137, June 2025.

[13] Benoît Gérin, Maxime Zanella, Maxence Wynen, Saïd Mahmoudi, Benoît Macq, and Christophe De Vleeschouwer. Exploring viability of test-time training: Application to 3d segmentation in multiple

sclerosis. In *IEEE Conference on Artificial Intelligence (CAI)*, pages 557–562, 2024.

[14] Sébastien Piérard, Anthony Cioppa, Anaïs Halin, Renaud Vandeghen, Maxime Zanella, Benoît Macq, Saïd Mahmoudi, and Marc Van Droogenbroeck. Mixture domain adaptation to improve semantic segmentation in real-world surveillance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 22–31, January 2023.

[15] Anaïs Halin, Sébastien Piérard, Renaud Vandeghen, Benoît Gérin, Maxime Zanella, Martin Colot, Jan Held, Anthony Cioppa, Emmanuel Jean, Gianluca Bontempi, Saïd Mahmoudi, Benoît Macq, and Marc Van Droogenbroeck. Physically interpretable probabilistic domain characterization. In *Proceedings of the Asian Conference on Computer Vision (ACCV) Workshops*, pages 15–33, December 2024.

[16] 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 (ICML)*, pages 8748–8763. PmLR, 2021.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[20] 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 (ICLR)*, 2021.

[21] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, pages 4904–4916. PMLR, 2021.

[22] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021.

[23] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18123–18133, 2022.

[24] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10965–10975, 2022.

[25] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022.

[26] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15659–15669, 2023.

[27] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision (ECCV)*, pages 493–510. Springer, 2022.

[28] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné,

Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

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

[30] Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023.

[31] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the Association for Computational Linguistics (ACL) (Volume 2: Short Papers)*, pages 1–9, 2022.

[32] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[33] 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 (ICML)*, pages 2790–2799. PMLR, 2019.

[34] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 1022–1035. Curran Associates, Inc., 2021.

[35] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:16664–16678, 2022.

[36] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:109–123, 2022.

[37] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3045–3059, 2021.

[38] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the Association for Computational Linguistics (ACL) and the International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[39] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, pages 709–727. Springer, 2022.

[40] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[41] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5206–5215, 2022.

[42] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor Guilherme Turrisi da Costa, Cees GM Snoek, Georgios Tzimiropoulos, and Brais Martinez. Variational prompt tuning improves generalization of vision-language models. *arXiv preprint arXiv:2210.02390*, 2022.

[43] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6757–6767, 2023.

[44] Adrian Bulat and Georgios Tzimiropoulos. Lasp: Text-to-text optimization for language-aware soft prompting of vision & language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23232–23241, June 2023.

[45] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Plot: Prompt learning with optimal transport for vision-language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[46] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022.

[47] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:14274–14289, 2022.

[48] Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2704–2714, 2023.

[49] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, pages 1–15, 2023.

[50] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10899–10909, 2023.

[51] Julio Silva-Rodriguez, Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. A closer look at the few-shot adaptation of large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23681–23690, 2024.

[52] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of

large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[53] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023.

[54] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv preprint arXiv:2210.07558*, 2022.

[55] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2022.

[56] Sanghyeon Kim, Hyunmo Yang, Younghyun Kim, Youngjoon Hong, and Eunbyung Park. Hydra: Multi-head low-rank adaptation for parameter efficient fine-tuning. *arXiv preprint arXiv:2309.06922*, 2023.

[57] Bojia Zi, Xianbiao Qi, Lingzhi Wang, Jianan Wang, Kam-Fai Wong, and Lei Zhang. Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices. *arXiv preprint arXiv:2309.02411*, 2023.

[58] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19113–19122, 2023.

[59] Yassine Ouali, Adrian Bulat, Brais Matinez, and Georgios Tzimiropoulos. Black box few-shot adaptation for vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15534–15546, 2023.

[60] Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896, 2021.

[61] Matteo Farina, Massimiliano Mancini, Giovanni Iacca, and Elisa Ricci. Rethinking few-shot adaptation of vision-language models in

two stages. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29989–29998, June 2025.

[62] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*, 2018.

[63] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the Association for Computational Linguistics (ACL) and the International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, 2021.

[64] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations (ICLR)*, 2021.

[65] Arnav Chavan, Zhuang Liu, Deepak Gupta, Eric Xing, and Zhiqiang Shen. One-for-all: Generalized lora for parameter-efficient fine-tuning. *arXiv preprint arXiv:2306.07967*, 2023.

[66] Hossein Rajabzadeh, Mojtaba Valipour, Tianshu Zhu, Marzieh Tahaei, Hyock Ju Kwon, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. Qdylora: Quantized dynamic low-rank adaptation for efficient large language model tuning. *arXiv preprint arXiv:2402.10462*, 2024.

[67] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 554–561, 2013.

[68] 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*, 12(7):2217–2226, 2019.

[69] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE Computer Society Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 3485–3492. IEEE, 2010.

[70] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[71] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, pages 446–461. Springer, 2014.

[72] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3498–3505. IEEE, 2012.

[73] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, pages 722–729. IEEE, 2008.

[74] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 178–178. IEEE, 2004.

[75] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, 2014.

[76] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[77] Qiangqiang Yuan, Huanfeng Shen, Tongwen Li, Zhiwei Li, Shuwen Li, Yun Jiang, Hongzhang Xu, Weiwei Tan, Qianqian Yang, Jiwen Wang, et al. Deep learning in environmental remote sensing: Achievements and challenges. *Remote sensing of Environment*, 241:111716, 2020.

[78] Swee King Phang, Tsai Hou Adam Chiang, Ari Happonen, and Miko May Lee Chang. From satellite to uav-based remote sensing: A review on precision agriculture. *Ieee Access*, 11:127057–127076, 2023.

[79] Karim El Khoury, Tiffanie Godelaine, Simon Delvaux, Sébastien Lugan, and Benoît Macq. Streamlined hybrid annotation framework using scalable codestream for bandwidth-restricted uav object detection. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 1581–1587, 2024.

[80] Gong Cheng, Xingxing Xie, Junwei Han, Lei Guo, and Gui-Song Xia. Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:3735–3756, 2020.

[81] Wang Miao, Jie Geng, and Wen Jiang. Semi-supervised remote-sensing image scene classification using representation consistency siamese network. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022.

[82] Eftychios Protopapadakis, Anastasios Doulamis, Nikolaos Doulamis, and Evangelos Maltezos. Stacked autoencoders driven by semi-supervised learning for building extraction from near infrared remote sensing imagery. *Remote Sensing*, 13(3):371, 2021.

[83] Hao Chen, Zhenghong Li, Jiangjiang Wu, Wei Xiong, and Chun Du. Semiroadexnet: A semi-supervised network for road extraction from remote sensing imagery via adversarial learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 198:169–183, 2023.

[84] Chunping Qiu, Xiaoyu Zhang, Xiaochong Tong, Naiyang Guan, Xiaodong Yi, Ke Yang, Junjie Zhu, and Anzhu Yu. Few-shot remote sensing image scene classification: Recent advances, new baselines, and future trends. *ISPRS Journal of Photogrammetry and Remote Sensing*, 209:368–382, 2024.

[85] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3965–3981, 2017.

[86] Xiaoman Qi, Panpan Zhu, Yuebin Wang, Liqiang Zhang, Junhuan Peng, Mengfan Wu, Jialong Chen, Xudong Zhao, Ning Zang, and P Takis Mathiopoulos. Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:337–350, 2020.

[87] Qi Wang, Shaoteng Liu, Jocelyn Chanussot, and Xuelong Li. Scene classification with recurrent attention of vhr remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):1155–1167, 2018.

[88] Weixun Zhou, Shawn Newsam, Congmin Li, and Zhenfeng Shao. Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval. *ISPRS journal of photogrammetry and remote sensing*, 145:197–209, 2018.

[89] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[90] Lijun Zhao, Ping Tang, and Lianzhi Huo. Feature significance-based multibag-of-visual-words model for remote sensing image scene classification. *Journal of Applied Remote Sensing*, 10(3):035004–035004, 2016.

[91] Haifeng Li, Xin Dou, Chao Tao, Zhixiang Wu, Jie Chen, Jian Peng, Min Deng, and Ling Zhao. Rsi-cb: A large-scale remote sensing image classification benchmark using crowdsourced data. *Sensors*, 20(6):1594, 2020.

[92] Gui-Song Xia, Wen Yang, Julie Delon, Yann Gousseau, Hong Sun, and Henri Maître. Structural high-resolution satellite image indexing. In *ISPRS TC VII Symposium-100 Years ISPRS*, volume 38, pages 298–303, 2010.

[93] Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu, and Jun Zhou. Remoteclip: A vision language foundation model for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[94] Zilun Zhang, Tiancheng Zhao, Yulong Guo, and Jianwei Yin. Rs5m and georsclip: A large scale vision-language dataset and a large

vision-language model for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[95] Zhecheng Wang, Rajanie Prabha, Tianyuan Huang, Jiajun Wu, and Ram Rajagopal. Skyscript: A large and semantically diverse vision-language dataset for remote sensing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5805–5813, 2024.

[96] Alan L Yuille and Anand Rangarajan. The concave-convex procedure (cccp). *Advances in Neural Information Processing Systems (NeurIPS)*, 14, 2001.

[97] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30, 2021.

[98] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 951–958. IEEE, 2009.

[99] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15691–15701, 2023.

[100] Zachary Novack, Julian McAuley, Zachary Chase Lipton, and Saurabh Garg. Chils: Zero-shot image classification with hierarchical label sets. In *International Conference on Machine Learning (ICML)*, pages 26342–26362. PMLR, 2023.

[101] Yunhao Ge, Jie Ren, Andrew Gallagher, Yuxiao Wang, Ming-Hsuan Yang, Hartwig Adam, Laurent Itti, Balaji Lakshminarayanan, and Jiaping Zhao. Improving zero-shot generalization and robustness of multi-modal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11093–11101, 2023.

[102] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024.

[103] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 38629–38642. Curran Associates, Inc., 2022.

[104] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

[105] Irene Solaiman. The gradient of generative ai release: Methods and considerations. In *Proceedings of the 2023 ACM conference on fairness, accountability, and transparency*, pages 111–122, 2023.

[106] Pierre Colombo, Victor Pellegrain, Malik Boudiaf, Victor Storchan, Myriam Tami, Ismail Ben Ayed, Celine Hudelot, and Pablo Piantanida. Transductive learning for textual few-shot classification in api-based embedding models. *arXiv preprint arXiv:2310.13998*, 2023.

[107] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.

[108] OpenAI. Openai. gpt-4v(ision) system card., 2023.

[109] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[110] Dan Hendrycks*, Norman Mu*, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations (ICLR)*, 2020.

[111] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF*

International Conference on Computer Vision (ICCV), pages 8340–8349, 2021.

[112] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[113] Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1197–1203. IEEE, 1999.

[114] Miguel Á. Carreira-Perpiñán. Gaussian mean-shift is an em algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:767–776, 2007.

[115] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. The variable bandwidth mean shift and data-driven scale selection. In *Proceedings Eighth IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 438–445. IEEE, 2001.

[116] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[117] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271, 2021.

[118] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, pages 5389–5400. PMLR, 2019.

[119] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[120] Zhiqiu Lin, Samuel Yu, Zhiyi Kuang, Deepak Pathak, and Deva Ramanan. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19325–19337, 2023.

[121] Vladimir N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

[122] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning (ICML)*, pages 200–209, 1999.

[123] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in Neural Information Processing Systems (NeurIPS)*, 16, 2003.

[124] Omer Belhasin, Guy Bar-Shalom, and Ran El-Yaniv. Transboost: Improving the best imagenet performance using deep transduction. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:28363–28373, 2022.

[125] Malik Boudiaf, Imtiaz Ziko, Jérôme Rony, José Dolz, Pablo Piantanida, and Ismail Ben Ayed. Information maximization for few-shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:2445–2457, 2020.

[126] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations (ICLR)*, 2019.

[127] Michalis Lazarou, Tania Stathaki, and Yannis Avrithis. Iterative label cleaning for transductive and semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8751–8760, 2021.

[128] Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 741–756. Springer, 2020.

[129] Ségolène Martin, Malik Boudiaf, Emilie Chouzenoux, Jean-Christophe Pesquet, and Ismail Ayed. Towards practical few-shot

query sets: Transductive minimum description length inference. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:34677–34688, 2022.

[130] Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian regularized few-shot learning. In *International Conference on Machine Learning (ICML)*, pages 11660–11670. PMLR, 2020.

[131] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Leveraging the feature distribution in transfer-based few-shot learning. In *International Conference on Artificial Neural Networks*, pages 487–499, 2021.

[132] Hao Zhu and Piotr Koniusz. Ease: Unsupervised discriminant subspace learning for transductive few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9078–9088, June 2022.

[133] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Swapprompt: Test-time prompt adaptation for vision-language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[134] Jameel Abdul Samadh, Mohammad Hanan Gani, Noor Hussein, Muhammad Uzair Khattak, Muhammad Muzammal Naseer, Fahad Shahbaz Khan, and Salman H Khan. Align your prompts: Test-time prompting with distribution alignment for zero-shot generalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.

[135] Eulrang Cho, Jooyeon Kim, and Hyunwoo J Kim. Distribution-aware prompt tuning for vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22004–22013, 2023.

[136] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15190–15200, 2023.

[137] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

[138] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

[139] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 6028–6039. PMLR, 2020.

[140] Y Liu, J Lee, M Park, S Kim, E Yang, SJ Hwang, and Y Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[141] Olivier Veilleux, Malik Boudiaf, Pablo Piantanida, and Ismail Ben Ayed. Realistic evaluation of transductive few-shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:9290–9302, 2021.

[142] Ségolène Martin, Yunshi Huang, Fereshteh Shakeri, Jean-Christophe Pesquet, and Ismail Ben Ayed. Transductive zero-shot and few-shot clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[143] Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. Eva-clip-18b: Scaling clip to 18 billion parameters, 2024.

[144] Zhengbo Wang, Jian Liang, Lijun Sheng, Ran He, Zilei Wang, and Tieniu Tan. A hard-to-beat baseline for training-free clip-based adaptation. In *International Conference on Learning Representations (ICLR)*, 2023.

[145] Christopher M Bishop. Pattern recognition and machine learning. *Springer*, 2:5–43, 2006.

[146] Michael Kearns, Yishay Mansour, and Andrew Y Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. *Learning in graphical models*, pages 495–520, 1998.

[147] Meng Tang, Dmitrii Marin, Ismail Ben Ayed, and Yuri Boykov. Kernel cuts: Kernel and spectral clustering meet regularization. *International Journal of Computer Vision*, 127:477–511, 2019.

[148] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010.

[149] Kenneth Lange, David R Hunter, and Ilsoon Yang. Optimization transfer using surrogate objective functions. *Journal of computational and graphical statistics*, 9(1):1–20, 2000.

[150] Mingyi Hong, Xiangfeng Wang, Meisam Razaviyayn, and Zhi-Quan Luo. Iteration complexity analysis of block coordinate descent methods. *Mathematical Programming*, 163:85–114, 2017.

[151] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning (ICML)*, pages 513–521. PMLR, 2013.

[152] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[153] Liron Pantanowitz. Digital images and the future of digital pathology. *Journal of pathology informatics*, 1, 2010.

[154] Anant Madabhushi. Digital pathology image analysis: opportunities and challenges. *Imaging in medicine*, 1(1):7, 2009.

[155] Daisuke Komura and Shumpei Ishikawa. Machine learning methods for histopathological image analysis. *Computational and structural biotechnology journal*, 16:34–42, 2018.

[156] Sokol Petushi, Fernando U Garcia, Marian M Haber, Constantine Katsinis, and Aydin Tozeren. Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer. *BMC medical imaging*, 6(1):1–11, 2006.

[157] Hammad Qureshi, Olcay Sertel, Nasir Rajpoot, Roland Wilson, and Metin Gurcan. Adaptive discriminant wavelet packet transform and local binary patterns for meningioma subtype classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 196–204. Springer, 2008.

[158] Ali Tabesh, Mikhail Teverovskiy, Ho-Yuen Pang, Vinay P Kumar, David Verbel, Angeliki Kotsianti, and Olivier Saidi. Multifeature prostate cancer diagnosis and gleason grading of histological images. *IEEE transactions on medical imaging*, 26(10):1366–1378, 2007.

[159] Cagatay Bilgin, Cigdem Demir, Chandandeep Nagi, and Bulent Yener. Cell-graph mining for breast tissue modeling and classification. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5311–5314. IEEE, 2007.

[160] Iryna Hartsock and Ghulam Rasool. Vision-language models for medical report generation and visual question answering: A review. *CoRR*, abs/2403.02469, 2024.

[161] Xuxin Chen, Ximin Wang, Ke Zhang, Kar-Ming Fung, Theresa C Thai, Kathleen Moore, Robert S Mannel, Hong Liu, Bin Zheng, and Yuchen Qiu. Recent advances and clinical applications of deep learning in medical image analysis. *Medical Image Analysis*, 79, 2022.

[162] Zhi Huang, Federico Bianchi, et al. A visual–language foundation model for pathology image analysis using medical twitter. *Nature medicine*, 29(9):2307–2316, 2023.

[163] Wisdom Ikezogwo, Saygin Seyfioglu, et al. Quilt-1m: One million image-text pairs for histopathology. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.

[164] Ming Y Lu, Bowen Chen, et al. A visual-language foundation model for computational pathology. *Nature Medicine*, 30(3):863–874, 2024.

[165] Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue. *Zenodo*, 5281, 2018.

[166] Julio Silva-Rodríguez, Arne Schmidt, María A Sales, Rafael Molina, and Valery Naranjo. Proportion constrained weakly supervised histopathology image classification. *Computers in Biology and Medicine*, 147:105714, 2022.

[167] Katharina Kriegsmann, Frithjof Lobers, Christiane Zgorzelski, Joerg Kriegsmann, Charlotte Janssen, Rolf Ruedinger Meliss, Thomas Muley, Ulrich Sack, Georg Steinbuss, and Mark Kriegsmann. Deep

learning for the detection of anatomical tissue structures and neoplasms of the skin on scanned histopathological tissue sections. *Frontiers in Oncology*, 12:1022967, 2022.

[168] Andrew A Borkowski, Marilyn M Bui, L Brannon Thomas, Catherine P Wilson, Lauren A DeLand, and Stephen M Mastorides. Lung and colon cancer histopathological image dataset (lc25000). *arXiv preprint arXiv:1912.12142*, 2019.

[169] Matteo Farina, Gianni Franchi, Giovanni Iacca, Massimiliano Mancini, and Elisa Ricci. Frustratingly easy test-time adaptation of vision-language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:129062–129093, 2024.

[170] Vladan Stojnić, Yannis Kalantidis, and Giorgos Tolias. Label propagation for zero-shot classification with vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[171] Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14162–14171, June 2024.

[172] Yabin Zhang, Wenjie Zhu, Hui Tang, Zhiyuan Ma, Kaiyang Zhou, and Lei Zhang. Dual memory networks: A versatile adaptation approach for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28718–28728, June 2024.

[173] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 2427–2440. Curran Associates, Inc., 2021.

[174] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 295–305, 2022.

[175] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:27253–27266, 2022.

[176] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8344–8353, 2022.

[177] Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15922–15932, 2023.

[178] Y. Boykov, H. N. Isack, C. Olsson, and I. Ben Ayed. Volumetric bias in segmentation and reconstruction: Secrets and solutions. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[179] Rie Kubota Ando and Tong Zhang. Learning on Graph with Laplacian Regularization. In Bernhard Schölkopf, John Platt, and Thomas Hofmann, editors, *Advances in Neural Information Processing Systems 19*, pages 25–32. The MIT Press, September 2007.

[180] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[181] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023.

[182] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.