TIM++: Transductive Information Maximization for Few-Shot CLIP

Yingping Li[1], Yutong Zou[1], Yunshi Huang[2], Changzhe Jiao[1], Xinlin Wang[1], Shen Peng[3], Zhang Guo[1*], Shuiping Gou[1*]

[1]Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China
[2] Shanghai Academy of Artificial Intelligence for Science, Shanghai 200003, China
[3] School of Mathematics and Statistics, Xidian University, Xi'an 710071, China
{guozhang@xidian.edu.cn, shpgou@mail.xidian.edu.cn}

Abstract

Transductive Information Maximization (TIM) is a leading transductive few-shot learning method that maximizes the mutual information between query features and their predicted labels, while incorporating supervision from the support set. However, its potential remains underexplored, primarily due to the limited utilization of textual knowledge provided by vision-language models (VLMs) such as CLIP. To address this, we propose TIM++, an enhanced framework that incorporates both visual and textual information for few-shot CLIP adaptation. Specifically, TIM++ introduces a Kullback-Leibler (KL) divergence-based regularization term that encourages the model's posterior predictions to align with CLIP's zero-shot output distribution, especially focusing on the most confident predictions. Additionally, we develop an improved prototype initialization strategy that leverages both support and query features enriched with CLIP-guided semantics. Extensive experiments on 11 public datasets demonstrate that TIM++ consistently outperforms the standard TIM, achieving average accuracy gains of 19.25% and 10.88% in 1-shot and 2-shot settings, respectively. TIM++ also surpasses other existing state-of-the-art methods, establishing a new benchmark for few-shot learning with VLMs.

Code — https://github.com/Yingping-LI/TIM-plus2

Introduction

In recent years, pre-trained vision-language models (VLMs) have emerged rapidly as powerful tools for various downstream tasks, such as image classification, object detection and segmentation, and visual question answering. These models are trained on large-scale image-text pairs to learn a shared embedding space that aligns visual and textual modalities, enabling cross-modal reasoning and strong generalization capabilities. Among them, Contrastive Language-Image Pretraining (CLIP) (Radford et al. 2021) stands out as a pioneering work. CLIP learns to associate images with natural language descriptions using a simple yet effective contrastive objective, achieving remarkable zero-shot classification performance without the need for task-specific fine-tuning. The success of CLIP has led to a series of follow-up VLMs, such as ALIGN (Jia et al. 2021), BLIP (Li et al. 2022, 2023; Xue et al. 2024), Flamingo (Alayrac et al. 2022), GPT-4V (Yang et al. 2023), and LLaVA (Liu et al. 2024), which have further improved visual reasoning by leveraging rich semantic priors from language.

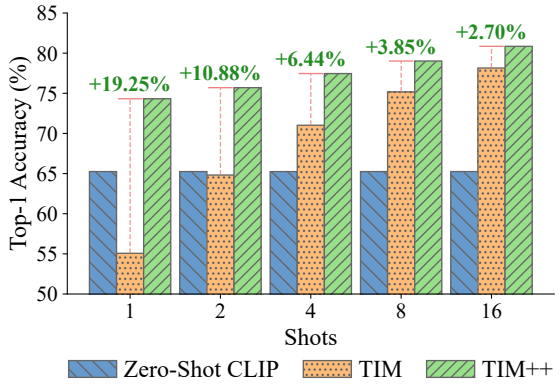

Figure 1: **TIM++ significantly outperforms standard TIM**. Average Top-1 accuracy is reported across 11 public datasets under different shot settings.

The strong generalization ability of these VLMs stems from their cross-modal pretraining on large-scale vision-language data, enabling effective performance even in zero-shot and few-shot scenarios. However, adapting these models to downstream tasks still poses significant challenges, particularly in the low-data regime. Most existing adaptation approaches rely on inductive learning, which treats each test sample independently and ignores statistical relationships among test instances. In contrast, transductive learning jointly considers the entire test set during inference, making it more suitable for few-shot scenarios.

Transductive methods have shown strong performance in visual domains (Belhasin, Bar-Shalom, and El-Yaniv 2022), with representative approaches such as LaplacianShot (Ziko et al. 2020), BD-CSPN (Liu, Song, and Qin 2020), and TIM (Boudiaf et al. 2020). Among them, Transductive Information Maximization (TIM) (Boudiaf et al. 2020) serves as a widely adopted baseline for transductive few-shot classification for its computational efficiency and effectiveness.

*Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

It jointly maximizes the mutual information between query features and their predicted labels, while leveraging support set supervision through a cross-entropy loss.

Nevertheless, these vision-only transductive methods overlook a key advantage of VLMs—the availability of textual information. Recent works such as LP++ (Huang et al. 2024) and TransCLIP (Zanella, Gérin, and Ayed 2024) have shown that incorporating CLIP-based textual supervision can substantially enhance few-shot performance. Inspired by these findings, we revisit the Transductive Information Maximization (TIM) method and observe that its full potential remains underexplored, as it still does not leverage textual information from VLMs like CLIP. To address this gap, we propose TIM++, a novel framework that enhances TIM by integrating both visual and textual information for transductive few-shot learning.

Our contributions are summarized as follows:

- We propose TIM++, an extension of standard TIM that integrates textual knowledge from CLIP by introducing a Kullback-Leibler (KL) divergence-based regularization. This regularization aligns the model's posterior predictions with zero-shot CLIP outputs, encouraging the model to match the most confident predictions in CLIP's output distribution.
- We design a new initialization strategy for class prototypes by augmenting the support set features with the query features and their corresponding textual predictions from CLIP. This leads to a more semantically enriched and task-adaptive classifier initialization.
- Extensive experiments on 11 public datasets show that TIM++ consistently outperforms the original TIM, achieving an average improvement of up to 19.25% in 1-shot settings, as shown in Figure 1. TIM++ also surpasses other state-of-the-art methods. Ablation studies further validate the complementary benefits of KL-based textual regularization and the improved initialization strategy.

By bridging the gap between transductive inference and textual supervision, TIM++ demonstrates the strength of multimodal learning in few-shot adaptation and sets a new benchmark for vision-language transfer in few-shot CLIP.

Related Work

Few-shot Learning with Vision-only Models

Before the advent of VLMs, few-shot learning methods were primarily developed for vision-only models. These approaches fall into two broad categories: *inductive* and *transductive* learning. Inductive learning methods infer each query sample independently, ignoring relationships among test samples. Although simple and efficient, they often underperform in few-shot settings due to their inability to leverage the structure within the query set. Transductive learning methods, on the other hand, perform joint inference by exploiting the statistical relationships among query samples. This leads to improved performance, especially when labeled data is scarce.

Several representative transductive methods have been proposed. Transductive Fine-Tuning (TF) (Dhillon et al. 2019) introduces the entropy of query predictions as a regularization term. LaplacianShot (Ziko et al. 2020) uses graph-based clustering to align each query with the nearest support prototype while encouraging consistent predictions for similar samples. BD-CSPN (Liu, Song, and Qin 2020) refines class prototypes by combining information from support samples and the highest-confidence query samples. PT-MAP (Hu, Gripon, and Pateux 2021) first preprocesses feature vectors to approximate Gaussian distributions and then applies an optimal transport–inspired algorithm that leverages unlabeled query samples and their class proportion priors. Transductive Information Maximization (TIM) (Boudiaf et al. 2020) maximizes the mutual information between query features and predicted labels while using cross-entropy supervision on the support set. Despite their effectiveness, these methods only utilize visual features and completely ignore textual knowledge available in VLMs, leaving the potential of multimodal learning untapped.

Recent advances have further enhanced transductive few-shot learning. ProtoLP (Zhu and Koniusz 2023) constructs an adaptive bipartite graph between class prototypes and query samples, iteratively updating prototypes and propagating labels to jointly infer the entire query set. Adaptive Manifold (Lazarou, Avrithis, and Stathaki 2024) constructs a manifold structure over labeled and unlabeled samples using manifold similarity, parameterized by per-class centroids and learnable manifold parameters optimized via a tunable loss function. Building on this foundation, the same authors introduce the iterative label propagation and cleaning (iLPC) method (Lazarou, Stathaki, and Avrithis 2025), which leverages manifold-based pseudo-labeling to iteratively incorporate the most confident unlabeled samples as labeled data, performs pseudo-label cleaning using unnormalized manifold similarities from label propagation, and remove distractor-class noise by exploiting imbalance patterns in pseudo-label distributions.

Zero-shot and Few-shot Learning with VLMs

The introduction of CLIP and its successors (e.g., ALIGN, BLIP, Flamingo, GPT-4V, LLaVA) has revolutionized zero-shot and few-shot learning by embedding visual and textual inputs into a shared embedding space. Zero-shot inference is achieved by computing similarity between image features and class-level textual descriptions. When adapting VLMs to downstream tasks with limited labeled data, three major paradigms have emerged:

Prompt Learning. This approach designs or learns task-specific prompts that are fed into the frozen text encoder to guide the model's predictions. Representative approaches such as CoOp (Zhou et al. 2022b), CoCoOp (Zhou et al. 2022a), ProGrad (Zhu et al. 2023), KgCoOP (Yao, Zhang, and Xu 2023) and others (Ma et al. 2023; Song, Wang, and Zhong 2024; Liu et al. 2025) treat the prompts as learnable textual embeddings. However, it risks overfitting on tiny datasets, lacks cross-task generalization, and depends heavily on initialization.

Adapter-based Methods. These approaches insert small trainable modules (adapters) into a frozen VLM, enabling

efficient fine-tuning without modifying the backbone, thus can operate in black-box settings. They typically focus on transforming these features, usually in a non-linear manner, to better adapt to downstream tasks. Representative works include CLIP-Adapter (Gao et al. 2024), Tip-Adapter (Zhang et al. 2022) and others (Ouali et al. 2023; Silva-Rodriguez et al. 2024). These methods make a balance between flexibility and efficiency. However, their performance often relies on careful design of the adapter architecture.

Feature Adaptation Methods. This category formulates downstream adaptation as an optimization problem directly over extracted visual and textual features. They can be seen as a lightweight variant of adapter-based methods, as they also transform features—typically through linear projections or clustering—without introducing additional trainable modules. For instance, TransCLIP (Zanella, Gérin, and Ayed 2024) performs feature clustering using a Gaussian Mixture Model and incorporates a KL divergence term to align predictions with CLIP-derived textual distributions. LP++ (Huang et al. 2024), another representative approach, extends the standard Linear Probe (LP) by incorporating CLIP-based textual knowledge into its objective. By analyzing the mathematical properties of its loss function, LP++ enables data-driven learning rate selection and efficient optimization, leading to improved accuracy and faster inference compared to existing few-shot methods.

Due to their efficiency, strong performance, and compatibility with frozen VLMs, feature adaptation methods are especially attractive in real-world, resource-limited and black-box scenarios. Our proposed TIM++ falls into this category and builds upon TIM. It combines the strengths of transductive inference with rich textual supervision, resulting in a fast and effective few-shot learning framework tailored for VLMs.

Our Proposed TIM++ Method

Preliminaries

Consider a pre-trained VLM such as CLIP. Let $\boldsymbol{x}_i$ represent an input image and $\boldsymbol{c}_k$ denote a textual description for class k, where $k = 1, \cdots, K$. The l_2-normalized embeddings of the input image $\boldsymbol{x}_i$ and the textual description $\boldsymbol{c}_k$, denoted as $\boldsymbol{f}_i \in \mathbb{R}^d$ and $\boldsymbol{t}_k \in \mathbb{R}^d$, respectively, are obtained by

$$\begin{cases} \boldsymbol{f}_i = \dfrac{\theta_v(\boldsymbol{x}_i)}{\| \theta_v(\boldsymbol{x}_i) \|_2} \\ \boldsymbol{t}_k = \dfrac{\theta_t(\boldsymbol{c}_k)}{\| \theta_t(\boldsymbol{c}_k) \|_2} \end{cases}. \tag{1}$$

Here, $\theta_v(\cdot)$ and $\theta_t(\cdot)$ are the frozen visual and text encoders of the pre-trained VLM, respectively, and d denotes the dimension of the embedding space.

In the standard Transductive Information Maximization (TIM) method, a soft classifier is trained on top of these extracted vision features. The posterior probability of class k given image $\boldsymbol{x}_i$ is defined as:

$$\begin{aligned} p_{ik} &:= \mathbb{P}(Y = k \mid X = \boldsymbol{x}_i; \mathbf{W}, \theta_v) \\ &= \frac{\exp(-\frac{\tau}{2} \| \boldsymbol{f}_i - \boldsymbol{w}_k \|^2)}{\sum_{j=1}^{K} \exp(-\frac{\tau}{2} \| \boldsymbol{f}_i - \boldsymbol{w}_j \|^2)}, \end{aligned} \tag{2}$$

where $\mathbf{W} := [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K] \in \mathbb{R}^{d \times K}$ denotes the matrix of classifier weights, and each $\boldsymbol{w}_k \in \mathbb{R}^d$ can be viewed as the feature prototype for class k. The scalar τ is a temperature parameter.

For a few-shot learning task with support set $\mathcal{S}$ and query set $\mathcal{Q}$, TIM takes a transductive modeling manner and defines its loss function as follows:

$$\min_{\mathbf{W}} \lambda \cdot \mathrm{CE} - \hat{\mathcal{I}}_\alpha(X_\mathcal{Q}; Y_\mathcal{Q}), \tag{3}$$

where CE represents cross-entropy over the support set, and $\hat{\mathcal{I}}_\alpha(X_\mathcal{Q}; Y_\mathcal{Q})$ represents the mutual information between query features and their label predictions. Specifically,

$$\mathrm{CE} := -\frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{k=1}^{K} y_{ik} \log p_{ik} \tag{4}$$

and

$$\begin{aligned} &\hat{\mathcal{I}}_\alpha(X_\mathcal{Q}; Y_\mathcal{Q}) \\ &= \underbrace{-\sum_{k=1}^{K} \widehat{p}_k \log \widehat{p}_k}_{\widehat{\mathcal{H}}(Y_\mathcal{Q})\text{: marginal entropy}} + \underbrace{\frac{\alpha}{|\mathcal{Q}|} \sum_{i \in \mathcal{Q}} \sum_{k=1}^{K} p_{ik} \log p_{ik}}_{-\widehat{\mathcal{H}}(Y_\mathcal{Q} | X_\mathcal{Q})\text{: conditional entropy}}, \end{aligned} \tag{5}$$

where $y_{ik} \in \{0, 1\}$ indicates whether sample $i \in \mathcal{S}$ belongs to class k, and $\widehat{p}_k$ is the marginal class distribution over the query set, defined as

$$\widehat{p}_k = \frac{1}{|\mathcal{Q}|} \sum_{i \in \mathcal{Q}} p_{ik}. \tag{6}$$

Notably, in the TIM objective, the cross-entropy loss provides supervision from labeled support samples. The conditional entropy term $\widehat{\mathcal{H}}(Y_\mathcal{Q} | X_\mathcal{Q})$ encourages the model to produce confident predictions on the query set by reducing posterior uncertainty. In parallel, the marginal entropy term $-\widehat{\mathcal{H}}(Y_\mathcal{Q})$ encourages the marginal distribution of labels to be uniform, thereby preventing degenerate solutions that may arise when only minimizing the cross-entropy term with conditional entropy. The trade-off coefficients $\lambda > 0$ and $\alpha > 0$ control the strengths of supervised loss and entropy regularization, respectively.

Formulation of TIM++

In what follows, we denote by $\boldsymbol{p} = [p_{ik}] \in \mathbb{R}^{|\mathcal{Q}| \times K}$ the matrix of model-predicted posterior probabilities for all query samples, where each row $\boldsymbol{p}_i = (p_{i1}, \ldots, p_{iK})$ represents the predicted posterior distribution over classes for the i-th query sample. Similarly, $\hat{\boldsymbol{y}} = [\hat{y}_{ik}] \in \mathbb{R}^{|\mathcal{Q}| \times K}$ denotes the matrix of zero-shot predictions from CLIP, with $\hat{\boldsymbol{y}}_i =$

$(\hat{y}_{i1}, \ldots, \hat{y}_{iK})$ as the corresponding zero-shot class probabilities for the i-th query. The zero-shot prediction from CLIP is computed as:

$$\hat{y}_{ik} = \frac{\exp(\eta \boldsymbol{f}_i^\top \boldsymbol{t}_k)}{\sum_{j=1}^{K} \exp(\eta \boldsymbol{f}_i^\top \boldsymbol{t}_j)}, \tag{7}$$

where η is a temperature scaling factor, fixed to 100 in our experiments.

The standard TIM framework completely ignores the text embeddings provided by CLIP, i.e., $(\boldsymbol{t}_k)_{1\le k\le K}$. To address this limitation, we propose TIM++, which incorporates these text embeddings into the objective function through a KL-divergence-based regularization term. The overall loss function of TIM++ is formulated as:

$$\min_{\mathbf{W}} \lambda \cdot \text{CE} - \hat{\mathcal{I}}_\alpha(X_\mathcal{Q}; Y_\mathcal{Q}) + \beta \cdot \mathcal{D}_{\text{KL}}(\boldsymbol{p} \parallel \hat{\boldsymbol{y}}) \,, \tag{8}$$

where $\beta > 0$ controls the overall strength of the KL regularization term, and

$$\mathcal{D}_{\text{KL}}(\boldsymbol{p} \parallel \hat{\boldsymbol{y}}) := \frac{1}{|\mathcal{Q}|} \sum_{i\in\mathcal{Q}} \mathcal{D}_{\text{KL}}(\boldsymbol{p}_i \parallel \hat{\boldsymbol{y}}_i) \tag{9}$$

is the average KL divergence between the model's posterior predictions $\boldsymbol{p}_i$ and CLIP's zero-shot predictions $\hat{\boldsymbol{y}}_i$. The KL divergence for each query sample i is defined as:

$$\mathcal{D}_{\text{KL}}(\boldsymbol{p}_i \parallel \hat{\boldsymbol{y}}_i) = \sum_{k=1}^{K} (p_{ik} \log p_{ik} - \frac{\gamma}{\beta}\, p_{ik} \log \hat{y}_{ik}), \tag{10}$$

where $\frac{\gamma}{\beta} > 0$ adjusts the relative weight between the entropy and cross-entropy terms. When $\gamma = \beta$, this reduces to the standard KL divergence.

Importantly, as demonstrated in our experiments, the direction of the KL divergence significantly affects model performance. We adopt the *model-seeking* form $\mathcal{D}_{\text{KL}}(\boldsymbol{p}_i \parallel \hat{\boldsymbol{y}}_i)$, which encourages the learned posterior $\boldsymbol{p}_i$ to concentrate on regions where $\hat{\boldsymbol{y}}_i$ assigns high probability, promoting precise alignment with strong semantic cues. In contrast, the *model-covering* form $\mathcal{D}_{\text{KL}}(\hat{\boldsymbol{y}}_i \parallel \boldsymbol{p}_i)$ penalizes low-probability regions in $\boldsymbol{p}_i$ where $\hat{\boldsymbol{y}}_i$ has non-zero mass. This can be problematic when CLIP predictions $\hat{\boldsymbol{y}}_i$ are noisy or uncertain, often leading to unstable optimization. For this reason, we choose $\mathcal{D}_{\text{KL}}(\boldsymbol{p}_i \parallel \hat{\boldsymbol{y}}_i)$ in constructing TIM++.

Optimization

The objective of the proposed TIM++ is defined in Eq. (8). Notably, the pre-trained vision and text encoders $\theta_v(\cdot)$ and $\theta_t(\cdot)$ from the VLMs are kept fixed, making TIM++ well-suited for adapting black-box models. The only parameter optimized in TIM++ is the classifier weight matrix $\mathbf{W}$. However, directly solving objective (8) is non-trivial due to nonlinear terms involving the prediction probabilities p_{ik} in both linear and logarithmic forms. To simplify the optimization, as in the standard TIM, we introduce auxiliary variables $\boldsymbol{q}$ to represent the latent class assignments of query samples, ensuring that p_{ik} appears only within logarithmic expressions. This leads to the following reformulation of objective (8), stated in the proposition below.

Proposition 1. *The objective function* (8) *can be approximately minimized by the following constrained problem:*

$$\begin{aligned}
\min_{\mathbf{W},\, \boldsymbol{q}} & \underbrace{-\frac{\lambda}{|\mathcal{S}|} \sum_{i\in\mathcal{S}} \sum_{k=1}^{K} y_{ik} \log p_{ik}}_{\text{CE}} + \underbrace{\sum_{k=1}^{K} \widehat{q}_k \log \widehat{q}_k}_{\sim -\widehat{\mathcal{H}}(Y_\mathcal{Q})} \\
& \underbrace{-\frac{\alpha}{|\mathcal{Q}|} \sum_{i\in\mathcal{Q}} \sum_{k=1}^{K} q_{ik} \log p_{ik}}_{\sim \widehat{\mathcal{H}}(Y_\mathcal{Q}|X_\mathcal{Q})} + \underbrace{\frac{1}{|\mathcal{Q}|} \sum_{i\in\mathcal{Q}} \sum_{k=1}^{K} q_{ik} \log \frac{q_{ik}}{p_{ik}}}_{\mathcal{D}_{\text{KL}}(\boldsymbol{q} \parallel \boldsymbol{p})} \\
& + \underbrace{\frac{\beta}{|\mathcal{Q}|} \sum_{i\in\mathcal{Q}} \sum_{k=1}^{K} q_{ik} \log p_{ik} - \frac{\gamma}{|\mathcal{Q}|} \sum_{i\in\mathcal{Q}} \sum_{k=1}^{K} q_{ik} \log \hat{y}_{ik}}_{\sim \mathcal{D}_{\text{KL}}(\boldsymbol{p} \parallel \hat{\boldsymbol{y}})} \\
\text{s.t.} \quad & \sum_{k=1}^{K} q_{ik} = 1, \quad i \in \mathcal{Q} \\
& q_{ik} \ge 0, \quad i \in \mathcal{Q}, \quad k \in \{1, \cdots, K\},
\end{aligned} \tag{11}$$

where $\boldsymbol{q} = [q_{ik}] \in \mathbb{R}^{|\mathcal{Q}|\times K}$ *denotes the auxiliary variables representing the class assignment probabilities for query samples, and* $\widehat{q}_k = \frac{1}{|\mathcal{Q}|} \sum_{i\in\mathcal{Q}} q_{ik}$ *is the corresponding marginal distribution.*

Proof. The term $\mathcal{D}_{\text{KL}}(\boldsymbol{q} \parallel \boldsymbol{p})$ acts as a soft penalty that encourages consistency between q_{ik} and p_{ik}. When $q_{ik} = p_{ik}$, this term vanishes, and the objectives in Eq. (8) and Eq. (11) become equivalent. A detailed proof is provided in Appendix. □

We now turn to solving the constrained problem (11). To this end, we adopt the Alternating Direction Method (ADM) (Boudiaf et al. 2020), which decomposes the original problem into two simpler subproblems and solves them alternately. Specifically, the subproblems are defined with respect to $\mathbf{W}$ and $\boldsymbol{q}$; when solving one, the other is kept fixed. The iterative procedure is summarized in the following proposition.

Proposition 2. *The constrained optimization problem in Eq.* (11) *can be approximately solved using the Alternating Direction Method (ADM), by alternately updating the auxiliary variables* $\boldsymbol{q}$ *and the classifier weights* $\mathbf{W}$. *At iteration* $t+1$, *the updates are given by:*

$$\begin{aligned}
q_{ik}^{(t+1)} & \propto \frac{\hat{y}_{ik}^{\gamma} \cdot (p_{ik}^{(t)})^{\alpha-\beta+1}}{\Big(\sum_{i\in\mathcal{Q}} \hat{y}_{ik}^{\gamma} \cdot (p_{ik}^{(t)})^{\alpha-\beta+1}\Big)^{\frac{1}{2}}} \\
\boldsymbol{w}_k^{(t+1)} & \leftarrow \frac{S_1 + S_2}{\frac{\lambda\tau}{|\mathcal{S}|} \sum_{i\in\mathcal{S}} y_{ik} + \frac{(\alpha-\beta+1)\tau}{|\mathcal{Q}|} \sum_{i\in\mathcal{Q}} q_{ik}^{(t+1)}}
\end{aligned}, \tag{12}$$

where the parameters must satisfy $\alpha - \beta + 1 > 0$, *and* S_1

and S_2 are defined as:

$$\begin{cases} S_1 = \dfrac{\lambda\tau}{|\mathcal{S}|}\displaystyle\sum_{i\in\mathcal{S}}\left[y_{ik}\boldsymbol{f}_i + p_{ik}^{(t)}(\boldsymbol{w}_k^{(t)} - \boldsymbol{f}_i)\right] \\ S_2 = \dfrac{(\alpha-\beta+1)\tau}{|\mathcal{Q}|}\displaystyle\sum_{i\in\mathcal{Q}}\left[q_{ik}^{(t+1)}\boldsymbol{f}_i + p_{ik}^{(t)}(\boldsymbol{w}_k^{(t)} - \boldsymbol{f}_i)\right] \end{cases}. \tag{13}$$

Proof. Detailed derivations are provided in Appendix. The ADM framework solves the problem by alternating between optimizing $\mathbf{W}$ and $\boldsymbol{q}$. When updating $\mathbf{W}$ with fixed $\boldsymbol{q}$, we require the condition $\alpha - \beta + 1 > 0$ to ensure a convex approximation of the objective function—hence the necessity of this assumption. Given $\mathbf{W}$, the update of $\boldsymbol{q}$ can be derived by solving the Karush-Kuhn-Tucker (KKT) conditions. □

Convergence and Initialization Strategy

According to the ADM algorithm, the value of the objective function does not increase at each iteration, and it is lower-bounded based on the update rules for $\mathbf{W}$ and $\boldsymbol{q}$. Therefore, local convergence of TIM++ is guaranteed. Notably, the standard TIM follows a similar alternating optimization structure and therefore also enjoys local convergence. Consequently, both standard TIM and our proposed TIM++ exhibit local convergence, making the quality of initialization critical in determining the final local optimum and overall performance.

In the standard TIM framework, the classifier weights $\mathbf{W}$ are initialized as the class-wise average of the encoded visual features from the support set $\mathcal{S}$:

$$\boldsymbol{w}_k^{(0)} = \frac{\sum_{i\in\mathcal{S}} y_{ik}\boldsymbol{f}_i}{\sum_{i\in\mathcal{S}} y_{ik}}, \quad k = 1, 2, \ldots, K. \tag{14}$$

However, our experiments show that incorporating information from the query set $\mathcal{Q}$ during initialization substantially improves performance for both TIM and TIM++. Motivated by this, we propose to initialize the classifier weights as:

$$\boldsymbol{w}_k^{(0)} = \frac{\sum_{i\in\mathcal{S}} y_{ik}\boldsymbol{f}_i + \sum_{i\in\mathcal{Q}} \hat{y}_{ik}\boldsymbol{f}_i}{\sum_{i\in\mathcal{S}} y_{ik} + \sum_{i\in\mathcal{Q}} \hat{y}_{ik}}, \quad k = 1, 2, \ldots, K, \tag{15}$$

where $\hat{y}_{ik}$ denotes the k-th component of the hard-coded CLIP zero-shot prediction vector for query sample $i \in \mathcal{Q}$. We use the hard-coded form because it aligns better with the ground-truth labels of the support set, leading to more stable convergence.

This initialization strategy brings two main benefits: (1) it leverages both labeled support features and pseudo-labeled query features to form more representative prototypes, and (2) it injects semantic priors from CLIP through $\hat{y}_{ik}$, yielding a stronger starting point for optimization. As a result, Eq. (15) is adopted as the default initialization in our proposed TIM++ method.

Extension to the Zero-shot Learning

Although TIM++ is originally designed for few-shot learning, its optimization framework can be naturally extended to the zero-shot setting. In the few-shot scenario, the model leverages supervision from the support set through the cross-entropy term in the objective function (8). To enable zero-shot learning, we simply remove this supervised component by setting $\lambda = 0$, thereby relying solely on the entropy-based and KL-divergence regularization terms computed on the query set.

In this setting, the prototype initialization is also modified to eliminate support-set supervision. Specifically, the initialization term is redefined as:

$$\boldsymbol{w}_k^{(0)} = \frac{\sum_{i\in\mathcal{Q}} \hat{y}_{ik}\boldsymbol{f}_i}{\sum_{i\in\mathcal{Q}} \hat{y}_{ik}}, \quad k = 1, 2, \ldots, K, \tag{16}$$

where $\hat{y}_{ik}$ denotes the soft zero-shot prediction from CLIP. Soft predictions are used instead of hard pseudo-labels because, without support-set supervision, hard CLIP labels lead to poor initialization due to accumulation of incorrect pseudo-labels, whereas soft probabilistic outputs provide richer semantic information and result in more reliable performance.

This extension is conceptually aligned with the TENT method (Wang et al. 2021), which performs fully test-time adaptation by minimizing test entropy without using any labeled data. Similarly, TIM++ achieves zero-shot adaptation by discarding the cross-entropy term and optimizing only the entropy-based objectives along with a KL-divergence regularization term on the query set. This demonstrates the flexibility and generality of our framework, seamlessly bridging few-shot and zero-shot learning within a unified transductive formulation.

Algorithm 1: Our proposed TIM++ algorithm

Input: Pre-trained visual encoder $\theta_v(\cdot)$ and text encoder $\theta_t(\cdot)$; Images $\boldsymbol{x}_i$ from support set $\mathcal{S}$ and query set $\mathcal{Q}$; Labels $y_{ik} \in \{0, 1\}$ for $i \in \mathcal{S}$; Text prompts for classes $\{\boldsymbol{c}_k\}_{k=1}^K$.
Parameter: # Iterations T; Temperature parameter η and τ; Trade-off coefficients $\{\lambda, \alpha, \beta, \gamma\}$ satisfying $\alpha - \beta + 1 > 0$.

// Initialization.
$\boldsymbol{f}_i \leftarrow \frac{\theta_v(\boldsymbol{x}_i)}{\|\theta_v(\boldsymbol{x}_i)\|_2},\ i \in \mathcal{Q}\cup\mathcal{S}.$
$\boldsymbol{t}_k \leftarrow \frac{\theta_t(\boldsymbol{c}_k)}{\|\theta_t(\boldsymbol{c}_k)\|_2},\ k = 1, 2, \cdots, K.$
$\hat{y}_{ik} \leftarrow \frac{\exp(\eta \boldsymbol{f}_i^\top \boldsymbol{t}_k)}{\sum_j \exp(\eta \boldsymbol{f}_i^\top \boldsymbol{t}_j)}, \quad i \in \mathcal{Q}, \quad k = 1, 2, \cdots, K.$
$\boldsymbol{w}_k^{(0)} \leftarrow \frac{\sum_{i\in\mathcal{S}} y_{ik}\boldsymbol{f}_i + \sum_{i\in\mathcal{Q}} \hat{y}_{ik}\boldsymbol{f}_i}{\sum_{i\in\mathcal{S}} y_{ik} + \sum_{i\in\mathcal{Q}} \hat{y}_{ik}}, \quad k = 1, 2, \cdots, K.$
// Iterative updates of parameters.
for $t = 0, 1, \cdots T$ **do**
  $p_{ik}^{(t)} \leftarrow \frac{\exp(-\frac{\tau}{2}\|\boldsymbol{f}_i - \boldsymbol{w}_k^{(t)}\|^2)}{\sum_{j=1}^K \exp(-\frac{\tau}{2}\|\boldsymbol{f}_i - \boldsymbol{w}_j^{(t)}\|^2)},\ i \in \mathcal{Q}\cup\mathcal{S}$
  $q_{ik}^{(t+1)} \leftarrow \frac{\hat{y}_{ik}^{\gamma}\cdot(p_{ik}^{(t)})^{\alpha-\beta+1}}{\left(\sum_{i\in\mathcal{Q}} \hat{y}_{ik}^{\gamma}\cdot(p_{ik}^{(t)})^{\alpha-\beta+1}\right)^{1/2}}, \quad i \in \mathcal{Q}$
  $\boldsymbol{w}_k^{(t+1)} \leftarrow \frac{S_1 + S_2}{\frac{\lambda\tau}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} y_{ik} + \frac{(\alpha-\beta+1)\tau}{|\mathcal{Q}|}\sum_{i\in\mathcal{Q}} q_{ik}^{(t+1)}},$
  see S_1, S_2 in Eq. (13), $k = 1, 2, \cdots, K$.
end for
Output: Predictions for sample $i \in \mathcal{Q}$: $\arg\max_k p_{ik}^{(T)}$.

Method	1-shot	2-shot	4-shot	8-shot	16-shot
Zero-shot CLIP $_{\text{ICML'21}}$			65.25		
TF $_{\text{Arxiv'19}}$	43.50 ± 2.13	56.99 ± 1.68	65.22 ± 0.84	70.70 ± 0.75	73.74 ± 0.57
BD-CSPN $_{\text{ICCV'20}}$	51.02 ± 1.83	61.86 ± 1.69	67.01 ± 1.23	70.81 ± 0.89	73.59 ± 0.65
LaplacianShot $_{\text{ICML'20}}$	51.53 ± 2.21	62.63 ± 1.28	67.45 ± 1.24	70.50 ± 0.71	72.94 ± 0.49
PT-MAP $_{\text{ICANN'21}}$	53.02 ± 2.23	62.39 ± 1.64	66.12 ± 0.56	68.70 ± 0.43	70.53 ± 0.45
LP $_{\text{ICML'21}}$	45.35 ± 1.33	56.57 ± 0.98	66.18 ± 1.30	73.36 ± 0.45	78.23 ± 0.43
LP++ $_{\text{CVPR'24}}$	70.51 ± 0.68	72.91 ± 0.53	75.58 ± 0.72	<u>77.90</u> ± 0.52	<u>80.33</u> ± 0.20
TransCLIP $_{\text{NeurIPS'24}}$	<u>73.29</u> ± 1.20	<u>74.18</u> ± 1.36	<u>75.97</u> ± 0.67	77.29 ± 0.62	78.34 ± 0.63
TIM $_{\text{NeurIPS'20}}$	55.07 ± 2.83	64.82 ± 1.85	71.02 ± 1.72	75.17 ± 1.34	78.15 ± 1.22
TIM++	**74.32** ± 0.61	**75.70** ± 0.64	**77.46** ± 0.72	**79.02** ± 0.48	**80.85** ± 0.34
Δ	+19.25 %	+10.88%	+6.44%	+3.85%	+2.70%

Table 1: **Comparison with state-of-the-art methods.** Average accuracy (%) ± standard deviation over three random seeds, averaged across 11 public datasets, is reported. The best results are shown in **bold**, the second-best are <u>underlined</u>, and Δ denotes the improvement of TIM++ over standard TIM.

Experiments

Experimental Settings

Datasets. In our experiments, we utilize 11 publicly available datasets: ImageNet (Deng et al. 2009), SUN397 (Xiao et al. 2010), FGVC-Aircraft (Maji et al. 2013), EuroSAT (Helber et al. 2019), Stanford Cars (Krause et al. 2013), Food101 (Bossard, Guillaumin, and Van Gool 2014), Oxford Pets (Parkhi et al. 2012), Oxford Flowers (Nilsback and Zisserman 2008), Caltech101 (Fei-Fei, Fergus, and Perona 2004), DTD (Cimpoi et al. 2014), and UCF101 (Soomro, Zamir, and Shah 2012). These datasets are consistent with those used in prior works such as TransCLIP and LP++, allowing for a fair comparison. This diverse set of benchmarks is selected to comprehensively assess the effectiveness and generalization ability of the proposed TIM++ method as well as other few-shot learning approaches.

Comparison Methods. We compare the proposed TIM++ method with state-of-the-art approaches in few-shot learning, with zero-shot CLIP serving as the baseline. Our work builds upon the standard TIM, and the comparisons include transductive vision-only methods (TF, BD-CSPN, LaplacianShot, PT-MAP) as well as recent feature adaptation approaches tailored for VLMs, such as TransCLIP and LP++. Prompt learning and adapter-based methods are excluded, as LP++ has already demonstrated superior performance and significantly lower inference time. Specifically, prompt learning typically requires hours or even days for training, adapter-based methods take minutes to hours, whereas feature adaptation approaches like LP++ and TransCLIP perform parameter estimation and inference directly within seconds. For fairness, all comparison methods use the same hyperparameter settings as in the official LP++ and TransCLIP implementations.

Implementation Details. In all experiments, we use the pre-trained CLIP ViT-B/16 as the visual encoder. To ensure consistency, we adopt fixed prompt templates for each dataset throughout all experiments, as summarized in Appendix Table 5. All experiments are conducted using three random seeds (1, 2, and 3), and the average Top-1 accuracy is reported. The experiments are conducted using PyTorch 2.0.1 with CUDA 12.2 on Ubuntu 22.04, running on a system with an NVIDIA RTX A5000 GPU, 64GB RAM, and a 13th Gen Intel(R) Core(TM) i9-13900K CPU.

The objective function of TIM++ involves several hyperparameters, including the temperature parameter τ used to compute the posterior distribution p_{ik}, and a set of trade-off coefficients: λ, α, β, and γ. To ensure fair and consistent evaluation, we fix all hyperparameters across all experiments: $\tau = 120$, $\lambda = 0.4$, and $\gamma = 0.05$. As summarized in Proposition 2, the iterative procedure depends only on the difference $\alpha - \beta$, so we directly set $\alpha - \beta = 0.1$, which also satisfies the requirement $\alpha - \beta + 1 > 0$. In addition, we perform $T = 150$ fine-tuning steps during the TIM++ optimization. These hyperparameters are selected through hyperparameter tuning based on validation performance (see details in Appendix), and the chosen configuration achieves the best average validation results across the 11 datasets.

Main results

Comparison to Standard TIM. Table 1 presents the performance of TIM++ compared to TIM and other baseline methods. As shown in Table 1 and Figure 1, TIM++ consistently and significantly outperforms the standard TIM across all few-shot settings. Specifically, on the 11 public datasets, TIM++ improves the average Top-1 accuracy over TIM by 19.25%, 10.88%, 6.44%, 3.85%, and 2.70% under the 1-shot, 2-shot, 4-shot, 8-shot, and 16-shot settings, respectively.

These consistent gains demonstrate the effectiveness of incorporating textual knowledge into TIM, particularly in low-shot scenarios. The performance gap is most pronounced in the extremely low-data settings (e.g., 1-shot), where textual information provides a strong prior and compensates for the lack of visual supervision. As the number of support samples increases, the reliance on textual knowledge gradually decreases, resulting in a smaller margin of improvement.

Additionally, TIM++ demonstrates remarkable stability: it consistently maintains an average accuracy above 74% across all few-shot settings, including the most challenging 1-shot case. In contrast, the original TIM achieves only 55.07% in the 1-shot setting, underscoring the critical role of textual priors and our improved optimization strategy in low-resource scenarios.

Comparison to the State-of-the-Art. Table 1 and Figure 2 present the average Top-1 accuracy of different methods across 11 public datasets under various few-shot settings (1, 2, 4, 8, and 16 shots). As shown, our proposed method TIM++ achieves the best performance in all settings, consistently outperforming both classical vision-only few-shot learning methods and recent state-of-the-art methods designed for VLMs like TransCLIP and LP++.

Compared to classical visual-only few-shot methods such as TF, BD-CSPN, LaplacianShot, PT-MAP, and LP, TIM++ delivers substantial performance gains. For instance, under the challenging 1-shot setting, TIM++ surpasses the best visual-only baseline (PT-MAP at 53.02%) by 21.30%, averaged on 11 public datasets. Compared to recent state-of-the-art methods that also incorporate textual knowledge, such as TransCLIP and LP++, TIM++ consistently achieves higher accuracy across all settings. On average, it outperforms TransCLIP and LP++ by 1.65% and 2.02%, respectively, demonstrating the effectiveness of the proposed TIM++.

Zero-shot Adaptation. As detailed in the previous section, our TIM++ framework can be naturally extended to zero-shot settings. Notably, among the comparison methods, only TransCLIP supports zero-shot adaptation and thus serves as the main baseline for this evaluation. Across the 11 public datasets, the average zero-shot Top-1 accuracy of CLIP is 65.25%. By applying our TIM++ adaptation method, this average improves notably to 69.81%, demonstrating the effectiveness of leveraging unsupervised objectives to refine CLIP's predictions without any labeled support data. This performance is comparable to TransCLIP's zero-shot adaptation, achieving an average Top-1 accuracy of 70.33%.

Moreover, similar to TransCLIP, the zero-shot version of TIM++ can be applied on top of prompt tuning methods, further boosting performance in both in-domain and domain generalization tasks. Experimental results for these settings are provided in the Appendix. We also include evaluations of dataset transferability for zero-shot TIM++ to further demonstrate the robustness and versatility of our approach.

Ablation Study

Impact of Initialization Strategies. Table 2 evaluates how different initialization strategies for the classifier matrix $\mathbf{W} \in \mathbb{R}^{d \times K}$ affect the performance of TIM++. We compare five strategies: support set only ($\mathcal{S}$), query set with soft or hard CLIP predictions ($\mathcal{Q}_{\text{soft}}$, $\mathcal{Q}_{\text{hard}}$), and their combinations. Results show that $\mathcal{S}+\mathcal{Q}_{\text{hard}}$ consistently yields the best performance across all shot settings, confirming that enriching the support prototypes with query samples assigned hard CLIP pseudo-labels yields a more reliable starting point.

Initialization	1	2	4	8	16
$\mathcal{S}$	65.80	71.75	75.41	78.01	80.34
$\mathcal{Q}_{\text{soft}}$	74.10	75.52	77.07	78.82	80.60
$\mathcal{Q}_{\text{hard}}$	46.05	46.40	46.92	47.36	47.96
$\mathcal{S}+\mathcal{Q}_{\text{soft}}$	74.07	75.48	77.10	78.79	80.52
$\mathcal{S}+\mathcal{Q}_{\text{hard}}$	**74.32**	**75.70**	**77.46**	**79.02**	**80.85**

Table 2: **Impact of initialization strategies for $\mathbf{W}$.** Accuracy (%) averaged over 11 datasets is reported. Columns indicate shots (1, 2, 4, 8, 16). $\mathcal{S}$: support set; $\mathcal{Q}$: query set; soft/hard: softmax/argmax CLIP predictions.

KL Direction	1	2	4	8	16
$\mathcal{D}_{\text{KL}}(\hat{\boldsymbol{y}}_i \parallel \boldsymbol{p}_i)$	45.79	50.57	53.47	56.34	60.25
$\mathcal{D}_{\text{KL}}(\boldsymbol{p}_i \parallel \hat{\boldsymbol{y}}_i)$	**74.32**	**75.70**	**77.46**	**79.02**	**80.85**

Table 3: **Impact of KL Divergence Direction for Textual Knowledge Integration.** Accuracy (%) averaged over 11 datasets is reported. Columns indicate shots (1, 2, 4, 8, 16).

This leads to substantial gains over the standard TIM initialization that relies solely on the support set (e.g., 74.32% vs. 65.80% in 1-shot), and is therefore adopted as the default strategy in TIM++.

Notably, using only query samples with hard CLIP predictions ($\mathcal{Q}_{\text{hard}}$) performs poorly, underscoring the need for labeled support features to guide the initialization and mitigate errors from incorrect hard pseudo-labels. Using soft CLIP predictions ($\mathcal{Q}_{\text{soft}}$) also improves over the support-only baseline, likely because soft predictions provide richer semantic cues and the larger query set offers broader class coverage—especially in low-shot settings with limited support data. When combined with the support set, however, hard predictions align more consistently with ground-truth labels, leading to more stable optimization and a slight but consistent improvement over $\mathcal{S}+\mathcal{Q}_{\text{soft}}$.

Impact of KL Divergence Direction for Textual Knowledge Integration. The KL divergence term in TIM++ is designed to align the model's posterior distribution $\boldsymbol{p}_i$ with the zero-shot CLIP prediction $\hat{\boldsymbol{y}}_i$ for each query sample $i \in \mathcal{Q}$, thereby injecting textual supervision into the model. However, the direction of the KL divergence plays a critical role in its effectiveness.

As shown in Table 3, using the model-seeking formulation $\mathcal{D}_{\text{KL}}(\boldsymbol{p}_i \parallel \hat{\boldsymbol{y}}_i)$ consistently and substantially outperforms the reverse direction $\mathcal{D}_{\text{KL}}(\hat{\boldsymbol{y}}_i \parallel \boldsymbol{p}_i)$ across all shot settings. This empirical finding aligns with the theoretical behavior of asymmetric KL divergence: the model-seeking form $\mathcal{D}_{\text{KL}}(\boldsymbol{p}_i \parallel \hat{\boldsymbol{y}}_i)$ treats the CLIP prediction $\hat{\boldsymbol{y}}_i$ as the reference and encourages the model to align with its high-confidence regions, making it better suited for our purpose.

Ablation Study of TIM++ Components. Table 4 presents an ablation study on the key components of TIM++, including the improved initialization strategy $\mathcal{S}+\mathcal{Q}_{\text{hard}}$ (compared to the original $\mathcal{S}$ in TIM) and the KL diver-

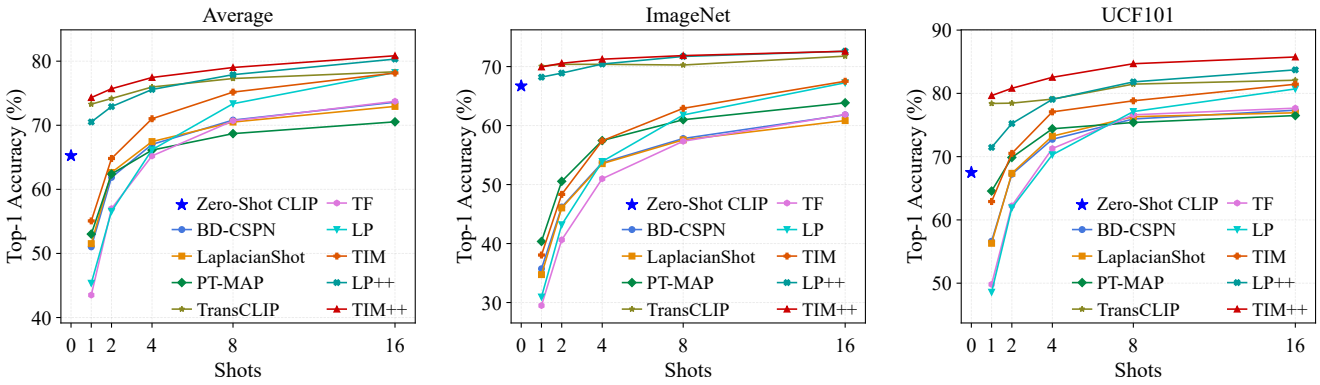

Figure 2: **TIM++ achieves superior performance over existing few-shot learning methods.** Average Top-1 accuracy is reported across 11 public datasets under different shot settings (1, 2, 4, 8, and 16).

Init.	KL	1	2	4	8	16
✗	✗	48.67	60.92	68.41	74.06	77.83
✓	✗	71.78	73.75	75.78	77.64	79.65
✗	✓	65.80	71.75	75.41	78.01	80.34
✓	✓	**74.32**	**75.70**	**77.46**	**79.02**	**80.85**

Table 4: **Ablation study of TIM++ components.** Top-1 accuracy (%) averaged over 11 datasets. Columns indicate shots (1, 2, 4, 8, 16). Init.: using $\mathcal{S} + \mathcal{Q}_{\text{hard}}$ (✓) or $\mathcal{S}$ (✗) for classifier weights $\mathbf{W}$ initialization. KL: whether the KL divergence term $D_{\text{KL}}(\boldsymbol{p}_i \parallel \hat{\boldsymbol{y}}_i)$ is used (✓) or not (✗).

gence term $\mathcal{D}_{\text{KL}}(\boldsymbol{p} \parallel \hat{\boldsymbol{y}})$ that incorporates textual information. When neither component is used, the method reduces to the standard TIM.

Clearly, the enhanced initialization brings substantial improvements across all shot settings. This gain can be attributed not only to the textual guidance from CLIP's zero-shot predictions but also to the inclusion of additional visual features from the query set. Together, these provide a more informative and representative class prototype than using the support set alone. Meanwhile, the KL divergence term also consistently enhances performance across all shot settings, demonstrating its central role in delivering robust textual supervision. Notably, combining both components yields the highest accuracy in every case, confirming their complementary benefits within the TIM++ framework.

Hyperparameter Analysis

To understand how each hyperparameter affects the performance of TIM++, we conducted a comprehensive analysis across the 11 public datasets under various few-shot settings. The hyperparameters are fixed as the final configuration: $\tau = 120$, $\lambda = 0.4$, $\gamma = 0.05$, and $\alpha - \beta = 0.1$. When examining the influence of one hyperparameter, all others are kept fixed. We observe that TIM++ exhibits relatively higher sensitivity on `EuroSAT`, `DTD`, `Oxford Flowers`, and `FGVC-Aircraft` datasets, especially in low-shot scenarios. In contrast, on the remaining seven datasets, performance remains relative stable across a broad range of values—for instance, $\alpha - \beta \in [-0.05, 0.8]$, $\tau \in [60, 200]$, $\gamma \in [0.025, 0.1]$, and $\lambda \in [0.025, 0.8]$. More details can be found in the Appendix.

Computational Cost

The parameters optimized in TIM++ include the classifier weight matrix $\mathbf{W} \in \mathbb{R}^{d \times K}$ and the auxiliary variables $\boldsymbol{q} \in \mathbb{R}^{|\mathcal{Q}| \times K}$. Therefore, the total number of parameters to be estimated is $K(d + |\mathcal{Q}|)$, which depends on the feature dimension d, the number of classes K, and the number of query samples $|\mathcal{Q}|$.

Importantly, TIM++ introduces no additional learnable parameters compared to the standard TIM, making its computational cost comparable. For instance, on the `ImageNet` dataset with $d = 512$, $K = 1000$, and $|\mathcal{Q}| = 50000$, under the 16-shot setting, TIM requires ~ 3.92 seconds for parameter estimation and inference, while TIM++ require approximately ~ 4.20 seconds. Similarly, on `UCF101` with $d = 512$, $K = 101$, and $|\mathcal{Q}| = 3783$, TIM requires ~ 0.087 seconds, whereas TIM++ takes around ~ 0.090 seconds.

Conclusion

In this paper, we proposed TIM++, a transductive few-shot learning framework that improves upon the standard TIM by effectively integrating textual knowledge from CLIP. Specifically, TIM++ introduces a KL divergence-based regularization in its modeling framework to align model predictions with zero-shot CLIP outputs, alongside an improved initialization strategy. Experiments on 11 datasets demonstrate that TIM++ consistently and significantly outperforms the original TIM, achieving up to 19.25% improvement in the 1-shot setting. It also surpasses other state-of-the-art methods across all few-shot settings. Notably, TIM++ maintains over 74% average accuracy in every shot setting, demonstrating strong robustness and generalization. Ablation studies further confirm the complementary benefits of the KL-based textual supervision and the enhanced initialization strategy, highlighting the effectiveness of TIM++ in leveraging textual information for few-shot CLIP adaptation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62302355, 62372358, 62301395, 12401429, No. T2541002, T2541005), the Key Research and Development Program of Shaanxi Province (Grant No. 2024NC-ZDCYL-05-04, 2024SF2-GJHX-35, 2024SF-GJHX-34), the Natural Science Basic Research Program of Shaanxi (No. 2024JC-YBQN-0046), the China Postdoctoral Science Foundation (No. 2023M742750), the Shaanxi Fundamental Science Research Project for Mathematics and Physics (Grant No. 23JSZ010), and the Fundamental Research Funds for the Central Universities (Grant No. ZYTS25210, XJSJ24071 and XJSJ24072).

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.

Belhasin, O.; Bar-Shalom, G.; and El-Yaniv, R. 2022. Transboost: Improving the best imagenet performance using deep transduction. *Advances in neural information processing systems*, 35: 28363–28373.

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, 446–461. Springer.

Boudiaf, M.; Ziko, I.; Rony, J.; Dolz, J.; Piantanida, P.; and Ben Ayed, I. 2020. Information maximization for few-shot learning. *Advances in Neural Information Processing Systems*, 33: 2445–2457.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3606–3613.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. IEEE.

Dhillon, G. S.; Chaudhari, P.; Ravichandran, A.; and Soatto, S. 2019. A Baseline for Few-Shot Image Classification. arXiv:1909.02729.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, 178–178. IEEE.

Gao, P.; Geng, S.; Zhang, R.; Ma, T.; Fang, R.; Zhang, Y.; Li, H.; and Qiao, Y. 2024. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2): 581–595.

Helber, P.; Bischke, B.; Dengel, A.; and Borth, D. 2019. 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.

Hu, Y.; Gripon, V.; and Pateux, S. 2021. Leveraging the feature distribution in transfer-based few-shot learning. In *International conference on artificial neural networks*, 487–499. Springer.

Huang, Y.; Shakeri, F.; Dolz, J.; Boudiaf, M.; Bahig, H.; and Ben Ayed, I. 2024. LP++: A Surprisingly Strong Linear Probe for Few-Shot CLIP. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23773–23782.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, 4904–4916. PMLR.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 554–561.

Lazarou, M.; Avrithis, Y.; and Stathaki, T. 2024. Adaptive manifold for imbalanced transductive few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2297–2306.

Lazarou, M.; Stathaki, T.; and Avrithis, Y. 2025. Exploiting unlabeled data in few-shot learning with manifold similarity and label cleaning. *Pattern Recognition*, 161: 111304.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 19730–19742. PMLR.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, 12888–12900. PMLR.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 26296–26306.

Liu, H.; Wang, Y.; Zhang, X.; Zhang, F.; Wang, W.; Ma, F.; and Yu, H. 2025. Multi-Label Few-Shot Image Classification via Pairwise Feature Augmentation and Flexible Prompt Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 5433–5441.

Liu, J.; Song, L.; and Qin, Y. 2020. Prototype rectification for few-shot learning. In *European conference on computer vision*, 741–756. Springer.

Ma, X.; Zhang, J.; Guo, S.; and Xu, W. 2023. Swapprompt: Test-time prompt adaptation for vision-language models. *Advances in Neural Information Processing Systems*, 36: 65252–65264.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-grained visual classification of aircraft. arXiv:1306.5151.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, 722–729. IEEE.

Ouali, Y.; Bulat, A.; Matinez, B.; and Tzimiropoulos, G. 2023. Black box few-shot adaptation for vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15534–15546.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, 3498–3505. IEEE.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Silva-Rodriguez, J.; Hajimiri, S.; Ben Ayed, I.; and Dolz, J. 2024. 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*, 23681–23690.

Song, M.; Wang, H.; and Zhong, G. 2024. Self-prompt mechanism for few-shot image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4934–4942.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv:1212.0402.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2021. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *International Conference on Learning Representations*.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 3485–3492. IEEE.

Xue, L.; Shu, M.; Awadalla, A.; Wang, J.; Yan, A.; Purushwalkam, S.; Zhou, H.; Prabhu, V.; Dai, Y.; Ryoo, M. S.; et al. 2024. xgen-mm (blip-3): A family of open large multimodal models. arXiv:2408.08872.

Yang, Z.; Li, L.; Lin, K.; Wang, J.; Lin, C.-C.; Liu, Z.; and Wang, L. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). arXiv:2309.17421.

Yao, H.; Zhang, R.; and Xu, C. 2023. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6757–6767.

Zanella, M.; Gérin, B.; and Ayed, I. 2024. Boosting vision-language models with transduction. *Advances in Neural Information Processing Systems*, 37: 62223–62256.

Zhang, R.; Zhang, W.; Fang, R.; Gao, P.; Li, K.; Dai, J.; Qiao, Y.; and Li, H. 2022. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, 493–510. Springer.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16816–16825.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348.

Zhu, B.; Niu, Y.; Han, Y.; Wu, Y.; and Zhang, H. 2023. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF international conference on computer vision*, 15659–15669.

Zhu, H.; and Koniusz, P. 2023. Transductive few-shot learning with prototype-based label propagation by iterative graph refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 23996–24006.

Ziko, I.; Dolz, J.; Granger, E.; and Ayed, I. B. 2020. Laplacian regularized few-shot learning. In *International conference on machine learning*, 11660–11670. PMLR.