Multi-level Constraint-Based Two-Stage Few-Shot Knowledge Distillation for Vision-Language Models

Yantao Liu
School of Software Engineering, South China University of Technology, Guangzhou 510006, Guangdong, China
E-mail: YantaoLiuu@outlook.com

Keywords: Vision-Language Model, few-shot knowledge distillation, multi-level constraint, two-stage

Received: January 26, 2026

Vision-Language Models (VLMs) exhibit excellent zero-shot and few-shot capabilities in downstream tasks by maximizing the similarity between matched image-text pairs. However, the dual-encoder structure of VLMs introduces a large number of parameters, which limits their practical deployment. Knowledge distillation transfers the knowledge of VLMs to lightweight student models to ensure computational efficiency, but the requirement of sufficient high-quality labeled data for knowledge distillation is difficult to meet in real-world scenarios. This paper proposes a Multi-level Two-stage few-shot Knowledge Distillation (MTKD) method. MTKD consists of two stages: in the first stage, a fine-tuned VLM is used as the teacher model. Under the few-shot setting, the knowledge of unlabeled data is transferred to the student model through multi-level constraints (instance-level, batch-level, and class-level) to enhance the few-shot knowledge representation. In the second stage, a small amount of labeled data is used, the student model from the first stage is frozen, and an adapter implemented by a residual structure is inserted at the end of the image encoder for supervised improvement. Ablation experiments on 6 commonly used public datasets verify the effectiveness of MTKD, and comparisons with other methods demonstrate its competitiveness. MTKD achieves an average performance improvement of 3.2% across the six public datasets, with a maximum gain of 8.6% on certain datasets. In addition, experiments on 3 medical datasets prove that MTKD also has high applicability in the field of medical image recognition, indicating that MTKD can be easily transferred to fields with significant differences from the pre-trained data distribution.

Povzetek: MTKD je metoda dvostopenjskega učenja s prenosom znanja, ki z malo označenimi podatki učinkovito prenese znanje velikih vizualno-jezikovnih modelov na manjše modele ter tako izboljša njihovo natančnost in uporabnost tudi na medicinskih slikah.

1 Introduction

Vision-Language Models (VLMs) [1, 36, 37, 53, 55] are pre-trained on large-scale web-level matched image-text pairs, endowing them with excellent zero-shot and few-shot capabilities. Instead of relying on traditional discrete labels, VLMs utilize text paired with images as category descriptions. During training, the cosine similarity between the feature representations of images and text in positive sample pairs is maximized, thereby achieving high alignment between images and text in the feature space. Full Fine-Tuning (FFT) adapts pre-trained VLMs to downstream tasks by updating all their parameters. However, since FFT fails to retain pre-trained parameters, it is highly prone to overfitting in few-shot scenarios [49]. Additionally, the dual-encoder architecture of VLMs results in a large parameter scale, leading to high computational costs. Parameter Efficient Fine-Tuning (PEFT) [2, 31, 32, 33, 34] enables rapid domain adaptation by freezing the pre-trained backbone of VLMs. It incorporates learnable prompts [3, 4, 5, 6] into the visual and textual inputs or inserts small adapters [7, 8] into the end of the backbone, after which only these few additional parameters are updated. Although PEFT enables effective generalization, its large parameter scale still creates a bottleneck in inference efficiency for practical applications [57, 58].

Knowledge Distillation (KD) [9] constrains the student model by minimizing the deviation between the outputs of the student and teacher models, aligning their category prediction probability distributions. This process transfers the knowledge of the teacher model to a lightweight student model, thereby improving computational efficiency and enabling flexible deployment in environments with limited computing resources [47]. Owing to its superior performance in knowledge transfer and model compression, KD has been introduced into VLMs to reduce their scale and enhance inference efficiency. Existing KD methods for VLMs can be categorized into four types: prediction alignment [10, 11, 14], feature mimicry [10, 12, 13], gradient matching [10], and interactive contrastive learning [10, 11]. These studies typically use a weighted average of the distillation loss and cross-entropy contrastive loss as the total loss function to constrain the training of the student model. Nevertheless, to enable the student model to acquire rich knowledge representations from the teacher model, large-

scale high-quality labeled data is usually required during training, which is difficult to obtain in real-world scenarios [38, 39].

Unsupervised Knowledge Distillation (UKD) aims to reduce reliance on large-scale labeled data. It leverages the intrinsic properties of input data or utilizes predictions generated by a teacher model for the input data to construct pseudo-labels. These pseudo-labels are then used to constrain the training of the student model. For example, SEED [15] utilizes an identity matrix as pseudo-labels. This approach constrains the student model's updates by maximizing the cosine similarity between feature vectors generated by the teacher and student models for the same input. UKD-CMH [16] extends UKD to multimodal models. It uses the distance between each pair of image and text features encoded by the teacher model as pseudo-labels. This enables the student model to mimic the teacher model's predictions regarding the semantic relevance between images and texts. PromptKD [17] investigates UKD for Vision-Language Models (VLMs). This method employs the teacher model to generate pseudo-labels for input images to supervise the student model's training. Concurrently, text features are precomputed and cached, and only the visual prompts of the student model are optimized. Although UKD avoids reliance on high-quality labels, it is still limited by the availability of large-scale data. Furthermore, single instance-level constraints only measure differences in probability distributions and fail to capture sample features or structural information, restricting the sufficient transfer of knowledge.

Based on the above discussion, this paper argues that knowledge distillation for VLMs should be conducted under few-shot data conditions to address the challenge of insufficient samples in real-world scenarios. To this end, this paper proposes a Multi-level Two-stage few-shot Knowledge Distillation (MTKD) method, whose overall structure is shown in Figure 1. MTKD achieves VLM scale compression through unsupervised knowledge distillation with a small number of samples and supervised performance enhancement with an extremely small number of samples. It consists of two stages: Multi-level Unsupervised Prompt Distillation (MUPD) and Adapter-based Supervised Fine-Tuning (ASFT). In the MUPD stage, a small amount of few-shot data is used to transfer the knowledge of the VLM to the student model under unsupervised constraints. Specifically, the category predictions generated by the student and teacher models are constrained to align at three levels: instance-level, batch-level, and class-level. These multi-level constraints eliminate the need for high-quality labeled data, effectively alleviate the problem of incomplete knowledge capture under single instance-level constraints, and mitigate the limitations imposed by insufficient training samples on knowledge transfer from the teacher to the student model [51, 52]. During this stage, the pre-trained parameters of the student model are frozen, and only the learnable prompt parameters are updated. In the second stage (ASFT), all parameters of the student model obtained from the first stage are frozen. An adapter with a residual structure is inserted at the end of the student model, and supervised enhancement training is performed using an extremely small number of samples (fewer than those used in the first stage) [50]. This design avoids reliance on large-scale labeled data and extensive computing resources [48]. The synergistic effect of MUPD and ASFT significantly enhances the performance of MTKD in few-shot scenarios, enabling the student model to more effectively extract and absorb knowledge from the teacher model.

The contributions of this paper are as follows:

A multi-level constraint-based two-stage few-shot knowledge distillation method is proposed, which achieves effective VLM scale compression through unsupervised few-shot knowledge transfer and supervised enhancement with an extremely small number of samples.

Multi-level knowledge distillation (at the instance, batch, and class levels) is employed to alleviate the limitation of single instance-level constraints, which struggle to capture complex features and cross-instance correlations under few-shot settings.

An adapter implemented with a residual structure is used to enhance the task-specific prediction capability of VLMs obtained through few-shot UKD, using only an extremely small number of labeled samples.

MTKD achieves an average performance improvement of 3.2% on 6 public datasets. Furthermore, results on 3 medical datasets demonstrate that the proposed method is also effective in tasks where the data distribution deviates from that of the pre-trained data.

The rest of this paper is organized as follows: Section 2 reviews relevant research on the Parameter-Efficient Fine-Tuning and Knowledge Distillation for VLMs. Section 3 presents a detailed description of the design of the MTKD, including the two-stage distillation framework and the respective roles of each stage. Section 4 evaluates the effectiveness of the proposed method and demonstrates its superiority by comparing it with other approaches. Finally, Section 5 summarizes the findings of this study and future work.

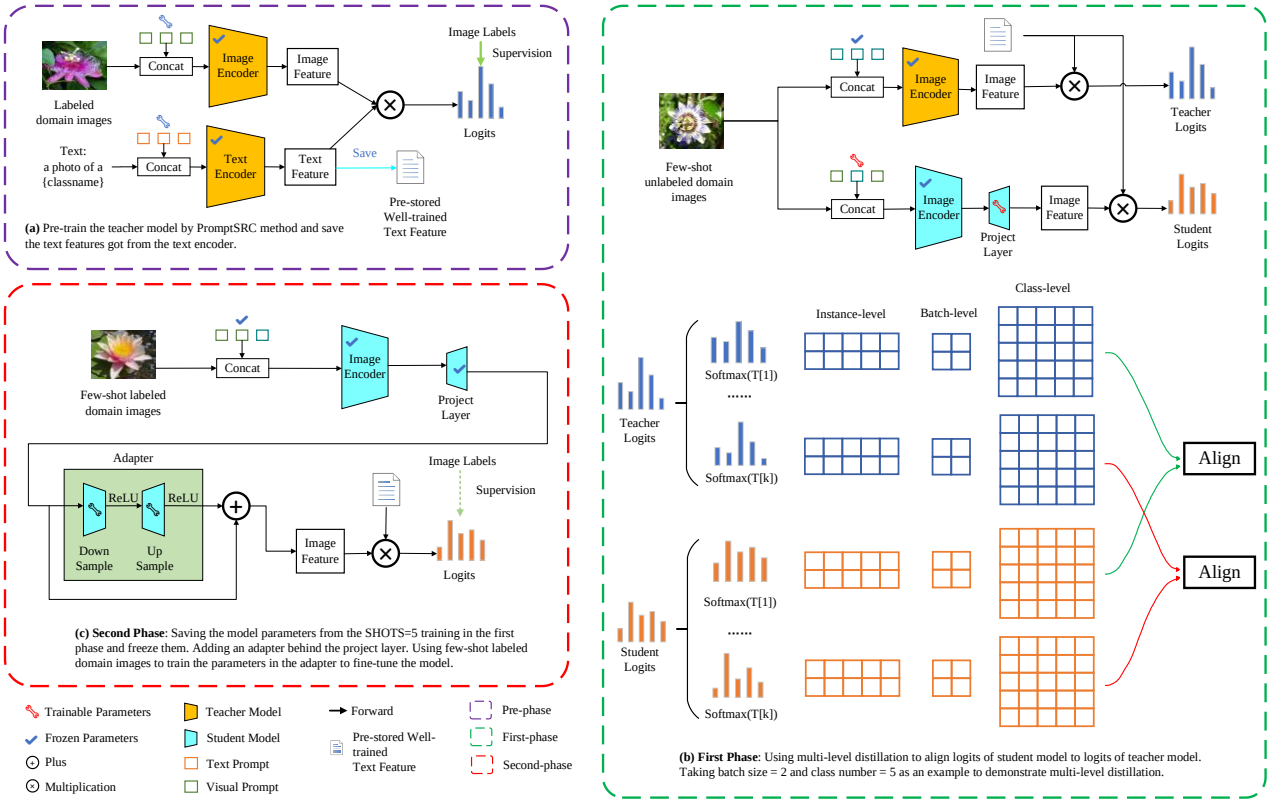

Figure 1. Overview of the MTKD method. (a) Pre-train the teacher model, and pre-compute and store text features through its text encoder. (b) First stage: Under a small number of unlabeled samples, use multiple temperature parameters to align the logits of the student model with those of the teacher model at three levels: instance-level, batch-level, and class-level. (c) Second stage: Freeze the parameters of the student model from the first stage, insert an adapter implemented with a residual structure at the end of its image encoder, and use a tiny number of labeled samples to further improve its performance.

2 Related work

2.1 Parameter-efficient fine-tuning for VLMs

Building on pre-trained models, Parameter-Efficient Fine-Tuning (PEFT) [2, 31, 32, 33, 34] optimizes only a small set of newly added parameters. This enables efficient generalization of VLMs such as CLIP [1] to downstream tasks with a small number of samples and relatively few training iterations. There are two common PEFT approaches for VLMs: Prompt Tuning [31, 32]: This method introduces learnable prompt vectors (with the same dimension as visual and text embeddings, respectively) into the input to guide the model in adapting to new tasks. CoOp [3] was the first to replace manually designed hard prompts with learnable text prompts. To address the limited scalability of fixed text prompts for input images of unseen classes, CoCoOp [4] proposes adding image features encoded by the model to text prompts via a linear layer. This allows the prompts to flexibly adapt to input images. However, introducing prompts solely in the text encoder fails to enable collaboration between the visual and text branches in downstream tasks. In response, MaPLe [5] applies both visual and text prompts and correlates them through a linear projection layer. To retain general knowledge while adapting to specific tasks, PromptSRC [6] not only enforces constraints based on real labels but also restricts the alignment of multi-modal features and prediction results with the corresponding outputs of the original pre-trained VLM. Adapter Tuning [33, 34]: This approach inserts lightweight adapters with a small number of parameters at the end of the model to achieve generalization in specific domains. CLIP-Adapter [7] inserts adapters with residual structures (implemented by two linear layers) at the ends of the image and text encoders. It facilitates the model's adaptation to a single downstream task. To enhance cross-task interaction, VMT-Adapter [40] trains adapters that share knowledge from multiple tasks. It inserts an adapter in parallel with the linear layer at the end of the feature encoder. It adds the adapter's output to the general features, and feeds the combined result into the decoder of each specific task. Considering that introducing additional trainable parameters increases computational costs, Tip-Adapter [8] constructs a cache using key-value pairs composed of image features encoded by the pre-trained VLM and the real labels of images. This cache is directly added to the end of the image encoder as an adapter. Compared with Full Fine-Tuning (FFT), PEFT only trains an extremely small set of parameters, which significantly reduces the risk of overfitting (easily induced under few-shot settings) and the demand for computing resources.

Although PEFT achieves effective generalization and reduces training costs, the excessively large parameter scale of VLMs means that deploying PEFT-modified VLMs to edge devices requires substantial storage resources and leads to inference efficiency issues. To address this, this paper proposes a two-stage few-shot knowledge distillation method. It realizes VLM compression through few-shot knowledge transfer to ensure inference efficiency and reduce the demand for storage resources.

2.2 Knowledge distillation

2.2.1 Knowledge distillation

Knowledge Distillation (KD) [9, 19, 20, 24] aims to transfer knowledge from a teacher model to a lightweight student model. While maintaining model performance, it reduces storage overhead and improves inference efficiency [54, 56]. Common KD methods involve aligning the student model with the teacher model in three aspects: the Logits of the output layer [20, 21, 22, 23], hidden layer features [24, 25, 26], and the relationships between predictions generated for different input images [18, 19, 27, 28]. KD enables the student model to not only learn the teacher model's ability to predict domain-specific images but also mimic the teacher model's structural features. As a result, the compressed student model can achieve performance comparable to that of the teacher model.

2.2.2 Knowledge distillation for VLMs

Given its ability to compress models while preserving performance, knowledge distillation is also applied to the lightweighting of VLMs. Existing methods can be categorized into four types: Direct Alignment [10, 11, 14]: This type of method uses the teacher model's category prediction probability distribution to supervise the update of the student model.

Feature Alignment [10, 12, 13]: To address the problem of incomplete knowledge representation caused by only applying instance-level constraints on the output layer, aligning the feature vectors (encoded from images and text) of the teacher and student models effectively enhances knowledge transfer.

Gradient Mimicry [10]: Unlike the first two types that only align output results, this method aligns the gradients of the contrastive loss function (with respect to image and text features) between the teacher and student models. This ensures that the student model follows the same learning trajectory as the teacher model.

Interactive Contrastive Alignment [11]: Considering the interaction between the teacher and student encoders, this method uses contrastive loss to constrain the alignment of image and text features across the teacher and student models.

However, these methods still rely on large-scale labeled data, which is difficult to obtain in practical applications. To solve this problem, Unsupervised Knowledge Distillation (UKD) [15, 16] proposes using the teacher model to generate pseudo-labels for input data to constrain the training of the student model, eliminating the need for real labels.

In the context of VLMs, PromptKD [17] activates only prompt vectors as the medium for distillation and shares the text features computed by the teacher model with the student model. This reduces computational overhead and simplifies the distillation process.

While UKD avoids reliance on data labels, it still requires a large amount of training data for support. To address this limitation, the MTKD method proposed in this paper enables the student model to acquire rich knowledge representations from the teacher model through multi-level constraints under unlabeled few-shot data conditions. This effectively reduces the reliance of VLM knowledge distillation on large-scale datasets.

Table 1 compares the MTKD method proposed in this paper with representative methods in related work. Among the related methods, CoOp and CoCoOp only use text prompts for fine-tuning. MaPLe and PromptSRC simultaneously leverage both text prompts and visual prompts. CLIP-Adapter, on the other hand, inserts adapters with residual structures at the end of the encoders. All the above methods are evaluated under a 16-shot setting. PromptKD uses prompts for knowledge distillation. It employs only a single instance-level constraint for distillation. Its experiments are conducted with full training samples, relying on a large amount of data. Our method proposed in this paper is a two-stage distillation framework. In the first stage, instance-level, batch-level, and class-level constraints are introduced in few-shot unsupervised distillation. It enables the student model to effectively learn knowledge representations with a small number of samples. In the second stage, an adapter is employed for supervised fine-tuning with a minimal number of samples. It aims to correct potential prediction biases of the teacher model when dealing with limited samples. This method requires only a 5-shot setting, significantly reducing the demand for training data compared to existing methods. Through multi-level constraints and two-stage optimization, MTKD achieves efficient knowledge transfer with extremely few samples.

Table 1: Comparison of design differences with related methods

Method	Use Text Prompt	Use Visual Prompt	Use Adapter	Shot	Knowledge Distillation	
					Use or not	Constraints
CoOp	√	×	×	16	×	\
CoCoOp	√	×	×	16	×	\
MaPLe	√	√	×	16	×	\
PromptSRC	√	√	×	16	×	\
CLIP-Adapter	×	×	√	16	×	\
PromptKD	×	√	×	full	√	Instance
Ours	×	√	√	5	√	Instance & Batch & Class

3 Method

3.1 KD in VLM

VLMs [1, 36] replace discrete labels with natural language texts that describe image categories. They are trained on large-scale matched image-text pairs and possess excellent zero-shot and few-shot reasoning capabilities. Since this paper uses CLIP to implement MTKD, we first review CLIP and introduce its basic KD process.

CLIP consists of an image encoder f_I and a text encoder f_T. For a labeled dataset $D=\left\{\{image_n,label_n\}_{n=1}^{N},\{class_c\}_{c=1}^{C}\right\}$, each image $image_n$ corresponds to a category label $label_n$, and each category has a text description $class_c$. The set $\{image_n\}_{n=1}^{N}$ is encoded by f_I to obtain normalized image features $\{I_n\}_{n=1}^{N}$, where $I_n=f_I(image_n)/\|f_I(image_n)\|\in R^d$, and d is the feature dimension. The set $\{class_c\}_{c=1}^{C}$ uses the template"a photo of a $\{class_c\}$ " to generate text descriptions $\{text_c\}_{c=1}^{C}$. f_T encodes $\{text_c\}_{c=1}^{C}$ into normalized text features $\{T_c\}_{c=1}^{C}$, where $T_c=f_T(text_c)/\|f_T(text_c)\|\in R^d$.

During training, a batch of matched image-text pairs $\{image_b,text_b\}_{b=1}^{B}$ is selected each time, and the training is constrained using the cross-entropy contrastive loss:

$$L_{CLIP}=-\frac{1}{2}\left(\sum_{b=1}^{B}\log\frac{\exp(I_b\cdot T_b)}{\sum_{t=1}^{B}\exp(I_b\cdot T_t)}+\sum_{b=1}^{B}\log\frac{\exp(I_b\cdot T_b)}{\sum_{t=1}^{B}\exp(I_t\cdot T_b)}\right) \quad (1)$$

Where, $\{I_b\}_{b=1}^{B}$ and $\{T_b\}_{b=1}^{B}$ are the image and text features (encoded from the batch of images and texts, respectively). CLIP maximizes the cosine similarity between the image and text features of positive sample pairs, thereby aligning the multi-modal feature space and ensuring that input images are correctly matched with their corresponding text descriptions. The category prediction probability distribution generated by the model for an input image is obtained using the softmax function:

$$p_\gamma=\text{softmax}(I\cdot T_\gamma)=\frac{\exp(I\cdot T_\gamma)}{\sum_{c=1}^{C}\exp(I\cdot T_c)},\gamma\in\{1,2,\ldots,C\} \quad (2)$$

where I is the encoded feature of the input image, and $\{T_c\}_{c=1}^{C}$) are the text features of C categories.

Knowledge distillation aims to extract knowledge from a large-scale teacher model to a lightweight student model by minimizing the deviation between the outputs generated by the teacher and student models. This reduces storage overhead and improves inference efficiency, allowing the student model to be flexibly deployed in resource-constrained application environments.

For VLMs, a common KD method uses the Kullback-Leibler (KL) divergence to align the category prediction probability distributions p_{tea} and p_{stu} of the teacher and student models. The loss function is given by the following formula:

$$L_{KD}=(1-\lambda)\cdot CE(p_{stu},labels)+\lambda\cdot D_{KL}(p_{tea}/\!/p_{stu}) \quad (3)$$

where λ is the weight of the distillation loss, labels are the labels of the input images, $CE(\cdot)$ is the cross-entropy loss function, and $D_{KL}(\cdot)$ is the KL divergence function. Knowledge distillation achieves effective knowledge transfer from the teacher model to the student model.

3.2 Overall architecture

The MTKD method proposed in this paper consists of two stages: MUPD and ASFT. The teacher model is obtained through few-shot supervised fine-tuning. Specifically, the text encoder f_T^{tea} of the trained teacher model pre-computes and stores normalized text features $T_{tea}\in R^{C\times d_{tea}}$ for category description texts. Here, C is the number of categories, and d_{tea} is the dimension of the encoded features of the teacher model. As shown in Figure 1(a), T_{tea} is shared between the teacher and the student.

In the first stage, a small amount of unlabeled data is used to align the probability distribution generated by the student model for input images with that of the teacher model at three levels: instance-level, batch-level, and class-level. Details are shown in Figure 1(b).

In the second stage, the parameters of the student model from the first stage are frozen. An adapter with a residual structure is inserted at the end of the student model, and supervised fine-tuning is performed with a tiny number of samples, as shown in Figure 1(c).

Sections 3.3 and 3.4 will introduce the two stages of MTKD in detail, respectively.

3.3 Multi-level unsupervised prompt distillation (MUPD)

Unlike the large amount of labeled data $D=\left\{\{image_n,label_n\}_{n=1}^{N},\{class_c\}_{c=1}^{C}\right\}$ used in previous vision-language models, this stage aims to utilize a few-shot unlabeled dataset $D_u=\{image_m\}_{m=1}^{M}$, where $M\ll N$. To transfer the knowledge of the large-scale teacher model to the lightweight student model under the constraint of few-shot data, MUPD makes full use of the Logits of the teacher and student models. It generates multiple probability distributions, and aligns them at three levels (instance-level, batch-level, and class-level) one by one.

Compared with aligning only a single probability distribution generated by the student and teacher models under instance-level constraints, MUPD, through multi-level constraints, enables the student model to comprehensively learn the knowledge representation of the teacher model using few-shot data.

Since the pre-stored text features T_{tea} are shared between the teacher and the student, the student model only updates the visual prompts, thus significantly reducing the computational overhead during training. A projection layer proj needs to be inserted at the end of the image encoder to align the differences in feature dimensions between the teacher and the student. Specifically, the image encoder f_I^{stu} of the student model is frozen, and only the visual prompts and the projection layer are activated as the distillation medium.

For each batch of input images $\{\text{image}_b\}_{b=1}^{B}$, normalized image features f_I^{tea} and f_I^{stu} are obtained by passing through the image encoders I_{tea} and I_{stu} of the teacher model and the student model, respectively. T_{tea} is used to compute prediction Logits for the input images:

$$z_{tea}=I_{tea}\times T_{tea}^{T} \quad (4)$$

$$z_{stu}=I_{stu}\times T_{tea}^{T} \quad (5)$$

here z_{tea}, $z_{stu}\in R^{B\times C}$, and B is the batch size.

Subsequently, K temperature parameters $\{\tau_k\}_{k=1}^{K}$ are used to pass z_{tea} and z_{stu} through the softmax function, respectively, to obtain K category prediction probability distributions $\{p_{tea}^{(k)}\}_{k=1}^{K}$ 与$\{p_{stu}^{(k)}\}_{k=1}^{K}$. Among them:

$$p_{tea}^{(k)}[i,j]=\text{softmax}(z_{tea}[i,j]/\tau_k)=\frac{\exp(z_{tea}[i,j]/\tau_k)}{\sum_{c=1}^{C}\exp(z_{tea}[i,c]/\tau_k)} \quad (6)$$

$$p_{stu}^{(k)}[i,j]=\text{softmax}(z_{stu}[i,j]/\tau_k)=\frac{\exp(z_{stu}[i,j]/\tau_k)}{\sum_{c=1}^{C}\exp(z_{stu}[i,c]/\tau_k)} \quad (7)$$

Let $z\in R^{B\times C}$ denote Logits, where $z[i,j]=\max_{1\le c\le C} z[i,c]$, τ_α and τ_β are temperature parameters, and $\tau_\alpha<\tau_\beta$. Then

$$\frac{e^{\frac{z[i,j]}{\tau_\alpha}}}{\sum_{c=1}^{C}e^{\frac{z[i,c]}{\tau_\alpha}}}=\frac{e^{\frac{z[i,j]}{\tau_\alpha}}\cdot e^{z[i,j]\cdot\left(\frac{1}{\tau_\beta}-\frac{1}{\tau_\alpha}\right)}}{\sum_{c=1}^{C}e^{\frac{z[i,c]}{\tau_\alpha}}\cdot e^{z[i,j]\cdot\left(\frac{1}{\tau_\beta}-\frac{1}{\tau_\alpha}\right)}}=\frac{e^{\frac{z[i,j]}{\tau_\beta}}}{\sum_{c=1}^{C}e^{\frac{z[i,c]}{\tau_\alpha}-\frac{z[i,j]}{\tau_\beta}+\frac{z[i,j]}{\tau_\alpha}}}=\frac{e^{\frac{z[i,j]}{\tau_\beta}}}{\sum_{c=1}^{C}e^{\frac{z[i,c]}{\tau_\alpha}-\frac{z[i,j]}{\tau_\beta}+\frac{z[i,c]}{\tau_\beta}}}$$

$$=\frac{e^{\frac{z[i,j]}{\tau_\beta}}}{\sum_{c=1}^{C}e^{(z[i,j]-z[i,c])\cdot\left(\frac{1}{\tau_\beta}-\frac{1}{\tau_\alpha}\right)+\frac{z[i,c]}{\tau_\beta}}}>\frac{e^{\frac{z[i,j]}{\tau_\beta}}}{\sum_{c=1}^{C}e^{\frac{z[i,c]}{\tau_\beta}}}$$

It can thus be seen that with a lower temperature parameter, the probability distribution has a larger maximum probability value, which results in a sharper probability distribution. This distinctly reflects the model's decision-making and reduces the interference from secondary categories. With a higher temperature parameter, the probability distribution has a smaller maximum probability value. This can generate a smoother probability distribution. It enables the student model to learn richer semantic information such as the relative relationships between categories. This helps to enhance its generalization ability. Multiple temperature parameters expand the Logits into multiple category prediction probability distributions with different degrees of sharpness. It allows the student model to obtain rich knowledge representations from the Logits of the teacher model, so as to achieve comprehensive and sufficient knowledge transfer.

Then, for each temperature parameter τ_k($k\in\{1,2,...,K\}$), the probability distribution $p_{stu}^{(k)}$ of the student model and the probability distribution $p_{tea}^{(k)}$ of the teacher model are aligned at the following three levels:

Instance-level: The KL divergence loss is used to align the probability distributions generated by the student and the teacher:

$$L_{ins}^{(k)}=D_{KL}\left(p_{tea}^{(k)}\,//\,p_{stu}^{(k)}\right)=\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{C}p_{tea}^{(k)}[i,j]\cdot\log\left(\frac{p_{tea}^{(k)}[i,j]}{p_{stu}^{(k)}[i,j]}\right) \quad (8)$$

For the convex function f(x)=-log x, according to Jensen's inequality:

$$f\left(\sum_i\lambda_i x_i\right)\le\sum_i\lambda_i f(x_i)$$

We get

$$0=-\log 1=f(1)=f\left(\sum_{j=1}^{C}p_{stu}^{(k)}[i,j]\right)=f\left(\sum_{j=1}^{C}p_{tea}^{(k)}[i,j]\cdot\frac{p_{stu}^{(k)}[i,j]}{p_{tea}^{(k)}[i,j]}\right)\le\sum_{j=1}^{C}p_{tea}^{(k)}[i,j]\cdot f\left(\frac{p_{stu}^{(k)}[i,j]}{p_{tea}^{(k)}[i,j]}\right)$$

$$=\sum_{j=1}^{C}p_{tea}^{(k)}[i,j]\cdot\left(-\log\left(\frac{p_{stu}^{(k)}[i,j]}{p_{tea}^{(k)}[i,j]}\right)\right)=\sum_{j=1}^{C}p_{tea}^{(k)}[i,j]\cdot\log\left(\frac{p_{tea}^{(k)}[i,j]}{p_{stu}^{(k)}[i,j]}\right)$$

Thus, $L_{ins}^{(k)}\ge 0$. If and only if $p_{tea}^{(k)}=p_{stu}^{(k)}$, $L_{ins}^{(k)}=0$. Therefore, minimizing the instance-level loss $L_{ins}^{(k)}$ can approximate the probability distributions generated by the student and teacher models, thereby enabling the student model to learn the teacher model's ability to predict domain-specific images. Under multiple temperature parameters, the student model can simultaneously learn the deterministic decisions of the teacher model from the multiple probability distributions generated by the teacher model, as well as the associations between the input image and other secondary categories.

Batch-level: Multiply the probability distribution generated by the model by its own transpose to compute the Gram matrix:

$$G_{tea}^{(k)}=p_{tea}^{(k)}\times p_{tea}^{(k)^{T}} \quad (9)$$

$$G_{stu}^{(k)}=p_{stu}^{(k)}\times p_{stu}^{(k)^{T}} \quad (10)$$

$G_{tea}^{(k)}$, $G_{stu}^{(k)}\in R^{B\times B}$, where B is the batch size. Among them

$$G_{tea}^{(k)}[i,j]=\sum_{c=1}^{C}p_{tea}^{(k)}[i,c]\cdot p_{tea}^{(k)}[j,c] \quad (11)$$

$$G_{stu}^{(k)}[i,j]=\sum_{c=1}^{C}p_{stu}^{(k)}[i,c]\cdot p_{stu}^{(k)}[j,c] \quad (12)$$

From (11) and (12), it can be seen that $G_{tea}^{(k)}[i,j]$ and $G_{stu}^{(k)}[i,j]$ are the similarities between the probability distributions generated by the teacher model and the student model for the i-th image and the j-th image, respectively. Therefore, the Gram matrices $G_{tea}^{(k)}$ and $G_{stu}^{(k)}$ store the degrees of association between the predictions of the teacher and the student for any two input images. The input correlation between the student and the teacher is aligned using the following loss function:

$$L_{batch}^{(k)}=\frac{1}{B}\|G_{tea}^{(k)}-G_{stu}^{(k)}\|^2=\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{B}\left(G_{tea}^{(k)}[i,j]-G_{stu}^{(k)}[i,j]\right)^2 \quad (13)$$

The batch-level loss enables the student model to extract rich high-level semantic information from the teacher model by imitating the teacher model's prediction of input correlation.

Class-level: Multiply the transpose of the probability distribution generated by the model by itself to obtain the Class Affinity matrix:

$$A_{tea}^{(k)}=p_{tea}^{(k)^{T}}\times p_{tea}^{(k)} \quad (14)$$

$$A_{stu}^{(k)}=p_{stu}^{(k)^{T}}\times p_{stu}^{(k)} \quad (15)$$

$A_{tea}^{(k)}$, $A_{stu}^{(k)} \in R^{C\times C}$, where C is the number of categories. Among them

$$A_{tea}^{(k)}[i,j]=\sum_{b=1}^{B} p_{tea}^{(k)}[b,i]\cdot p_{tea}^{(k)}[b,j] \quad (16)$$

$$A_{stu}^{(k)}[i,j]=\sum_{b=1}^{B} p_{stu}^{(k)}[b,i]\cdot p_{stu}^{(k)}[b,j] \quad (17)$$

From (16) and (17), it can be seen that $A_{tea}^{(k)}[i,j]$ and $A_{stu}^{(k)}[i,j]$ are the similarities between the predictions of the teacher model and the student model for the i-th category and the j-th category, respectively. Therefore, the class affinity matrices $A_{tea}^{(k)}$ and $A_{stu}^{(k)}$ store the degrees of association between the predictions of the teacher and the student for any two categories, respectively. The class correlation between the student and the teacher is aligned using the following loss function:

$$L_{class}^{(k)}=\frac{1}{C}\|A_{tea}^{(k)}-A_{stu}^{(k)}\|^2=\frac{1}{C}\sum_{i=1}^{C}\sum_{j=1}^{C}\left(A_{tea}^{(k)}[i,j]-A_{stu}^{(k)}[i,j]\right)^2 \quad (18)$$

The class-level loss enables the student model to learn the structured features of the teacher model from a macro perspective by imitating the teacher model's prediction of class correlation. Finally, the loss function L_{MUPD} of the first stage is obtained by the following formula:

$$L_{MUPD}=\sum_{k=1}^{K}\left(L_{ins}^{(k)}+L_{batch}^{(k)}+L_{class}^{(k)}\right) \quad (19)$$

Each of these loss terms is non-negative, and all losses vanish simultaneously if and only if $\boldsymbol{p_{stu}^{(k)}=p_{tea}^{(k)}}$. Therefore, minimizing $\boldsymbol{\mathcal{L}_{MUPD}}$ means requiring the student model to fully imitate not only the predictive distributions but also the structural characteristics of the teacher model under each temperature parameter. According to formula (19), the student model imitates the teacher model in three aspects: instance-level prediction, input correlation, and class correlation. Thus, it captures the rich knowledge representation in the teacher model under the few-shot setting.

3.4 Adapter-based supervised fine-tuning (ASFT)

Unsupervised knowledge distillation does not rely on data labels at all, so the performance of the student model is limited by the prediction accuracy of the teacher model. Different from the few-shot unlabeled dataset $D_u=\{image_m\}_{m=1}^{M}$ used in MUPD, this stage uses a tiny labeled dataset $D_l=\{image_{m'},label_{m'}\}_{m'=1}^{M'}$ to further fine-tune the student model, where $M'<M$. ASFT aims to further improve the prediction ability of the student model through real data labels, targeting the situation where the teacher model has a few inaccurate predictions. A direct approach is to fine-tune the learnable parameters in the first stage. However, this method is prone to overfitting the training data under few-shot settings, and the learnable prompts located at the input layer bring high computational overhead.

In this stage, the visual prompts and projection layer of the student model trained by MUPD are frozen. An adapter with a residual structure adapter(·)=$f_{up}(f_{down}(\cdot))$ is inserted at the end of the image encoder of the student model for fine-tuning. The adapter consists of a downsampling linear layer f_{down} and an upsampling linear layer f_{up}. It retains the learnable prompts and parameters in the projection layer obtained from MUPD distillation, thereby reducing the risk of overfitting; the activated adapter is located at the output end, resulting in low gradient calculation complexity. For the image feature I_{stu} encoded by the student model in the first stage, after passing through the adapter, $I_{stu}^{'}$ is obtained:

$$I'_{stu}=adapter(I_{stu})=(1-\alpha)\cdot I_{stu}+\alpha\cdot f_{up}(f_{down}(I_{stu})) \quad (20)$$

Among them, $\alpha \in (0,1)$ is the residual ratio. The value of the residual ratio $\boldsymbol{\alpha}$ has a significant impact on model performance. When $\boldsymbol{\alpha}$ is relatively high, the adapter dominates feature transformation, which helps the model learn task-specific feature representations more thoroughly from a small number of labeled samples. However, an excessively high α also leads to issues. Since the adapter weights lack pre-training initialization, the adapter is highly prone to overfitting the training data when only very few labeled samples are available. When $\boldsymbol{\alpha}$ is relatively low, the original features are largely preserved, which reduces the risk of overfitting and retains the knowledge acquired in the first stage. Yet, an overly low $\boldsymbol{\alpha}$ also restricts the adapter's ability to correct prediction biases from the teacher model, preventing the model from making full use of the supervisory information. Therefore, choosing an appropriate $\boldsymbol{\alpha}$ requires a balance between preserving the acquired knowledge and adapting to the new task.

The Logits z'_{stu} are still calculated using the pre-stored T_{tea}.

$$z'_{stu}=I'_{stu}\times T_{tea}^{T} \quad (21)$$

Align the prediction results of the student model with the ground-truth labels of input images using cross-entropy loss:

$$\mathcal{L}_{ASFT}=\mathrm{CE}(z'_{stu},\ label_{1\ldots B})=-\sum_{b=1}^{B}\log\left(\mathrm{softmax}(z'_{stu}[b,cls_b])\right)=-\sum_{b=1}^{B}\log\left(\frac{\exp(z'_{stu}[b,cls_b])}{\sum_{c=1}^{C}\exp(z'_{stu}[b,c])}\right) \quad (22)$$

where denotes the class index corresponding to the - th input image in the respective batch. Finally, the image encoder of the student model consists of pre-trained weights, visual prompts, a projection layer, and an adapter module. Together with the text encoder of the teacher model, it serves as the compressed Vision-Language Model (VLM).

4 Experiments

4.1 Datasets

This paper evaluates the performance of the MTKD method on 6 natural image datasets and 3 medical image datasets. Specifically, the natural image datasets include DTD [41] for texture classification, OxfordPets [42], Flowers102 [43], and FGVC-Aircraft [44] for fine-grained classification, Caltech101 [45] for general object classification, and EuroSAT [46] for satellite image

recognition. The medical image datasets are used to test the effectiveness of this method in tasks deviating from the pre-trained data, including the breast cancer dataset BACH, the brain tumor dataset Brain, and the diabetic retinopathy dataset EyeDR. The selection of the above datasets is not only highly diverse in terms of task types but also challenging in terms of image characteristics and classification difficulty. In choosing these datasets, this paper has balanced breadth, complexity, and domain diversity, aiming to comprehensively validate the applicability and generalization potential of MTKD.

4.2 Experimental settings

4.2.1 Base-to-novel

Following previous works [3, 4, 5, 6, 17], the experiments partition each dataset into base classes and novel classes. The training set contains only data of base classes, while the test set includes data of both base and novel classes.

In the first-stage experiments of this paper, models are trained under 1-shot, 3-shot, and 5-shot settings respectively, to evaluate the improvement effect of the proposed MUPD method in low-resource scenarios. Subsequently, the model trained under the 5-shot setting in the first stage is adopted, its parameters are frozen, and the performance improvement brought by the second stage is evaluated under 1-shot, 3-shot, and 5-shot conditions. The evaluation metrics for both stages are the performances of models on base class and novel class data under different shots, and the harmonic mean (HM) of the prediction accuracies of the model on base class and novel class data is calculated.

4.2.2 Implementation details

The hardware and software configurations for the experiments are as follows: Model training and evaluation were conducted using the PyTorch 2.5.1 deep learning framework, with Python 3.12 as the programming language, Ubuntu 22.04 as the operating system, and CUDA 12.4 as the parallel computing platform. For the experiments, ViT-L/14 is used as the teacher model, and ViT-B/16 is used as the student model. The teacher model is trained using the PromptSRC [6] method. For learnable prompts, this method adopts the same settings as [6, 17], setting the depth of prompts for both the visual branch and the text branch to 9, and the length of prompts to 4. The input of the text encoder is constructed using"a photo of a {class}". The model is trained for 20 epochs, and the learning rate is decayed using the cosine annealing method. By adjusting the learning rate periodically, the model can both fine-tune parameters and get out of local optima during training. The length of the Warm-up stage is one epoch, and the value of its learning rate is set to 1e-5. We refer to the work of [18] and set the temperature parameter τ to the range $\{1, 2, 3, 5, 6\}$ to systematically investigate its impact on the distillation effectiveness.

4.3 Ablation experiments

This subsection conducts ablation experiments on different components of the method to verify its effectiveness.

4.3.1 Multi-level unsupervised prompt distillation

Different Distillation Constraint Methods

In the first stage, this paper aligns the probability distributions generated by the teacher and student models at three levels: instance-level, batch-level, and class-level under multiple temperature parameters. To demonstrate the effectiveness of the multi-temperature parameter and multi-level constraint strategy, experiments compare the impacts of different distillation constraint methods on the performance of the student model, as shown in Figure 2."Baseline" represents the case of using only the instance-level constraint under a single temperature parameter setting."Ins" represents the case of using only the instance-level constraint under the multi-temperature parameter setting."Ins & Batch" represents the case of using both instance-level and batch-level constraints under the multi-temperature parameter setting."Ins & Batch & Class" represents the case of using instance-level, batch-level, and class-level constraints simultaneously under the multi-temperature parameter setting. It can be seen from Figure 2 that"Ins" has a higher prediction accuracy on the test set than"Baseline", which indicates that the student model acquires more sufficient information including model decisions and inter-class relationships from probability distributions with different degrees of sharpness. "Ins & Batch" generally achieves higher accuracy compared to "Ins", indicating that introducing batch-level constraints helps the student model capture relationships among input samples and thus learn richer knowledge representations. Furthermore, "Ins & Batch & Class" outperforms "Ins & Batch" in terms of accuracy, suggesting that the incorporation of class-level constraints encourages the model to learn semantic associations between categories, thereby obtaining more discriminative representational capabilities. Experiments prove that the distillation strategy of multi-temperature parameters and multi-level constraints adopted by MTKD plays a key role in the final performance.

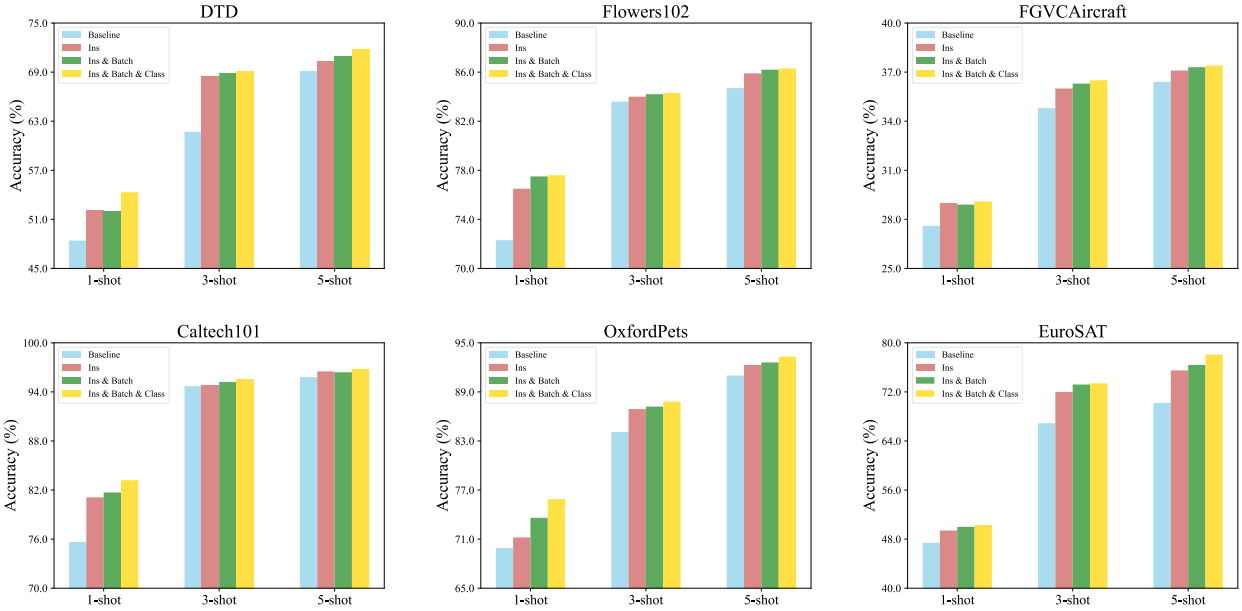

Figure 2: Impact of distillation methods with different level constraints in the first stage on the performance (HM) of the student model.

Teacher models of different sizes

In the first stage, this paper adopts an unsupervised distillation strategy, that is, the output of the student model completely imitates the teacher model without being interfered by data labels. Therefore, the performance of the teacher model is a key factor affecting the distillation effect. Theoretically, a smaller-sized teacher model has higher efficiency, so the knowledge distillation in the first stage has lower training complexity. However, the reduction in the size of the teacher model will also lead to a decline in its performance. At this time, 4 datasets including DTD, Flowers102, OxfordPets, and Caltech101 are used to verify the impact of the size of the teacher model on the distillation performance in the first stage. The student model uses ViT-B/16 by default. As shown in Figure 3, in the above 4 datasets, as the size of the teacher model decreases, the prediction accuracy of the student model distilled in the MUPD stage on the test set decreases significantly. On the 4 datasets, the distillation performance of the teacher model using ViT-B/16 is 3.9% higher on average than that using ViT-B/32, and the distillation performance using ViT-L/14 is 8.3% higher on average than that using ViT-B/16. The results show that a larger-sized teacher model will bring significantly better distillation performance. Therefore, under the condition that computing resources permit, a larger-sized teacher model should be selected to make the student model obtained by distillation have higher prediction ability.

Student models of different sizes

MTKD effectively alleviates the lightweight problem of VLM under few-shot settings. Although a smaller-sized student model is more likely to adapt to application environments with limited storage and computing resources, it will also affect the distillation performance at the same time. Therefore, evaluation is needed to select different student models according to tasks. At this time, 4 datasets including DTD, Flowers102, OxfordPets, and Caltech101 are used to verify the impact of the size of the student model on the distillation performance in the first stage. The teacher model uses ViT-L/14 by default. As shown in Figure 4, in the above 4 datasets, as the size of the student model decreases, the prediction accuracy of the student model distilled in the MUPD stage on the test set decreases significantly. On the 4 datasets, the distillation performance of the student model using ViT-B/16 is 2.4% lower on average than that using ViT-L/14, and the distillation performance using ViT-B/32 is 4.9% lower on average than that using ViT-B/16. The results show that the reduction in the size of the student model will have a significant impact on the distillation performance in the first stage. Therefore, the selection of the student model needs to fully consider factors such as task requirements, task complexity, and deployment environment, and avoid choosing a student model that is too small, which leads to performance degradation.

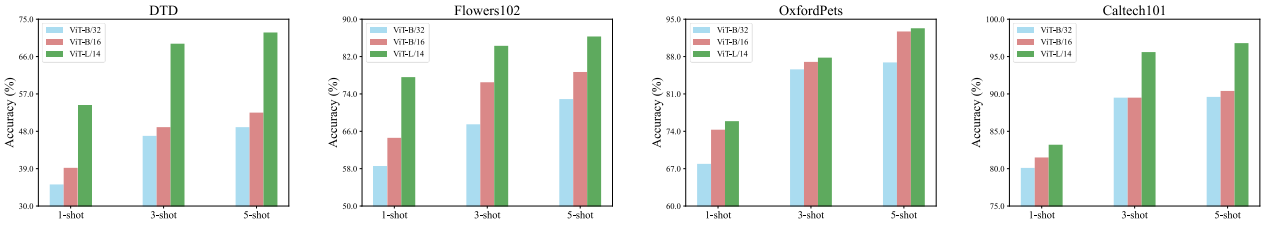

Figure 3: Impact of teacher models of different sizes on the distillation performance in the first stage (HM).

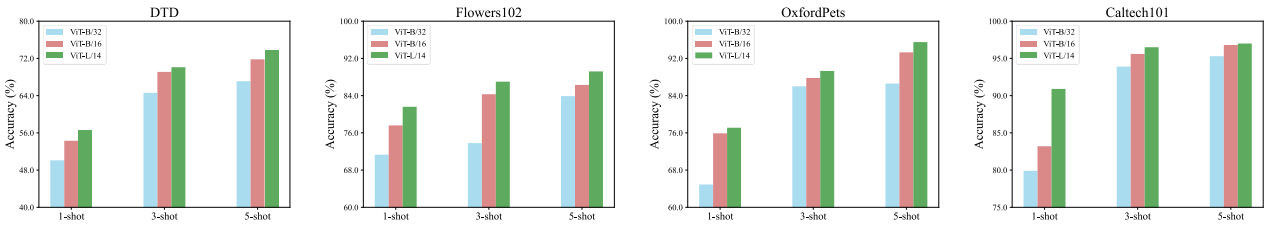

Figure 4: Impact of student models of different sizes on the distillation performance in the first stage (HM).

Performance contribution of learnable prompts

MTKD freezes the pre-trained backbone of the student model and originally conducts the unsupervised knowledge distillation in the first stage by activating learnable prompts. However, since MTKD shares the text features encoded by the teacher model with the student model, a projection layer needs to be introduced at the end of the image encoder of the student model to align the dimensions of image and text features. Therefore, when the student model does not introduce learnable visual prompts, only the projection layer can be used as the activated parameter to participate in distillation. This subsection verifies the impact of learnable prompts on the performance of the student model obtained by distillation in the first stage, and the results are shown in Table 2."Project" means not introducing learnable prompts and only using the projection layer for distillation."Prompt & Project" means using both learnable prompts and the projection layer. Table 2 shows that on 6 natural image datasets, the student model achieves better prediction accuracy when introducing learnable prompts. One possible reason is that, compared with "Project", learnable prompts will act on the attention mechanism of the image encoder, so that the student model can fully learn global correlations and establish long-range dependencies, so as to improve the understanding ability of the semantic level of images in specific domains.

Table 2: Impact of learnable prompts on the distillation performance in the first stage (HM)

DTD			Flowers102			FGVCAircraft		
shot	Project	Prompt & Project	shot	Project	Prompt & Project	shot	Project	Prompt & Project
1-shot	50.2	54.3	1-shot	75.2	77.6	1-shot	28.6	29.1
3-shot	63.8	69.1	3-shot	82.8	84.3	3-shot	34.3	36.5
5-shot	68.6	71.8	5-shot	85.7	86.3	5-shot	36.7	37.4
Caltech101			OxfordPets			EuroSAT		
shot	Project	Prompt & Project	shot	Project	Prompt & Project	shot	Project	Prompt & Project
1-shot	81.3	83.2	1-shot	70.8	75.9	1-shot	48.3	50.3
3-shot	93.6	95.6	3-shot	83.2	87.8	3-shot	72.5	73.4
5-shot	94.2	96.8	5-shot	89.1	93.3	5-shot	74.8	78.1

4.3.2 Adapter-based supervised fine-tuning (ASFT)

To verify the effectiveness of the second stage of the proposed method, the experiment saves the model parameters obtained from distillation under the 5-shot setting in the first stage. Experiments for the second stage are conducted under 1-shot, 3-shot, and 5-shot settings respectively. We compare the results with those under the 5-shot setting in the first stage, as shown in Table 3. The results indicate that on all 6 datasets, the prediction accuracy of the student model trained in the second stage under different shot settings is higher than that trained under the 5-shot setting in the first stage. Since the first stage of MTKD adopts an unsupervised distillation strategy, the output of the student model completely

imitates the teacher model. It directly causes the student model to inherit some errors of the teacher model. By aligning the output of the student model with the real labels of the data, ASFT corrects some incorrect prediction cases, thereby further improving the performance of the student model.

To confirm that the bilinear-layer adapter with a residual structure in the second stage is an advanced method for enhancing the performance of the student model, the experiment compares different supervised fine-tuning methods, as shown in Figure 5. "Prompt & Project" refers to fine-tuning the parameters in the visual prompts and projection layer trained by MUPD. "Linear" refers to fine-tuning using an adapter with a residual structure composed of only one linear layer. "Adapter" refers to fine-tuning using an adapter with a residual structure composed of a downsampling linear layer and an upsampling linear layer. As can be seen from Figure 5, except for the EuroSAT dataset where the "Prompt & Project" method shows better performance, the "Adapter" method generally achieves higher prediction accuracy on the test set compared to other supervised fine-tuning strategies. This demonstrates that the bilinear-layer adapter with a residual structure is a more effective method for supervised fine-tuning with a tiny number of samples. The reason is that the bilinear-layer structure of the adapter has a more complex structure than a single linear layer, allowing the student model to learn more fully from the tiny labeled dataset. Additionally, it retains the prompt parameters distilled by MUPD, which effectively alleviates the overfitting problem caused by the tiny number of samples.

Table 3: Experimental results (HM) of the second stage (ASFT) on 6 natural image datasets

		DTD	Flowers102	FGVCAircraft	Caltech101	OxfordPets	EuroSAT
First Phase	5-shot	71.8	86.3	37.4	96.8	93.3	78.1
Second Phase	1-shot	72.1	86.6	38.1	96.9	93.7	78.6
	3-shot	72.0	86.6	38.0	96.9	93.8	78.4
	5-shot	72.0	86.5	37.7	96.8	93.7	78.8

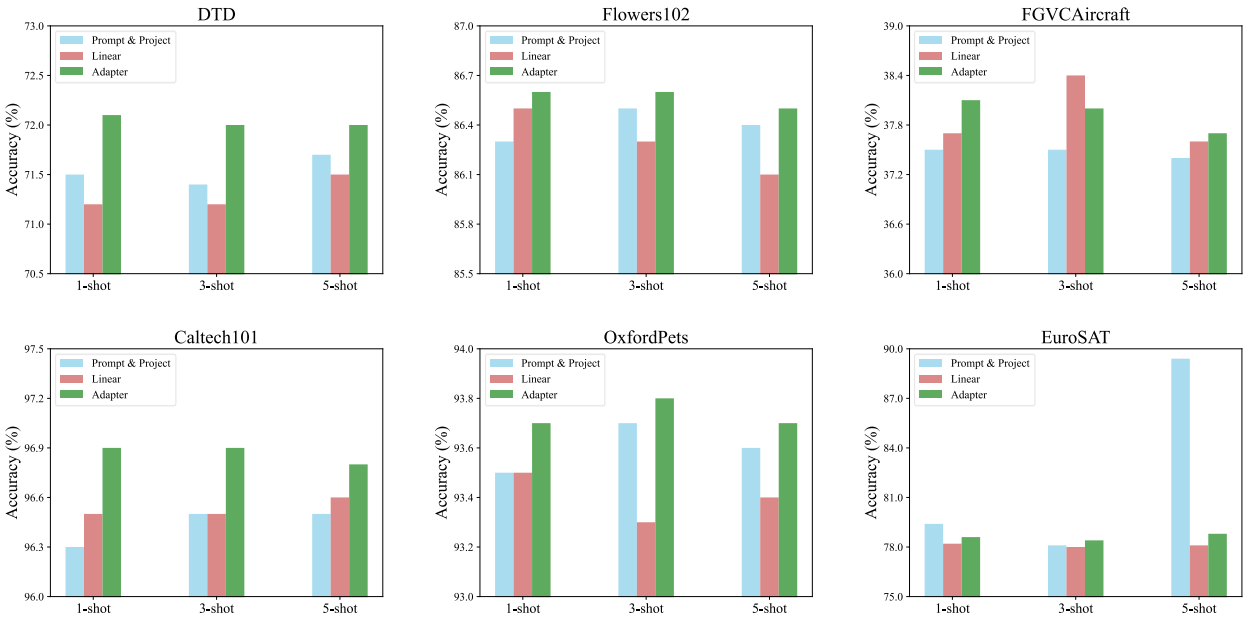

Figure 5: Ablation experiment results (HM) of the second stage on 6 natural image datasets

To more intuitively demonstrate the effectiveness of the MTKD method under few-shot conditions, this paper presents t-SNE plots illustrating the distribution of logits for images across different categories in the test sets of the Flowers102 and DTD datasets. These logits are generated by student models trained using both the Baseline method and the two stages (MUPD and ASFT) of the MTKD method. Both the Baseline and the two stages of MTKD are trained under 5-shot settings. As shown in Figure 6, compared to the Baseline, the student model trained in the MUPD stage produces logits for images of the same category that form tighter clusters in the low-dimensional space, while the logits for images of different categories exhibit less overlap in the low-dimensional space. Compared to the MUPD stage, the student model trained in the ASFT stage further tightens the clusters of logits for images of the same category and further reduces the overlap of logits for images of different categories in the low-dimensional space. This indicates that MTKD is an effective method for few-shot knowledge distillation in VLM, and it also highlights the effectiveness of each stage of MTKD.

4.4 Validation on medical images

To verify the effectiveness of MTKD in tasks where data deviates from pre-trained data, experiments were also conducted on 3 medical datasets (BACH, Brain, and EyeDR), as shown in Table 4. "Baseline" refers to the use of only instance-level constraints under a single temperature parameter setting. Δ(1) represents the performance improvement of the distillation result under multi-level constraints in the first stage compared to the Baseline. Δ(2) represents the performance improvement of the supervised fine-tuning result in the second stage compared to the distillation result under the 5-shot setting in the first stage. As can be seen from Table 4, on all 3 medical datasets, the student model distilled via the first stage of MTKD achieves higher prediction accuracy than that via the Baseline method. Additionally, the student model fine-tuned in the second stage performs better than the one distilled under the 5-shot setting in the first stage. This demonstrates that MTKD is also effective in tasks with significant differences from the distribution of pre-trained data.

4.5 Comparison with other methods

To further illustrate the advancement of MTKD, this subsection compares MTKD with previous methods, as shown in Table 5. For the MTKD method, the result with the highest harmonic mean (HM) among all shots in the second stage is selected for each dataset. For other methods, except for the 5-shot training sample setting, the remaining training settings are consistent with those reported in the original papers. The results show that when using ViT-B/16 as the student model, MTKD achieves state-of-the-art (SOTA) results on all datasets. With the same amount of data, MTKD achieves higher accuracy, which indicates better data utilization efficiency. The reason for these advanced results lies in two aspects: first, the first stage of MTKD aligns the probability distributions generated by the student model with those of the teacher model at three levels (instance-level, batch-level, and class-level), effectively enhancing knowledge representation under few-shot settings; second, the second stage fine-tunes the student model using an adapter with a residual structure on a tiny labeled sample set, alleviating the impact of minor inaccurate predictions of the teacher model on knowledge distillation performance. The combined effect of MUPD and ASFT significantly enhances the performance of MTKD in few-shot scenarios, rendering it notably superior to existing methods in terms of both efficiency and robustness. Additionally, MTKD offers clear advantages in terms of model deployment costs. Table 6 visually presents the differences in parameter count, processing time, and FLOPs between the teacher and student models used in the experiment. The learnable parameter count of the teacher model (ViT-L/14) is about 408M, while that of the student model (ViT-B/16) is about 143M. When processing 100 images, the processing time of the teacher model is 968.89 ms, and its floating-point operations (FLOPs) amount to 11.251T; while the processing time of the student model is 339.04 ms, and its FLOPs amount to 2.642T. The parameter count, FLOPs, and processing time of the student model are significantly lower than those of the teacher model, achieving high accuracy while substantially reducing computational and storage overhead. This makes MTKD suitable for deployment in scenarios with limited computational resources, such as on edge devices.

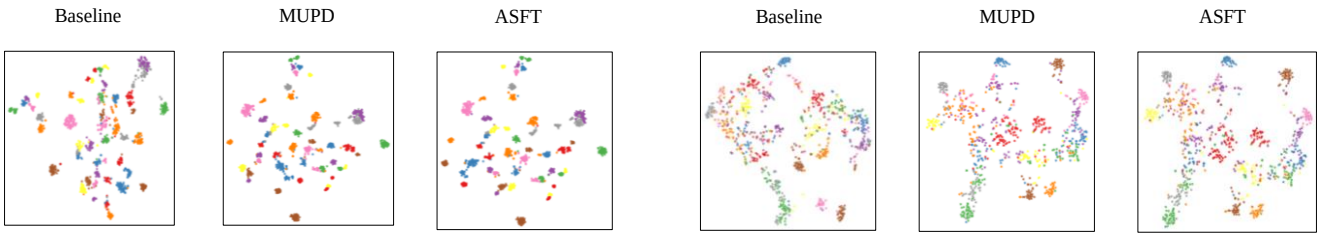

Figure 6: On Flowers102 and DTD datasets, t-SNE plots showing the distribution of logits generated by student models (trained using the instance-level constraint method under a single temperature parameter and the MTKD method, respectively) for input images of different categories in the test set. Different colors are used to distinguish different categories.

Table 4: Base-to-novel results on 3 medical image datasets

Dataset		1-shot			3-shot			5-shot		
		Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
BACH	Baseline	51.7	49.4	50.5	51.7	50.0	50.8	58.3	51.2	54.5
	First Phase	53.3	50.0	51.6	57.3	50.4	53.6	60.0	51.7	55.5
	$\Delta^{(1)}$	**+1.6**	**+0.6**	**+1.1**	**+5.6**	**+0.4**	**+2.8**	**+1.7**	**+0.5**	**+1.0**
	Second Phase	61.7	55.0	58.2	61.3	55.0	58.0	61.7	60.0	60.8
	$\Delta^{(2)}$	**+1.7**	**+3.3**	**+2.7**	**+1.3**	**+3.3**	**+2.5**	**+1.7**	**+8.3**	**+5.3**

Brain	Baseline	66.8	84.7	74.7	67.1	85.6	75.2	70.3	87.8	78.1
	First Phase	67.2	84.9	75.0	67.8	89.4	77.1	70.6	90.2	79.2
	$\Delta^{(1)}$	**+0.4**	**+0.2**	**+0.3**	**+0.7**	**+3.8**	**+1.9**	**+0.3**	**+2.4**	**+1.1**
	Second Phase	71.2	89.7	79.4	71.3	90.4	79.7	70.9	90.5	79.5
	$\Delta^{(2)}$	**+0.6**	**-0.5**	**+0.2**	**+0.7**	**+0.2**	**+0.5**	**+0.3**	**+0.3**	**+0.3**
EyeDR	Baseline	75.3	51.0	60.8	74.0	40.7	52.5	75.2	39.3	51.6
	First Phase	75.5	59.3	66.4	74.3	45.5	56.4	73.7	49.7	59.4
	$\Delta^{(1)}$	**+0.2**	**+8.3**	**+5.6**	**+0.3**	**+4.8**	**+3.9**	**-1.5**	**+10.4**	**+7.8**
	Second Phase	75.2	46.2	57.2	74.6	49.8	59.7	74.4	49.9	59.7
	$\Delta^{(2)}$	**+1.5**	**-3.5**	**-2.2**	**+0.9**	**+0.1**	**+0.3**	**+0.7**	**+0.2**	**+0.3**

Table 5: Comparison between MTKD and other methods (HM)

	DTD	**Flowers102**	**FGVCAircraft**	**Caltech101**	**OxfordPets**	**EuroSAT**
CoOp	50.0	66.8	24.7	92.7	89.0	58.9
CoCoOp	58.7	78.7	24.8	95.3	90.2	60.7
MaPLe	62.8	79.7	31.3	95.4	90.5	65.2
PromptSRC	65.7	82.5	34.1	95.5	90.2	65.2
PromptKD	69.1	84.7	36.4	95.8	91.0	70.2
Ours	72.1	86.6	38.1	96.9	93.8	78.8

Table 6: Differences between teacher model and student model

	Parameters	**Time of 100 samples / ms**	**FLOPs of 100 samples / T**
Teacher Model (ViT-L/14)	408M	968.89	11.251
Student Model (ViT-B/16)	143M	339.04	2.642

5 Conclusions

This paper proposes a two-stage few-shot knowledge distillation method based on multi-level constraints, aiming to solve the lightweight problem of Vision-Language Models (VLMs) when labeled data is insufficient. MTKD first uses a small amount of unlabeled data to transfer the knowledge of a large-scale teacher VLM to a lightweight student model through instance-level, batch-level, and class-level constraints. Then, the distilled student model is frozen, and an adapter implemented with a residual structure is inserted at the end of the student model. Supervised performance improvement is achieved using a tiny amount of labeled data. Extensive ablation experiments are conducted on 6 natural image datasets and 3 medical image datasets to verify the effectiveness of the proposed method.

6 Discussion

Although MTKD has achieved advanced performance on multiple standard datasets, few-shot knowledge distillation still faces numerous challenges and directions worthy of further exploration. Firstly, in few-shot knowledge distillation, the quality of training data is a critical factor influencing student model performance. The current method assumes that the training samples possess a certain degree of representativeness and diversity. However, in practical applications, if the training data lacks representativeness for the downstream task, the knowledge from the teacher model cannot be fully manifested through few-shot learning. Future work could further explore sample selection strategies to automatically choose the most informative or representative samples from a large pool of unlabeled data for distillation. Secondly, the current two-stage structure is relatively fixed. Future research could investigate more adaptive dynamic distillation frameworks. For instance, dynamically adjusting the distillation loss weights based on sample difficulty or model confidence. We hope that MTKD can serve as an effective baseline method for related fields and look forward to future work advancing the field in the directions mentioned above.

References

[1] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PmLR, 2021: 8748-8763.

[2] Han Z, Gao C, Liu J, et al. Parameter-efficient fine-tuning for large models: A comprehensive survey[J]. arXiv preprint arXiv:2403.14608, 2024.

[3] Zhou K, Yang J, Loy C C, et al. Learning to prompt for vision-language models[J]. International Journal of Computer Vision, 2022, 130(9): 2337-2348. https://doi.org/10.1007/s11263-022-01653-1

[4] Zhou K, Yang J, Loy C C, et al. Conditional prompt learning for vision-language models[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 16816-16825. https://doi.org/10.1109/CVPR52688.2022.01631

[5] Khattak M U, Rasheed H, Maaz M, et al. Maple: Multi-modal prompt learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 19113-19122. https://doi.org/10.1109/CVPR52729.2023.01832

[6] Khattak M U, Wasim S T, Naseer M, et al. Self-regulating prompts: Foundational model adaptation without forgetting[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2023:

15190-15200. https://doi.org/10.1109/ICCV51070.2023.01394
[7] Gao P, Geng S, Zhang R, et al. Clip-adapter: Better vision-language models with feature adapters[J]. International Journal of Computer Vision, 2024, 132(2): 581-595. https://doi.org/10.1007/s11263-023-01891-x
[8] Zhang R R, Zhang W, Fang R Y, et al. Tip-Adapter: Training-Free Adaption of CLIP for Few-Shot Classification[C]// Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXV. Berlin, Heidelberg: Springer-Verlag, 2022: 493-510. https://doi.org/10.1007/978-3-031-19833-5_29
[9] Mansourian A M, Ahmadi R, Ghafouri M, et al. A Comprehensive Survey on Knowledge Distillation[J]. arXiv preprint arXiv:2503.12067, 2025.
[10] Yang C, An Z, Huang L, et al. Clip-kd: An empirical study of clip model distillation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 15952-15962. https://doi.org/10.1109/CVPR52733.2024.01510
[11] Yang K, Gu T, An X, et al. Clip-cid: Efficient clip distillation via cluster-instance discrimination[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2025, 39(20): 21974-21982. https://doi.org/10.1609/aaai.v39i20.35505
[12] Nair L. CLIP-Embed-KD: Computationally Efficient Knowledge Distillation Using Embeddings as Teachers[J]. arXiv preprint arXiv:2404.06170, 2024.
[13]Liu Z, Hu X, Nevatia R. Efficient Feature Distillation for Zero-shot Annotation Object Detection[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024: 893-902. https://doi.org/10.1109/WACV57701.2024.00094
[14] Pei R, Liu J, Li W, et al. Clipping: Distilling clip-based models with a student base for video-language retrieval[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 18983-18992. https://doi.org/10.1109/CVPR52729.2023.01820
[15] Fang Z Y, Wang J F, Wang L J, et al. Seed: Self-supervised distillation for visual representation[C]// International Conference on Learning Representations (ICLR), 2021.
[16] Hu H, Xie L, Hong R, et al. Creating something from nothing: Unsupervised knowledge distillation for cross-modal hashing[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 3123-3132. https://doi.org/10.1109/CVPR42600.2020.00319
[17] Li Z, Li X, Fu X, et al. Promptkd: Unsupervised prompt distillation for vision-language models[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 26617-26626. https://doi.org/10.1109/CVPR52733.2024.02513
[18] Jin Y, Wang J, Lin D. Multi-level logit distillation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 24276-24285. https://doi.org/10.1109/CVPR52729.2023.02325
[19] Park W, Kim D, Lu Y, et al. Relational knowledge distillation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 3967-3976. https://doi.org/10.1109/CVPR.2019.00409
[20] Mirzadeh S I, Farajtabar M, Li A, et al. Improved knowledge distillation via teacher assistant[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(04): 5191-5198. https://doi.org/10.1609/aaai.v34i04.5963
[21] Hao Z, Guo J, Han K, et al. Revisit the power of vanilla knowledge distillation: from small scale to large scale[J]. Advances in Neural Information Processing Systems, 2023, 36: 10170-10183. https://doi.org/10.52202/075280-0444
[22] Song L, Gong X, Zhou H, et al. Exploring the Knowledge Transferred by Response-Based Teacher-Student Distillation[C]//Proceedings of the 31st ACM International Conference on Multimedia. 2023: 2704-2713. https://doi.org/10.1145/3581783.3612162
[23] Sau B B, Balasubramanian V N. Deep model compression: Distilling knowledge from noisy teachers[J]. arXiv preprint arXiv:1610.09650, 2016.
[24] Romero A, Ballas N, Kahou S E ,et al. Fitnets: Hints for Thin Deep Nets[C]//ICLR.2015.
[25] Kim J, Park S U, Kwak N. Paraphrasing complex network: Network compression via factor transfer[J]. Advances in neural information processing systems, 2018, 31.
[26] Heo B, Kim J, Yun S, et al. A comprehensive overhaul of feature distillation[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1921-1930. https://doi.org/10.1109/ICCV.2019.00201
[27] Yim J, Joo D, Bae J, et al. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4133-4141. https://doi.org/10.1109/CVPR.2017.754
[28] Xie J, Shuai B, Hu J F, et al. Improving fast segmentation with teacher-student learning[J]. arXiv preprint arXiv:1810.08476, 2018.
[29] Wang Y, Yao Q, Kwok J T, et al. Generalizing from a few examples: A survey on few-shot learning[J]. ACM computing surveys (csur), 2020, 53(3): 1-34. https://doi.org/10.1145/3386252
[30] Parnami A, Lee M. Learning from few examples: A summary of approaches to few-shot learning[J]. arXiv preprint arXiv:2203.04291, 2022.
[31] Jia M, Tang L, Chen B C, et al. Visual prompt tuning[C]//European conference on computer vision. Cham: Springer Nature Switzerland, 2022: 709-727. https://doi.org/10.1007/978-3-031-19827-4_41

[32] Prakash R, Sharma P. Prompt Tuning on Vision-Language Models: A Survey[J].

[33] Yin D, Hu L, Li B, et al. Adapter is all you need for tuning visual tasks[J]. arXiv preprint arXiv:2311.15010, 2023.

[34] Yang L, Zhang R Y, Wang Y, et al. MMA: Multi-Modal Adapter for Vision-Language Models[C]//2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2024. DOI: 10.1109/CVPR52733.2024.02249. https://doi.org/10.1109/CVPR52733.2024.02249

[35] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[36] Jia C, Yang Y, Xia Y, et al. Scaling up visual and vision-language representation learning with noisy text supervision[C]//International conference on machine learning. PMLR, 2021: 4904-4916.

[37] Li J, Li D, Xiong C, et al. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation[C]//International conference on machine learning. PMLR, 2022: 12888-12900.

[38] Lan W, Cheung Y, Xu Q, et al. Improve knowledge distillation via label revision and data selection[J]. IEEE Transactions on Cognitive and Developmental Systems, 2025. https://doi.org/10.1109/TCDS.2025.3559881

[39] Wang H, Lohit S, Jones M N, et al. What makes a" good" data augmentation in knowledge distillation-a statistical perspective[J]. Advances in Neural Information Processing Systems, 2022, 35: 13456-13469. https://doi.org/10.52202/068431-0978

[40] Xin Y, Du J, Wang Q, et al. Vmt-adapter: Parameter-efficient transfer learning for multi-task dense scene understanding[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2024, 38(14): 16085-16093. https://doi.org/10.1609/aaai.v38i14.29541

[41] Cimpoi M, Maji S, Kokkinos I, et al. Describing textures in the wild[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 3606-3613. https://doi.org/10.1109/CVPR.2014.461

[42] Parkhi O M, Vedaldi A, Zisserman A, et al. Cats and dogs[C]//2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012: 3498-3505. https://doi.org/10.1109/CVPR.2012.6248092

[43] Nilsback M E, Zisserman A. Automated flower classification over a large number of classes[C]//2008 Sixth Indian conference on computer vision, graphics & image processing. IEEE, 2008: 722-729. https://doi.org/10.1109/ICVGIP.2008.47

[44] Maji S, Rahtu E, Kannala J, et al. Fine-grained visual classification of aircraft[J]. arXiv preprint arXiv:1306.5151, 2013.

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

[46] Helber P, Bischke B, Dengel A, et al. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019, 12(7): 2217-2226. https://doi.org/10.1109/JSTARS.2019.2918242

[47] Boulkroune, A., Boubellouta, A., Bouzeriba, A. et al. Practical Finite-Time Fuzzy Synchronization of Chaotic Systems with Non-Integer Orders: Two Chattering-Free Approaches. J. Syst. Sci. Syst. Eng. 34, 334-359 (2025). https://doi.org/10.1007/s11518-024-5635-7

[48] Rigatos, G., Abbaszadeh, M., Busawon, K., Dala, L., Pomares, J., and Zouari, F. (December 6, 2023). "Flatness-Based Control in Successive Loops for Autonomous Quadrotors." ASME. J. Dyn. Sys., Meas., Control. March 2024; 146(2): 024501. https://doi.org/10.1115/1.4063907

[49] Rigatos, G., Siano, P., Zouari, F. et al. Nonlinear optimal control of autonomous submarines' diving. Mar Syst Ocean Technol 15, 57-69 (2020). https://doi.org/10.1007/s40868-019-00070-3

[50] Rigatos, G., Busawon, K., Abbaszadeh, M., Pomares, J., Gao, Z., & Zouari, F. (2024, August). Flatness-based control in successive loops for dual-arm robotic manipulators. In 2024 IEEE Conference on Control Technology and Applications (CCTA) (pp. 793-798). IEEE. https://doi.org/10.1109/CCTA60707.2024.10666567

[51] G. Rigatos, P. Siano, F. Zouari and S. Ademi, "A nonlinear optimal control method for autonomous submarines' diving," 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 2017, pp. 1061-1066, doi: 10.1109/ISIE.2017.8001393. https://doi.org/10.1109/ISIE.2017.8001393

[52] Zouari, F., & Mahmud, M. (2024, April). Neural Network-Based Robust Adaptive Output Feedback Control for MIMO Time-Varying Delay Systems. In Global Conference on Applications of Artificial (pp. 60-77). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-98498-3_5

[53] Almazroi, A. A., Alkinani, M. H., Al-Shareeda, M. A., & Manickam, S. (2024). A novel ddos mitigation strategy in 5g-based vehicular networks using chebyshev polynomials. Arabian Journal for Science and Engineering, 49(9), 11991-12004.https://doi.org/10.1007/s13369-023-08535-9

[54] Almazroi, A. A., Alqarni, M. A., Al-Shareeda, M. A., Alkinani, M. H., Almazroey, A. A., & Gaber, T. (2024). FCA-VBN: Fog computing-based authentication scheme for 5G-assisted vehicular blockchain network. Internet of Things, 25, 101096. https://doi.org/10.1016/j.iot.2024.101096

[55] Al-shareeda, M. A., Anbar, M., Hasbullah, I. H., Manickam, S., Abdullah, N., & Hamdi, M. M. (2020, September). Review of prevention schemes for

replay attack in vehicular ad hoc networks (vanets). In 2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP) (pp. 394-398). IEEE. https://doi.org/10.1109/ICICSP50920.2020.9232047

[56] Almazroi, A. A., Aldhahri, E. A., Al-Shareeda, M. A., & Manickam, S. (2023). ECA-VFog: An efficient certificateless authentication scheme for 5G-assisted vehicular fog computing. Plos one, 18(6), e0287291. https://doi.org/10.1371/journal.pone.0287291

[57] Almazroi, A. A., Alqarni, M. A., Al-Shareeda, M. A., & Manickam, S. (2023). L-CPPA: Lattice-based conditional privacy-preserving authentication scheme for fog computing with 5G-enabled vehicular system. Plos one, 18(10), e0292690. https://doi.org/10.1371/journal.pone.0292690

[58] Al-Shareeda, M. A., Gaber, T., Alqarni, M. A., Alkinani, M. H., Almazroey, A. A., & Almazroi, A. A. (2025). Chebyshev polynomial based emergency conditions with authentication scheme for 5G-assisted vehicular fog computing. IEEE Transactions on Dependable and Secure Computing. ttps://doi.org/10.1109/TDSC.2025.3553868