Towards Difficulty-Agnostic Efficient Transfer Learning for Vision-Language Models

Anonymous ACL submission

Abstract

Vision-language models (VLMs) like CLIP have demonstrated remarkable applicability across a variety of downstream tasks, including zero-shot image classification. Recently, the use of prompts or adapters for efficient transfer learning (ETL) has gained significant attention for effectively adapting to downstream tasks. However, previous studies have overlooked the challenge of varying transfer difficulty of downstream tasks. In this paper, we empirically analyze how each ETL method behaves with respect to transfer difficulty. Our observations indicate that utilizing vision prompts and text adapters is crucial for adaptability and generalizability in domains with high difficulty. Also, by applying an adaptive ensemble approach that integrates task-adapted VLMs with pre-trained VLMs and strategically leverages more general knowledge in low-difficulty and less in high-difficulty domains, we consistently enhance performance across both types of domains. Based on these observations, we propose an adaptive ensemble method that combines visual prompts and text adapters with pre-trained VLMs, tailored by transfer difficulty, to achieve optimal performance for any target domain. Upon experimenting with extensive benchmarks, our method consistently outperforms all baselines, particularly on unseen tasks, demonstrating its effectiveness.

1 Introduction

Vision-language models (VLMs), such as CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), have demonstrated remarkable applicability across various downstream tasks such as image classification. A distinctive feature of these VLMs for image classification is their ability to classify unseen classes that have not been encountered during pre-training through zero-shot inference, which is not possible to traditional vision models.

The primary challenge of VLMs for downstream tasks is to excel in classifying both seen and unseen class sets. In the context of VLM classification tasks, the ability to accurately classify seen class sets is termed *adaptability*, while the capability to extend this proficiency to unseen class sets is referred to as *generalizability*. To boost these abilities, recent research has introduced efficient transfer learning (ETL) methods to fine-tune VLMs. One strategy involves the use of soft prompt tuning (Zhou et al., 2022b,a; khattak et al., 2023; Khattak et al., 2023). Another research direction involves adapter-style tuning (Gao et al., 2023; Zhang et al., 2022; Zhu et al., 2023b) either by adjusting specific parameters or employing cache-based techniques. These approaches empower VLMs to swiftly adapt to new tasks using only a few samples (i.e. few-shot image classification task).

However, previous approaches do not consider a significant factor for adapting to downstream tasks: varying transfer difficulty (Yu et al., 2023). This refers to the challenge of adapting pre-trained VLMs according to the target domain. For instance, transferring pre-trained VLMs to specific fine-grained domains, such as FGVC Aircraft, is more challenging than transferring to general coarse-grained domains. In a real-world scenario, it is hard to predict the specific target task and domain that will emerge. Therefore, without investigating how each type of ETL behaves in response to different levels of transfer difficulty and applying an adaptive method based on this investigation, the result for each target domain can be suboptimal. Some works manually train models differently for each dataset (Gao et al., 2023; Zhang et al., 2022), but this approach is not feasible in real-world scenarios as prior knowledge for the target task is not given.

To overcome these limitations and apply an adaptive method for tuning adapters and prompts for downstream tasks, we empirically investigate the characteristics of applying different tuning methods for each modality on multiple domains with varying transfer difficulty, revealing four key findings.

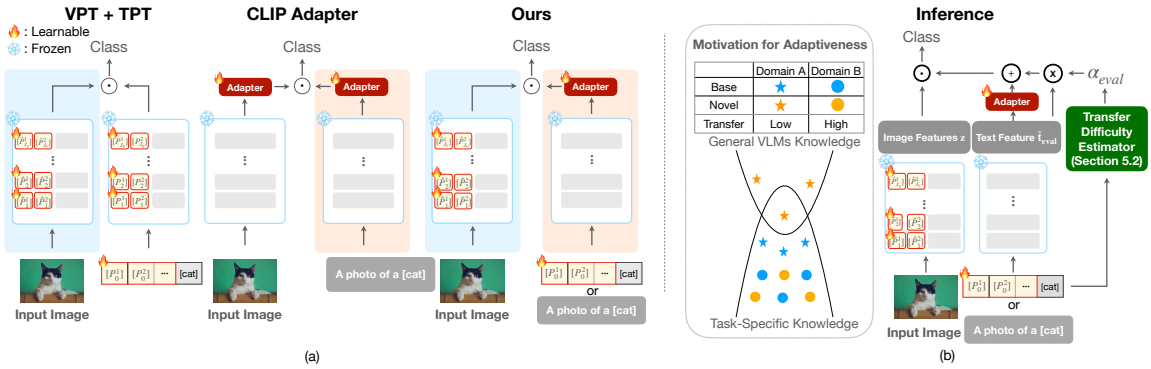

Figure 1: Overview of **APEX** compared to the conventional ETL methods. **APEX** exhibits two key differences: **(a)**: *Firstly*, **APEX** integrates prompt tuning for the visual encoder and a linear adapter for the text encoder, each tailored to the specific properties of their respective modalities, which performs better on high-difficulty domains. **(b)**: *Secondly*, **APEX** integrates an adaptive coefficient within the text encoder to strategically balance pre-adapter and post-adapter features to properly combine task-specific knowledge and general VLMs knowledge based on transfer difficulty. A detailed explanation, including notations and the algorithm, can be found in Section 4 and Appendix B.

Firstly, we find that visual prompt tuning (VPT) generalizes better to unseen classes compared to text prompt tuning (TPT) in cases of high-difficulty domains, as TPT tends to overfit on base classes for these domains. (▷ Obs. 1). This occurs because, in high-difficulty domains, the class separability of visual features from a visual encoder is low, causing TPT to overly adapt in classifying these challenging features (▷ Obs. 2). *Moreover*, text adapter (TA) can significantly boost the adaptability of VPT, resulting in high adaptability and generalizability, especially for highly difficult domains (▷ Obs. 3). However, fine-tuning with adapters could compromise generalizability in easier domains. Our *last* observation is that combining pre- and post-adapter features to leverage pre-trained VLMs knowledge can address this concern with a proper balance between them. For instance, using more pre-adapter features can maintain generalizability in easier domains. The ideal balance depends on the domain's difficulty, highlighting the need to adjust the ensemble coefficient accordingly (▷ Obs. 4).

Based on our observations, we present a **APEX** (text Adapter, visual Prompt, and adaptive Ensemble for cross(X)-modality) that utilizes an adaptive ensemble with VPT and TA. Specifically, we use the combination of VPT and TA, which have shown high generalizability and adaptability for high-difficulty domains, as shown in Obs. 1-3 (Fig. 1(a)). Also, motivated by Obs. 4, we employ an adaptive ensemble approach that determines the optimal ensemble coefficient for each domain by using the distances to learned classes in pre-trained VLMs to estimate transfer difficulty (Fig. 1(b)). This adaptive ensemble controls the level of adaptation, by primarily utilizing task-specific knowledge with adapted VLMs for high-difficulty domains but leveraging general knowledge for low-difficulty domains, as pre-trained VLMs already possess sufficient ability and prevent an overfitting from excessive adaptation. With this, our method acts as a difficulty-agnostic solution, enabling the model to effectively adapt to all target domains regardless of transfer difficulty. In summary, our main contributions are:

- We investigate prompt tuning and adapter tuning methods to understand their effectiveness across domains with varying transfer difficulties. Our findings reveal that the efficacy of each method with each modality varies across different of transfer difficulty, with notable performance of VPT and TA for high-difficulty domains.
- We propose **APEX**, which utilizes VPT and TA for tuning and employ an adaptive ensemble approach to optimally leverage the general knowledge of VLMs for each domain. The ensemble's coefficient is adaptively determined by the distances to learned classes, serving as an estimate of transfer difficulty.
- We show that **APEX** achieves state-of-the-art performance across various downstream tasks, with particularly notable improvements in unseen tasks during adaptation.

2 Backgrounds

Here, we provide a brief overview of the background related to our method. For a detailed explanation with more related works is in Appendix E.

Zero-shot CLIP. CLIP (Radford et al., 2021) is designed for creating visual features based on natural language guidance. The CLIP model can perform zero-shot inference, classifying an image into one of C possible classes without additional training. This is achieved by calculating the cosine similarity between an visual feature $\mathbf{z}$, derived from the visual encoder, and the text features of each class $\{\mathbf{t}_i\}_{i=1}^{C}$, which are obtained from the text encoder.

For processing the image, let us define the visual encoder as $\mathcal{V}$, which comprises $L_{\mathcal{V}}$ layers, denoted as $\{\mathcal{V}_i\}_{i=1}^{L_{\mathcal{V}}}$. The encoder takes patch embeddings $\mathbf{E}_0 \in \mathbb{R}^{M \times d_v}$ as input, which are obtained by dividing the image I into M fixed-size patches. Patch embeddings $\mathbf{E}_i$ is then fed into the $(i+1)^{\text{th}}$ transformer block ($\mathcal{V}_{i+1}$) along with a learnable class ([CLS]) tokens $\mathbf{c}_i$. This process is sequentially carried out through all $L_{\mathcal{V}}$ transformer blocks, formulated as follows:

$$[\mathbf{c}_i, \mathbf{E}_i] = \mathcal{V}_i\left([\mathbf{c}_{i-1}, \mathbf{E}_{i-1}]\right) \quad i = 1, \ldots, L_{\mathcal{V}}, \quad (1)$$

$$\mathbf{z} = \texttt{ImageProj}(\mathbf{c}_{L_{\mathcal{V}}}), \quad (2)$$

Here, $[\cdot, \cdot]$ denotes the concatenation operation. We can obtain the text features in a similar way with word embeddings $\mathbf{W}_0 = [\mathbf{w}_0^1, \ldots, \mathbf{w}_0^N] \in \mathbb{R}^{N \times d_l}$ and text encoder $\mathcal{T}$ which is consist of $L_{\mathcal{T}}$ layers $\{\mathcal{T}_i\}_{i=1}^{L_{\mathcal{T}}}$, as follows:

$$[\mathbf{W}_i] = \mathcal{T}_i(\mathbf{W}_{i-1}) \quad i = 1, \ldots, L_{\mathcal{T}} \quad (3)$$

$$\mathbf{t}_i = \texttt{TextProj}(\mathbf{w}_{L_{\mathcal{T}}}^N) \quad (4)$$

The predicted probability for class i is as:

$$\Pr(y = i|\mathbf{z}, \mathbf{t}) = \frac{\exp\left(\text{sim}(\mathbf{z}, \mathbf{t}_i)/\tau\right)}{\sum_{j=1}^{C} \exp\left(\text{sim}(\mathbf{z}, \mathbf{t}_j)/\tau\right)}, \quad (5)$$

where $\text{sim}(\cdot, \cdot)$ indicates cosine similarity and τ is the learned temperature of CLIP. We can also interpret the text features as a **classifier** (Gao et al., 2023; Zhang et al., 2022), where $\mathbf{t}_i$ is the classifier weight for class i.

Prompt Tuning for CLIP. To enable prompt tuning (Zhou et al., 2022a; khattak et al., 2023; Zhu et al., 2023a; Khattak et al., 2023), we replace the Eq. (1) and Eq. (3) by newly introducing $b_{\mathcal{V}}$ and $b_{\mathcal{T}}$ learnable tokens $\{\hat{P}_i^k \in \mathbb{R}^{d_v}\}_{k=1}^{b_{\mathcal{V}}}$ and $\{P_i^k \in \mathbb{R}^{d_l}\}_{k=1}^{b_{\mathcal{T}}}$ for i^{th} layer, and their concatenation $\hat{\mathbf{P}}_i$ and $\mathbf{P}_i$. We can introduce the visual prompt for the first $J_{\mathcal{V}}$ layers of the visual encoder, then we can compute as follows:

$$[\mathbf{c}_i, \mathbf{E}_i, \_] = \mathcal{V}_i([\mathbf{c}_{i-1}, \mathbf{E}_{i-1}, \hat{\mathbf{P}}_{i-1}]), \quad (6)$$

$$[\mathbf{c}_j, \mathbf{E}_j, \hat{\mathbf{P}}_j] = \mathcal{V}_j([\mathbf{c}_{j-1}, \mathbf{E}_{j-1}, \hat{\mathbf{P}}_{j-1}]),$$

for $i = 1, \ldots, J_{\mathcal{V}}$ and $j = J_{\mathcal{V}} + 1, \ldots, L_{\mathcal{V}}$. Also, we can replace Eq. (3) to belows by introducing text prompt for the fisrt $J_{\mathcal{T}}$ layers of text encoder:

$$[\_, \mathbf{W}_i] = \mathcal{T}_i([\mathbf{P}_{i-1}, \mathbf{W}_{i-1}]) \quad i = 1, \ldots, J_{\mathcal{T}}, \quad (7)$$

$$[\mathbf{P}_j, \mathbf{W}_j] = \mathcal{T}_j([\mathbf{P}_{j-1}, \mathbf{W}_{j-1}]) \quad j = J_{\mathcal{T}} + 1, \ldots, L_{\mathcal{T}}.$$

Here, we train the visual and text prompt for the first $J_{\mathcal{V}}$ and $J_{\mathcal{T}}$ layers of corresponding encoders.

Adapter-style Tuning for CLIP. To enable adapter-style tuning, we replace Eq. (2) and Eq. (4) by introducing **`ImgAdapt`** and **`TxtAdapt`** which are shallow stacking networks upon the frozen CLIP model (Gao et al., 2023; Zhang et al., 2022; Zhu et al., 2023b).

$$\tilde{\mathbf{z}} = \texttt{ImgProj}(\mathbf{c}_{L_{\mathcal{V}}}), \quad \mathbf{z} = \texttt{ImgAdapt}(\tilde{\mathbf{z}}) \quad (8)$$

$$\tilde{\mathbf{t}} = \texttt{TxtProj}(\mathbf{w}_{L_{\mathcal{T}}}^N), \quad \mathbf{t} = \texttt{TxtAdapt}(\tilde{\mathbf{t}}) \quad (9)$$

3 Motivating Observations

Here, we analyze the behavior of visual and text encoders depending on different tuning methods and transfer difficulty of target domains within the framework of ETL. To accomplish this, we begin by categorizing domains based on their relative transfer difficulty (RTD), which is a metric first defined by Yu et al. (2023).

Definition 1 (Relative Transfer Difficulty (Yu et al., 2023))**.** *Let $f(\cdot)$ and $g(\cdot)$ be random classifiers where the precision of each equals $1/C$, and zero-shot CLIP, respectively. Also, $Prec_f$ and $Prec_g$ denote the precision of classifiers f and g. Then, RTD is formulated as follows:*

$$RTD = \frac{Prec_f}{Prec_g} = \frac{1/C}{Prec_g} = \frac{1}{C \cdot Prec_g}$$

Under this metric, we identify EuroSAT, DTD, and FGVC Aircraft as the three most challenging domains, while ImageNet, SUN397, and Stanford Cars are recognized as the three easiest domains. We will primarily focus on these six domains to clearly demonstrate the impact of RTD on VLMs' behavior. To assess adaptability and generalizability, we train the CLIP-B/16 utilizing each prompt tuning approach on tasks requiring generalization from base to novel categories. Here, "base category" refers to a subset of classes within the domain learned through few-shot methods, and "novel category" is those not included in the training. Each dataset is split into these categories; the model is trained on base classes with 16 shots and tested

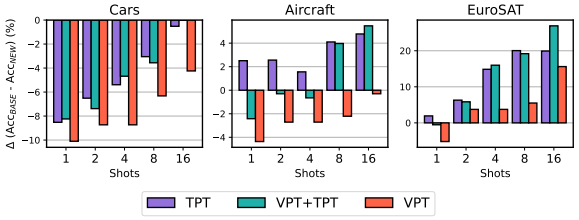

Figure 2: Comparison of accuracy differences (%) between base and novel categories across three prompt tuning options (TPT, VPT+TPT, VPT) with varying numbers of shots.

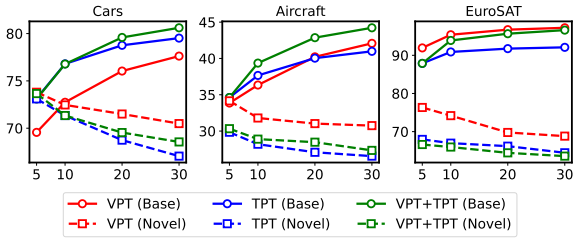

Figure 3: Comparison of the accuracy (%) of base and novel categories using TPT, VPT, and their combination (VPT+TPT) on three transfer learning datasets over various training epochs.

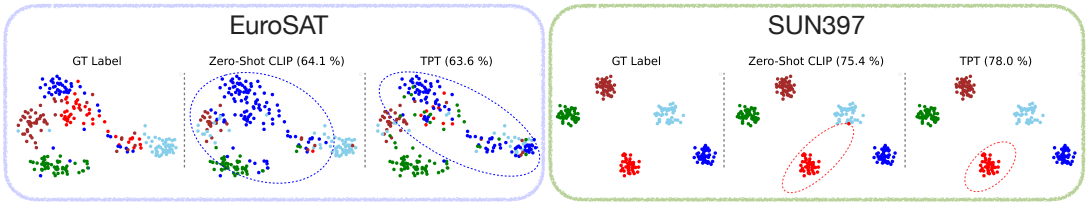

Figure 4: t-SNE (Van der Maaten and Hinton, 2008) plots of visual features for novel category with their corresponding labels **(left)**, zero-shot CLIP prediction **(middle)**, and prediction with TPT **(right)**. 50 samples are randomly selected from each class in EuroSAT and SUN397, using all 5 classes in EuroSAT and 5 randomly chosen classes from SUN397. Dotted lines within the t-SNE plot represent the decision boundaries corresponding to each class, indicated by the same color.

on both. Therefore, performance on the "base category" is related to adaptability, and performance in the "novel category" is related to generalizability. More detailed values are present in Appendix D.

Observation 1. *VPT offers better generalizability than TPT. While TPT has greater adaptability to seen classes in low-difficulty domains, it is not effective for high-difficulty domains and shows overfitting to the base classes.*

We commence with an analysis of the separate behavior of visual and text prompts during the tuning process. Fig. 2 illustrates the performance discrepancy between the two categories for each method. Across all domains, VPT consistently shows the smallest performance gap for every shot number, indicating reduced overfitting to base classes. This observation is especially prominent in domains with high RTD though the trend is not as pronounced in domains with low RTD. We also observe that combining VPT and TPT does not consistently mitigate the overfitting of TPT, as evidenced by the larger performance gap in FGVC Aircraft and EuroSAT compared to TPT alone.

Fig. 3 displays the comparative performance of base and novel categories over different epochs. While all prompt tuning methods show an improvement in base category performance at the expense of generalization, VPT consistently exhibits a lesser decline in novel category performance. Notably, for challenging domains like FGVC Aircraft and EuroSAT, VPT exceeds the novel performance of TPT and their combination regardless of epoch.

Observation 2. *Low class separability of visual features is the primary reason for the overfitting of TPT on high RTD.*

Class separability is a critical factor in determining the transferability of a source model to a target domain (Pándy et al., 2022). To determine the class separability of visual features, we use the ratio of intra- to inter-class cosine similarities (Oh et al., 2021; Zhu et al., 2023b). Fig. 5 demonstrates that the ratio is higher in domains with lower RTD, which are considered easier, and lower in more challenging datasets with higher RTD. These findings suggest that the class separability highly correlates with transfer difficulty, strongly influencing the overfitting risk of TPT on high RTD domains.

To see how class separability affects TPT, we further explore the visual features and predictions of zero-shot CLIP and TPT. As shown in Fig. 4, EuroSAT, which exhibits a high RTD, shows lower class separability compared to SUN397 that has a lower RTD. Furthermore, in EuroSAT, when TPT attempts to classify visual features with low class separability, its performance for novel classes is lower than zero-shot CLIP. This is because TPT

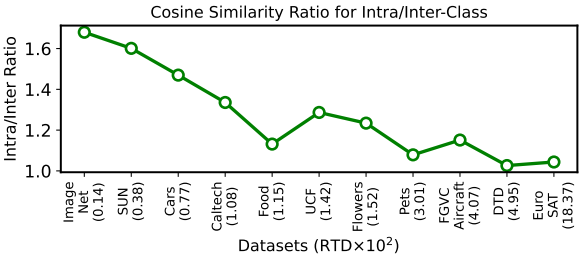

Figure 5: Comparison of intra- and inter-class ratios to show class separability across different datasets with their RTD, arranged from low to high RTD.

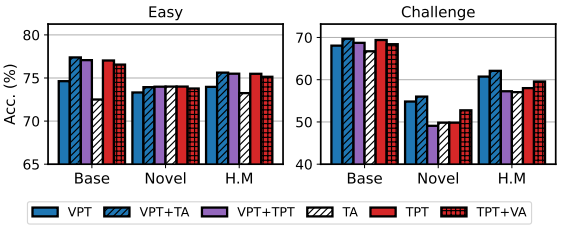

Figure 6: Comparison of the combined effectiveness of prompt tuning and adapter-style tuning. “Easy" refers to three domains with low RTD, and “Challenge" refers to three domains with high RTD.

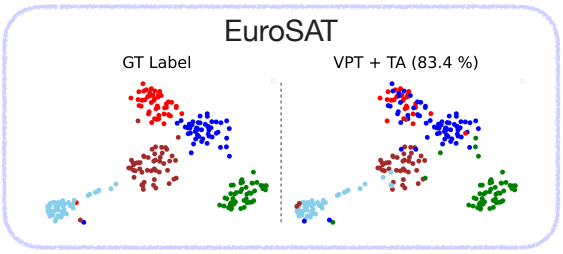

Figure 7: t-SNE plots of visual features of CLIP with VPT for a novel category with their corresponding labels **(left)** and prediction with TA **(right)**. 50 samples are randomly selected from each class.

tries to fit the decision boundary, represented as dotted lines, to features that are challenging to classify by solely adjusting classifier weights with multiple stacks of learnable prompts. This underscore the significance of separable visual features, a factor closely linked to VPT. Consequently, this leads to significant overfitting, where the decision boundary of one class overlaps with others. Conversely, with visual features that exhibit high class separability, TPT's predictions are more accurate than those of zero-shot CLIP as it can easily determine the better decision boundary.These results underscore the significance of separable visual features, a factor closely linked to VPT.

Observation 3. *TA effectively enhances adaptability with a low risk of overfitting when employed with VPT, especially on higher RTD datasets.*

Fig. 6 shows that while TA and VPT each exhibit less adaptability than TPT alone, together they outperform across all categories, signifying both high adaptability and generalizability. This advantageous combination is particularly significant for higher RTD, while the performance improvement in novel categories with lower RTD is marginal.

This synergy occurs because VPT enhances the class separability in visual features, allowing the linear transformation of classifier weights to suffice for adaptation, as depicted in Fig. 7. TA simply modifies the features of the pre-trained text encoder, preventing overconfidence in the decision boundary, especially for domains with high RTD and low class separability. In addition, we conduct experiments using a combination of TPT and a visual adapter (VA). However, this combination proves less effective than integrating VPT and TA, further emphasizing the importance of visual feature separability.

Observation 4. *By modulating the influence of TA through an ensemble of pre-adapter and post-adapter features, each with a domain-specific coefficient, we can significantly improve generalization in low RTD domains while maintaining high performance in high RTD domains.*

While combining VPT and TA has great synergy in high RTD domains, utilizing TA can result in the loss of some general knowledge from the original CLIP, which is crucial for domains with low RTD. This is evident in Tab. 1, as naïvely using VPT and TA together may lead to a degradation in performance on novel classes in domains with low RTD. This is because for low RTD, a lot of tasks within the domain need to lie in the region of general knowledge, as illustrated in Fig. 1(b). But the training of a TA creates a task-specific boundary which may not be optimal for other tasks within the same domain. In domains with high RTD, task-specific knowledge gained from adapters can also enhance performance on unseen tasks, as the general knowledge is often insufficient for these domains.

This degradation in domains with low RTD can be mitigated by diminishing the influence of TA. Inspired by the residual connection in adapter-style tuning methods (Zhang et al., 2022; Gao et al., 2023), we use an ensemble of pre-adapter and post-adapter features for the text encoder. This ensemble, defined with coefficient α, can be expressed as:

$$\mathbf{t} = \alpha \cdot \texttt{TxtAdapt}(\tilde{\mathbf{t}}) + (1 - \alpha) \cdot \tilde{\mathbf{t}}. \quad (10)$$

Table 1: Comparison of accuracy (%) on novel classes between zero-shot CLIP, without an ensemble, an ensemble with fixed coefficient, and an ensemble with optimal coefficient. We determine the fixed coefficient as 0.4, based on average novel performance.

Dataset	SUN397	Stanford Cars	DTD	EuroSAT
ZS CLIP	75.35	74.89	59.90	64.05
VPT + TA	74.52 (-0.83)	68.40 (-6.49)	63.05 (+3.15)	77.73 (+13.68)
+ Fixed Ens ($\alpha = 0.4$)	78.68 (+3.33)	74.22 (-0.67)	64.16 (+4.26)	75.87 (+11.82)
+ Opt. Ens	78.90 (+3.55)	75.19 (+0.30)	64.32 (+4.42)	77.73 (+13.68)
Opt. α	0.3	0.0	0.5	1.0

As Tab. 1 illustrates, the ensemble method improves performance in domains with low RTD. However, using pre-adapter features can yield suboptimal outcomes in more challenging domains. For instance, performance on EuroSAT drops from 77.73% to 75.87% when α is set as a fixed coefficient, as domains with high RTD demand more from TA. By optimally setting α for each domain, we consistently outperform zero-shot CLIP across all domains by effectively combining general and task-specific knowledge tailored to each domain's needs. Observing this optimal coefficient, we note that that more challenging domains typically require a higher coefficient. These findings highlight the necessity of a method to calculate an adaptive coefficient of ensemble, which would modulate TA activation according to domain and its RTD.

4 Method

Based on our observations, we propose a new method, **APEX**, which is a difficulty-agnostic approach that utilizes an adaptive ensemble with tuning methods including VPT and TA.

4.1 Configuration Design & Training

Due to the need for a combination of VPT and TA to achieve adaptability and generalizability in highly difficult domains, we configure the trainable parameters to include multiple stacks of visual prompts, and a linear text adaptation layer following the pre-trained text encoder. While existing adapter-style methods (Zhang et al., 2022; Zhu et al., 2023b; Gao et al., 2023) rely on manually optimized text prompts for different datasets, we use learnable text prompts just for the input because manually creating prompt templates for each domain in the real world is challenging. The learnable text prompts are unnecessary if manual

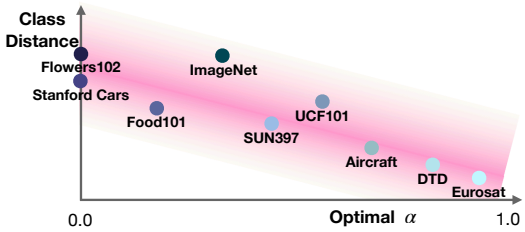

Figure 8: The relationship between class distance and optimal α for each domain used in Eq. (10) and Table 1.

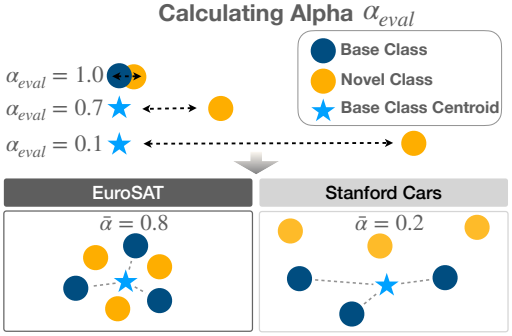

Figure 9: A concept figure for calculating the adaptive coefficient α_{eval} for ensemble upon its class distance.

prompts are already well-formed, which is further explained in Section 5.

We extract the visual feature $\mathbf{z}$ using Eq. (6) and Eq. (2) and the text feature $\mathbf{t}$ using Eq. (7) with $J_{\mathcal{T}} = 1$ and Eq. (9). We apply linear adapter parameterized as matrix $\mathbf{A}$ and bias $\mathbf{b}$ for **`TextAdapter`** in Eq. (9) rather than using bottleneck structure (Zhang et al., 2022; Gao et al., 2023) based on our results in Fig. 10. Our adapter can be formulated as follows:

$$\mathbf{t} = \texttt{TxtAdapt}(\tilde{\mathbf{t}}) := \mathbf{A}^\intercal \tilde{\mathbf{t}} + \mathbf{b} \qquad (11)$$

During the training procedure, our objective is to maximize the predicted probability $\Pr(y = y_{\text{gt}}|\mathbf{z}, \mathbf{t})$ for ground truth label y_{gt} by using cross-entropy loss $\ell_{\text{CE}}(\mathbf{z}, \mathbf{t}, y_{\text{gt}})$ which is defined as follows:

$$\ell_{\text{CE}}(\mathbf{z}, \mathbf{t}, y_{\text{gt}}) = \log \Pr(y = y_{\text{gt}}|\mathbf{z}, \mathbf{t}),$$

where the predicted probability is computed as Eq. (5).

4.2 Adaptive Ensemble for Evaluation

Due to the various levels of transfer difficulty encountered during deployment, an adaptive method is necessary to avoid suboptimal results for each target domain. Motivated by our observations, in the evaluation stage, we use an adaptive ensemble approach that combines pre-adapter ($\tilde{\mathbf{t}}_{\text{eval}}$) and

post-adapter text features (Eq. (11)), described as follows:

$$\mathbf{t}_{\text{eval}} = \alpha_{\text{eval}} \cdot (\mathbf{A}^{\intercal}\tilde{\mathbf{t}}_{\text{eval}} + \mathbf{b}) + (1 - \alpha_{\text{eval}}) \cdot \tilde{\mathbf{t}}_{\text{eval}},$$

where α_{eval} is the ensemble coefficient for a target class at evaluation and $\mathbf{t}_{\text{eval}}$ is the final representation for that class. With this ensemble approach, for domains with high RTD, the model relies on the adaptability and generalizability of VPT and TA. Conversely, for domains with low RTD, it leverages general knowledge from the pre-trained model to avoid excessive adaptation.

To determine the optimal α_{eval} for each class, which estimates transfer difficulty and acts as a controller for adaptation, we employ a non-parametric method based on the distance between the text features of the evaluation class and the classes learned during training. This approach is based on the assumption that in domains with high RTD, class features are typically less separable in the text embedding space, similarly to their separability in the image embedding space. Hence, domains like EuroSAT exhibit low class distances, while those with low RTD, such as Stanford Cars, display high class distances. Fig. 8 shows that the optimal α, used in Eq. (10) and Tab. 1, is highly correlated with the distance between class features. This tendency suggests that α_{eval} based on the distance between class features can effectively represent transfer difficulty.

Moreover, instead of applying a single α_{eval} for all classes, we adopt a class-wise approach. This is because, within the same domain, target features considered as out-of-task should rely more on the general knowledge of pre-trained VLMs, whereas features closer to the learned classes should leverage more task-specific knowledge. With regard to this, we adaptively set α_{eval} by comparing the text feature of the evaluation class with the features of the learned classes, as illustrated in Fig. 9. Specifically, we calculate both the average and nearest distances between the evaluation class and the C learned classes in the following manner:

$$d_{\text{eval}}^{\text{avg}} = 1.0 - \frac{1}{C}\sum_{j=1}^{C} \text{sim}(\mathbf{t'}_{\text{eval}}, \mathbf{t'}_j),$$
$$d_{\text{eval}}^{\text{nn}} = 1.0 - \min_{\forall j \in \{1,\ldots,C\}} \text{sim}(\mathbf{t'}_{\text{eval}}, \mathbf{t'}_j),$$

where $\mathbf{t'}_{\text{eval}}$ and $\mathbf{t'}_j$ indicate text feature of evaluation class and learned class $j \in \{1, \ldots, C\}$ from pre-trained VLMs and sim denotes cosine similarity. Using these distance metrics, we compute the coefficient α_{eval} as follows:

Table 2: Accuracy comparison on base-to-novel generalization of **APEX** with previous methods.

Dataset		CLIP	CLIP -Adapter	Co -CoOp	MaPLe	Pro -Grad	APEX
Average on 11 datasets	Base	69.34	83.23	81.11	82.52	82.55	**83.99**
	Novel	74.22	70.13	70.55	74.24	72.20	**76.76**
	HM	71.70	75.64	75.03	77.86	76.77	**80.04**
ImageNet	Base	72.43	76.06	76.47	77.02	76.97	**77.12**
	Novel	68.14	68.40	69.60	70.15	67.20	**71.10**
	HM	70.22	72.03	72.87	73.42	71.75	**73.99**
Caltech101	Base	96.84	98.00	97.70	97.95	97.88	**98.18**
	Novel	94.00	93.66	93.96	94.60	93.57	**95.06**
	HM	95.40	95.78	95.78	96.25	95.68	**96.59**
OxfordPets	Base	91.17	94.86	95.66	**95.80**	95.00	95.11
	Novel	97.26	94.49	96.32	**97.82**	97.46	97.27
	HM	94.12	94.67	95.99	**96.80**	96.21	96.18
Stanford Cars	Base	63.37	77.62	72.92	74.69	78.64	**80.53**
	Novel	74.89	68.53	71.98	73.53	70.23	**75.08**
	HM	68.65	72.79	72.45	74.11	74.20	**77.71**
Flowers102	Base	72.08	96.88	94.82	95.90	94.83	**97.47**
	Novel	**77.80**	69.20	70.71	72.96	74.70	77.58
	HM	74.83	80.73	81.01	82.87	83.57	**86.40**
Food101	Base	90.10	90.02	**90.63**	90.46	90.40	89.60
	Novel	91.22	89.76	91.13	91.71	90.43	**92.06**
	HM	74.83	89.89	90.88	**91.08**	90.41	90.81
FGVC Aircraft	Base	27.19	40.14	36.19	37.76	40.77	**42.69**
	Novel	**36.29**	31.77	26.82	34.67	30.16	35.21
	HM	31.09	35.47	30.81	36.15	34.67	**38.59**
SUN397	Base	69.36	**81.72**	80.55	81.33	81.19	81.17
	Novel	75.35	73.54	75.48	77.75	73.42	**78.98**
	HM	72.23	77.41	77.93	79.50	77.11	**80.06**
DTD	Base	53.24	81.77	77.34	79.34	76.64	**82.45**
	Novel	59.90	49.02	48.86	56.64	54.23	**63.80**
	HM	56.37	61.29	59.89	66.10	63.52	**71.94**
EuroSAT	Base	56.48	91.55	87.05	**93.00**	91.23	92.83
	Novel	64.05	61.10	61.27	69.17	68.58	**79.89**
	HM	60.03	73.29	71.92	79.33	78.30	**85.88**
UCF101	Base	70.53	**86.87**	82.86	84.43	84.54	86.74
	Novel	77.50	71.94	69.92	77.64	74.24	**78.37**
	HM	73.85	78.70	75.84	80.89	79.06	**82.34**

$$\alpha_{\text{eval}} = \exp\left(-\beta \cdot (d_{\text{eval}}^{\text{avg}}) \cdot \mathbf{1}_{(d_{\text{eval}}^{\text{nn}} > \epsilon)}\right),$$

where β is a scaling factor. The equation indicates a preference for pre-adapter features when the text feature distance from learned classes is large, and for trained TA when it is small. The condition of $d_{\text{eval}}^{\text{nn}} > \epsilon$, where ϵ is a small value set at 0.05, serves to treat an evaluation class that is very similar to the base class as identical. This adaptive α_{eval} enables flexible use of general and task-specific knowledge. Moreover, since text embeddings are usually pre-calculated (Radford et al., 2021), this adaptive coefficient incurs only a minor computational overhead.

Vision Ensemble. Additionally, to further improve the performance by leveraging more general knowledge of the pretrained VLMs, we can also employ an ensemble technique for the visual encoder that combines the visual feature of the pre-trained VLM ($\mathbf{z}'$) with the task-adapted VLMs ($\mathbf{z}$) as follows:

$$\mathbf{z} = \bar{\alpha} \cdot \mathbf{z}' + (1 - \bar{\alpha}) \cdot \mathbf{z},$$

$\bar{\alpha}$, the mean value of α_{eval}, is used for image ensemble since class-specific α_{eval} cannot be applied at the image level.

Table 3: Comparison of accuracy on cross-dataset of **APEX** with previous methods.

Dataset		C-Adapter	CoCoOp	MaPLe	ProGrad	APEX
Source	ImageNet	70.12	71.46	70.58	71.73	**72.00**
Target	Caltech101	92.94	93.24	93.46	93.30	**94.46**
	OxfordPets	86.80	**90.38**	90.28	89.95	90.06
	Cars	64.22	64.08	65.22	65.25	**65.46**
	Flower102	69.06	70.50	**71.80**	69.34	71.58
	Food101	85.20	85.64	86.24	86.22	**86.44**
	Aircraft	24.24	21.58	23.62	21.22	**24.44**
	SUN397	64.36	66.30	**67.32**	65.32	67.20
	DTD	43.44	43.68	45.04	42.19	**45.70**
	EuroSAT	**47.66**	45.48	46.24	45.33	47.58
	UCF101	65.52	67.42	68.26	67.62	**68.80**
Average		64.34	64.83	65.75	64.57	**66.16**

Table 4: Comparison of accuracy on domain generalization of **APEX** with previous methods.

	Source	Target				
	ImageNet	-V2	-S	-A	-R	Avg.
C-Adapter	70.12	61.78	46.70	48.56	74.00	57.76
CoCoOp	71.46	64.44	48.58	50.20	75.64	59.72
MaPLe	70.58	63.95	**48.78**	50.53	**76.78**	59.90
ProGrad	71.73	64.54	48.59	50.38	75.87	59.85
APEX	**72.00**	**64.70**	48.48	**50.68**	76.76	**60.16**

5 Experiments

We describe our experimental setup and results for verifying superiority of our method. Additional experimental results are described in Appendix C.

5.1 Experimental Setup

We evaluate **APEX** on the three most commonly used transfer learning tasks: base-to-novel generalization, cross-dataset evaluation, and domain generalization. For all the few-shot experiments except domain generalization, we follow CoCoOp (Zhou et al., 2022a) which uses 11 image recognition datasets. For the domain generalization, we use ImageNet (Deng et al., 2009) as a source dataset and use wide range of variants of ImageNet. We use multiple baselines for comparison with our methods in experiments. These include the standard zero-shot CLIP (Radford et al., 2021), CLIP-Adapter (Gao et al., 2023), CoCoOp (Zhou et al., 2022a) and MaPLe (khattak et al., 2023). We also consider ProGrad (Zhu et al., 2023a), which uses gradient alignment for prompt learning. We use the **average of 20 seeds** to determine the final value for base-to-novel and the **average of 5 seeds** for cross-evaluation and domain-generalization. More experimental details can be found in the Appendix A.

5.2 Main Results

Base-to-Novel Generalization. In this scenario, the datasets are evenly divided into base and novel categories. The model is trained on the base classes using 16 shots and is subsequently tested on both the base and novel classes. As indicated in Table 2, **APEX** consistently outperforms the best of the previous methods in average accuracy across all datasets, with a margin of 1∼6%. In particular, our method exhibits superior performance in novel classes on all datasets, demonstrating **APEX**'s enhanced generalizability. The exceptions are Oxford Pets and FGVC Aircraft, where the performance is already exceptionally high and low, respectively. This improvement is especially notable in domains with high RTD, such as EuroSAT (+15.84%) and DTD (+3.90%). Additionally, the **APEX** method also shows superior performance in base categories, highlighting the high adaptability of our approach.

Cross-dataset Evaluation. We train the model to generalize across different domains by using a cross-dataset evaluation task. Specifically, we first train the model on the ImageNet dataset and then transfer it to the 10 other datasets. Table 3 summarizes that **APEX** shows the best overall performance compared to existing baselines. Our proposed method achieves the best performance on 7 out of 11 tasks. This demonstrates **APEX**'s effectiveness, especially in difficult situations where both the task and domain are unseen.

Domain Generalization. We assess the capability of **APEX** to generalize to out-of-distribution data by training on the source dataset, ImageNet, and subsequently testing on various modified versions of ImageNet. Our method does not achieve a large margin of superiority since our adaptive ensemble is primarily designed to enhance performance in novel classes. Nonetheless, our method still surpasses all baseline models on average accuracy in this domain generalization task.

6 Conclusion

We propose **APEX** to address the challenges of conventional prompt and adapter-style tuning methods for ETL for VLMs. Our approach incorporates two key components based on our observations: (1) using VPT and TA for exploiting the property of each modality and (2) adaptive ensemble coefficient in the inference stage. We empirically demonstrate the superior performance of **APEX**, consistently achieving a better performance than the previous methods.

Limitation

We focus on two types of ETL, prompt tuning and adapter-style tuning, for VLMs for vision-language understanding tasks such as CLIP, EVA-CLIP, and CoCA-CLIP. While our extensive analyses provide valuable insights, our paper primarily centers on understanding tasks, with opportunities for further exploration in vision-language generation tasks such as BLIP (Li et al., 2022a) and LLaVA (Liu et al., 2024). Additionally, though we focus on two main representative ETL methods, further analyses could be conducted on other ETL methods like LoRA (Hu et al., 2022) and IA3 (Liu et al., 2022). We leave these aspects for future work but wish to emphasize the comprehensive exploration provided by our study on the two representative ETL methods for VLMs.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 - mining discriminative components with random forests. In *European Conference on Computer Vision*.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2013. Describing textures in the wild. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. Imagenet: a large-scale hierarchical image database. pages 248–255.

Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178.

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

Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. 2022. Cyclip: Cyclic contrastive language-image pretraining. *Advances in Neural Information Processing Systems*, 35:6704–6719.

Patrick Helber, Benjamin Bischke, Andreas R. Dengel, and Damian Borth. 2017. 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*, 12:2217–2226.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. 2020. The many faces of robustness: A critical analysis of out-of-distribution generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8320–8329.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Xiaodong Song. 2019. Natural adversarial examples. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15257–15266.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

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

Muhammad Uzair khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. 2023. Maple: Multi-modal prompt learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. 2023. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15190–15200.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022a. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.

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

Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. 2023. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400.

Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Xuejing Liu, Wei Tang, Jinghui Lu, Rui Zhao, Zhaojun Guo, and Fei Tan. 2023. Deeply coupled cross-modal prompt learning. *arXiv preprint arXiv:2305.17903*.

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

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. 2013. Fine-grained visual classification of aircraft. *ArXiv*, abs/1306.5151.

Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729.

Jaehoon Oh, Hyungjun Yoo, ChangHwan Kim, and Se-Young Yun. 2021. {BOIL}: Towards representation change for few-shot learning. In *International Conference on Learning Representations*.

Michal Pándy, Andrea Agostinelli, Jasper Uijlings, Vittorio Ferrari, and Thomas Mensink. 2022. Transferability estimation using bhattacharyya class separability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9172–9182.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. 2012. Cats and dogs. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Haohan Wang, Songwei Ge, Eric P. Xing, and Zachary Chase Lipton. 2019. Learning robust global representations by penalizing local predictive power. In *Neural Information Processing Systems*.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492.

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

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. 2022. FILIP: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

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

Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. 2022. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2022. Tip-adapter: Training-free adaption of clip for few-shot classification. In *Computer Vision – ECCV 2022*, pages 493–510, Cham. Springer Nature Switzerland.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348.

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

Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. 2023b. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2605–2615.

A Implementation Details

As explained in Section 5, we utilize the ViT-B/16 model as the CLIP image encoder and a standard GPT2-like structure with an End Of Text (EOT) token as the classification token for the text encoder. To implement **APEX**, we use visual prompts for all layers, setting $J_V = 12$ for base-to-novel generalization and $J_V = 3$ for cross-evaluation and domain generalization. The text prompt is applied only to the shallow prompt, and therefore, $J_V = 1$ for all experiments. The number of prompts for each layer, $b_{\mathcal{V}}$ and $b_{\mathcal{T}}$, is set to 2. The initial text prompt is fixed as *"a photo of a"*, and the visual prompts are initialized with a zero-mean Gaussian distribution with a standard deviation of 0.02. The matrix term of the text adapter is initialized with an identity matrix, and the bias vector is initialized with a zero vector.

The datasets cover multiple recognition tasks including ImageNet (Deng et al., 2009) and Caltech101 (Fei-Fei et al., 2004) which consists of generic objects; OxfordPets (Parkhi et al., 2012), Stanford Cars (Krause et al., 2013), Flowers102 (Nilsback and Zisserman, 2008), Food101 (Bossard et al., 2014), and FGVC Aircraft (Maji et al., 2013) for fine-grained classification, SUN397 (Xiao et al., 2010) for scene recognition, UCF101 (Soomro et al., 2012) for action recognition, DTD (Cimpoi et al., 2013) for texture classification, and EuroSAT (Helber et al., 2017) which consists of satellite images. For the domain generalization benchmark, we use ImageNet as a source dataset and use ImageNet-A (Hendrycks et al., 2019), ImageNet-R (Hendrycks et al., 2020), ImageNet-Sketch (Wang et al., 2019), and ImageNetV2 (Recht et al., 2019) as out-of-domain datasets.

For training, we use the Adadelta optimizer (Zeiler, 2012) with a learning rate of 0.15 and a cosine learning rate scheduler. The batch size is set to 16, and we train for 15 epochs, except for ImageNet, where we train for 5 epochs. As in previous works, we apply augmentation techniques of random cropping and flipping. The scaling factor β, used for calculating α_{eval}, is set to 4.0. In the SGD experiments presented in Appendix C, we adopt a batch size of 16 and epochs of 30 and 5 for ImageNet, along with a learning rate of 0.0015 and a cosine learning rate scheduler. The augmentation and scaling factors are set the same as in the Adadelta experiments.

For reproducing baselines, we use the Adadelta optimizer with a learning rate of 0.25, selected after a grid search with values $[0.1, 0.15, 0.2, 0.25, 0.3]$. The rest of the settings remain the same as in the original papers. Results with their original configurations using SGD optimizer are listed in Appendix C. All our experiments were conducted on a single NVIDIA RTX 3090.

B Notation and Algorithm

In this section, we present the notation and algorithm of our method, **APEX**. The notation is detailed in Table 5. The training algorithm for **APEX** is outlined in Algorithm 1, and the adaptive inference algorithm is presented in Algorithm 2.

Table 5: The notation table for Section 3

Notation	Description
The notation for VLMs	
$\mathcal{V}$	The visual encoder of VLMs
$\mathcal{T}$	The text encoder of VLMs
$L_{\mathcal{V}}$	The number of layers of visual encoder
$L_{\mathcal{T}}$	The number of layers of text encoder
$\mathcal{V}_\ell$	The ℓ^{th} Transformer layer of visual encoder
$\mathcal{T}_\ell$	The ℓ^{th} Transformer layer of text encoder
$\mathbf{E}_\ell$	The patch embeddings of ℓ^{th} layer of visual encoder
$\mathbf{W}_\ell$	The word embeddings of ℓ^{th} layer of text encoder
The inputs for VLMs or prompt tuning	
$J_{\mathcal{V}}$	The number of layers of VPT
$J_{\mathcal{T}}$	The number of layers of TPT
$b_{\mathcal{V}}$	The context length of VPT
$b_{\mathcal{T}}$	The context length of TPT
$\hat{\mathbf{P}}_\ell$	The visual prompt of ℓ^{th} layer of visual encoder
$\mathbf{P}_\ell$	The text prompt of ℓ^{th} layer of text encoder
The outputs for VLMs	
$\mathbf{c}_\ell$	The embedded features of ℓ^{th} layer for [CLS] token
$\mathbf{t}_i$	The text features of i^{th} class
$\mathbf{z}$	The visual features from visual encoder
The outputs for VLMs related to **APEX**	
$\mathbf{z}'$	The visual features from visual encoder of pretrained VLMs for adaptive ensemble
$\mathbf{t}'$	The text features from text encoder of pretrained VLMs for adaptive ensemble
$\tilde{\mathbf{t}}$	The pre-adapter text features of text encoder of adapted VLMs

Algorithm 1 Pseudo-Algorithm for Training of **APEX**

Require: Pretrained visual encoder $\mathcal{V}$, Pretrained text encoder $\mathcal{T}$, Learnable vision prompts $\hat{\mathbf{P}}$, Shallow text prompts $\mathbf{P_0}$, Adapter parameterized by matrix $\mathbf{A}$ and $\mathbf{b}$

Require: Training Samples $\mathcal{S}$, Initial Text Embeddings $\mathbf{W}_0$

Randomly initialize $\phi = [\hat{\mathbf{P}}, \mathbf{A}, \mathbf{b}]$
while not done **do**
  Sample Batch $\mathcal{B} = (I, y_{gt})$
  $\mathbf{E}_0 = \texttt{PathEmbedding(I)}$
  for $i = 1, \ldots, J_{\mathcal{V}}$ **do**
    $[\mathbf{c}_i, \mathbf{E}_i, \_\_] \leftarrow \mathcal{V}_i([\mathbf{c}_{i-1}, \mathbf{E}_{i-1}, \hat{\mathbf{P}}_{i-1}])$
  end for
  for $i = J_{\mathcal{V}} + 1, \ldots, L_{\mathcal{V}}$ **do**
    $[\mathbf{c}_i, \mathbf{E}_i, \hat{\mathbf{P}}_i] \leftarrow \mathcal{V}_i([\mathbf{c}_{i-1}, \mathbf{E}_{i-1}, \hat{\mathbf{P}}_{i-1}])$
  end for
  $\mathbf{z} \leftarrow \texttt{ImageProj}(\mathbf{c}_{L_{\mathcal{V}}})$
  $\tilde{\mathbf{t}} = \mathcal{T}([\mathbf{W_0}, \mathbf{P_0}])$
  $\mathbf{t} = \mathbf{A}^\intercal \tilde{\mathbf{t}} + \mathbf{b}$
  /* Calculate the probability for class i */
  $\Pr(y = i|\mathbf{z}, \mathbf{t}) = \frac{\exp(\text{sim}(\mathbf{z}, \mathbf{t}_i)/\tau)}{\sum_{j=1}^{C} \exp(\text{sim}(\mathbf{z}, \mathbf{t}_j)/\tau)}$
  $\ell_{\text{CE}}(\mathbf{z}, \mathbf{t}, y_{\text{gt}}) = \log \Pr(y = y_{\text{gt}}|\mathbf{z}, \mathbf{t})$
  $\phi = \phi - \gamma \nabla_\phi \ell_{\text{CE}}(\mathbf{z}, \mathbf{t}, y_{\text{gt}}; \phi)$
end while

C Additional Experiments

C.1 Ablation on Adaptive Ensemble

Table 6 illustrates the complete results of the component analysis of the adaptive ensemble. We only display results for novel classes, as these ensemble components do not affect the results for base classes, given that α_{eval} is set to 1.0 for seen classes. AThe ensemble of the text encoder is crucial as its removal leads to a significant performance drop in domains with low RTD, such as Stanford Cars and SUN397. This demonstrates that moderating TA with an adaptive ensemble helps to leverage both task-specific knowledge and general VLMs knowledge effectively. The ensemble on the visual encoder offers marginal improvement, but combining both still yields the most superior performance on average.

C.2 Results on Low-Rank Experiments

Figure 10 presents detailed results for each dataset using low-rank methods. The result demonstrates that our linear adapter provides better overall results, particularly for novel classes across most

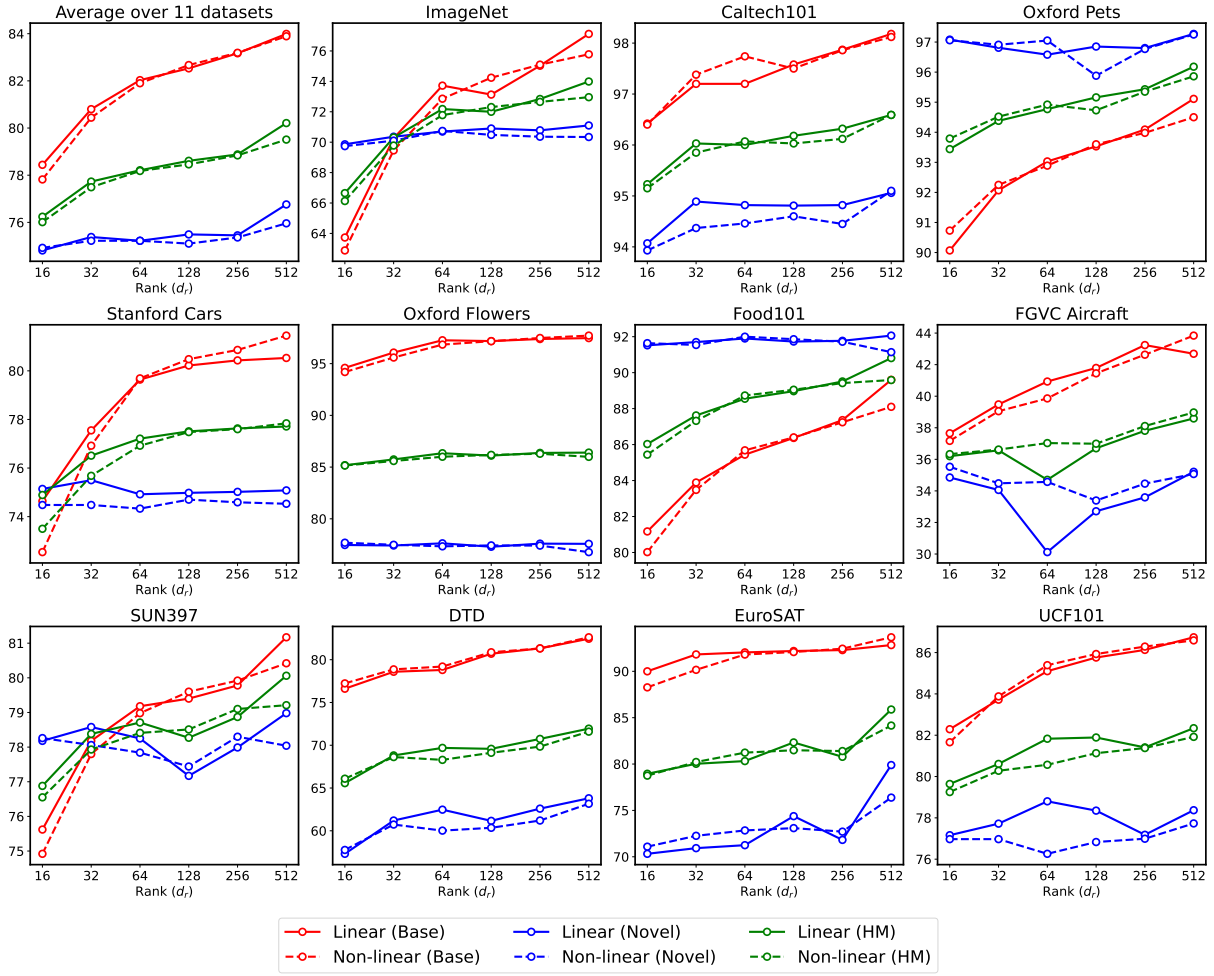

Figure 10: Results for the performance of the low-rank approach with different ranks.

Table 6: Comparison of the effect of adaptive ensemble technique between text and visual encoder by RTD.

Visual / **Text**	✗ / ✗	✗ / ✓	✓ / ✗	✓(**APEX**) / ✓(**APEX**)
ImageNet	69.08	70.09	69.22	**71.10**
Caltech101	94.91	94.80	95.01	**95.06**
OxfordPets	97.24	**97.39**	97.07	97.27
Cars	68.40	74.46	68.32	**75.08**
Flower102	73.71	76.40	74.43	**77.58**
Food101	90.70	91.83	90.82	**92.06**
Aircraft	33.97	33.89	33.87	**35.21**
SUN397	74.52	**78.98**	74.82	**78.98**
DTD	63.05	63.05	**63.82**	63.80
EuroSAT	77.73	79.04	78.25	**79.89**
UCF101	77.39	78.17	77.55	**78.37**
Average	74.61	76.19	74.83	**76.76**

datasets. This parameter-efficient approach exhibits relative robustness in performance, even outperforming MaPLe (khattak et al., 2023) for rank 64 (+0.32%) on average. These encouraging results have led us to adopt the linear adapter for the text encoder. Furthermore, we observe that initializing the adapter with an identity matrix improves performance, a strategy that can be explored more thoroughly in future work.

C.3 Full Results on Manual Text Prompts

Table 7 presents the detailed results for each dataset using manual prompts, which are summarized in Table 13. The manual prompts, designed for each dataset as described in (Gao et al., 2023; Zhang et al., 2022), appear to underperform compared to other methods. This suggests that they may not be the optimal choice for every dataset, and that designing these prompts manually is challenging. In contrast, just ensembling multiple manual prompts (Radford et al., 2021) works significantly better, indicating that optimal prompts may exist among these manual options. This finding also implies that utilizing improved manual prompts can substantially enhance performance, potentially replacing shallow prompts. Shallow prompt tuning for the text input yields the best results, demonstrat-

Table 7: Full results on each dataset of Table 13

		Average on 11 datasets	ImageNet	Caltech101	OxfordPets	Stanford Cars	Flowers102	Food101	FGVC Aircraft	SUN397	DTD	EuroSAT	UCF101
Opt. manual prompt (Zhang et al., 2022)	Base	**84.15**	76.64	98.15	95.05	**80.75**	97.45	89.35	**42.92**	81.24	**83.02**	**93.93**	87.10
	Novel	75.24	69.00	94.33	97.04	75.32	**77.66**	91.28	**36.42**	77.60	57.59	71.74	**79.70**
	HM	79.17	72.62	96.20	96.03	77.94	**86.44**	90.30	**39.40**	79.38	68.01	81.35	**83.24**
Ens. (60 manual prompts) (Radford et al., 2021)	Base	84.02	76.48	98.15	95.09	80.70	97.37	89.56	42.56	**81.46**	82.62	93.01	**87.18**
	Novel	76.17	70.24	93.93	96.44	**75.88**	77.16	91.20	35.64	78.36	59.45	**80.35**	79.21
	HM	79.70	73.23	95.99	95.76	**78.22**	86.09	90.37	38.79	79.88	69.15	**86.22**	83.00
Shallow prompt (**APEX**)	Base	83.99	**77.12**	**98.18**	**95.11**	80.53	**97.47**	**89.60**	42.69	81.17	82.45	92.83	86.74
	Novel	**76.76**	**71.10**	**95.06**	**97.27**	75.08	77.58	**92.06**	35.21	**78.98**	**63.80**	79.89	78.37
	HM	**80.04**	**73.99**	**96.59**	**96.18**	77.71	86.40	**90.81**	38.59	**80.06**	**71.94**	85.88	82.34

ing its effectiveness and flexibility. Therefore, we adopt this approach for our main results.

C.4 Baseline Results with SGD

Table 8 displays the reproduced results using the SGD optimizer, in contrast to the Adadelta optimizer presented in Table 2. As observed, the results with SGD are slightly lower compared to those with Adadelta. This difference is likely due to the adaptive learning rate of Adadelta, which facilitates training in this unstable few-shot scenario. Nonetheless, even with the SGD optimizer, our method significantly outperforms all baselines, particularly in domains with high RTD, maintaining the same trend observed with the Adadelta optimizer.

C.5 Comparison with More Baselines

Due to the page limit, we present a comparison with additional baselines for base-to-novel generalization experiments in Table 9, which are not included in Table 2. These include training with VPT, TPT, and a combination of VPT and TPT. We also compare our method with the recently proposed PromptSRC (Khattak et al., 2023), which employs various regularization techniques such as self-consistency loss and Gaussian averaging. Our method outperforms all these baselines in terms of harmonic mean and demonstrates particularly high performance for novel classes. Compared to PromptSRC, our method significantly outperforms in novel classes of high RTD domains, such as EuroSAT (+8.39%) and DTD (+4.22%), while maintaining comparable performance in other domains. Notably, our method achieves these results with a simpler training approach, without the need for numerous manual prompts for SRC loss, and with fewer hyperparameters, unlike the many required by PromptSRC's regularization techniques. Additionally, our method surpasses the simpler baselines of naive training using VPT, TPT, and their combination, highlighting the effectiveness of our configuration design and adaptive ensemble.

C.6 Ablation on Configuration

To further analyze the optimal configuration in combination with an adaptive ensemble, we conduct additional ablation studies on configurations. The results, present in Table 10, show that utilizing VPT and TA yields the best outcomes, confirming their effectiveness when paired with the adaptive ensemble. However, adding TPT to VPT and TA does not enhance performance, especially in high RTD scenarios, as evidenced by decreased performance in DTD (-4.98%) and EuroSAT (-6.78%) compared to configurations without TPT. While combining TPT with VA demonstrates reasonable performance, it is not as effective as the combination of VPT and TA. This highlights the importance of class separability of visual features achieved through multiple stacks of prompts. Overall, the configuration of **APEX** outperforms the other setups.

C.7 Ablation on β

Table 11 presents the results of an ablation study on the hyperparameter β, which is used to calculate α_{eval}. A higher β leads to a lower α_{eval}, indicating greater reliance on the general knowledge of VLMs, which is beneficial for domains with low RTD, and vice versa. As observed, the performance in domains with low RTD, such as Stanford Cars and SUN397, tends to improve with a higher β. However, the optimal performance for difficult domains like Aircraft and DTD is achieved with β values between 1.0 and 3.0. Not all domains follow this tendency since α_{eval} is calculated on a class-wise basis, as demonstrated in the case of EuroSAT. Interestingly, except for the value of 2.0, our method demonstrates robustness to variations in β, as it does not significantly affect the average performance. Overall, setting β to 4.0 yields the best performance, and therefore, this value has

Table 8: Comparison of baselines using their own configuration (SGD optimizer) with our method.

Dataset		CLIP	CLIP -Adapter	Co -CoOp	MaPLe	Pro -Grad	APEX
Average on 11 datasets	Base	69.34	81.81	80.28	81.74	81.78	**84.04**
	Novel	74.22	71.43	72.03	73.89	69.42	**75.67**
	HM	71.70	75.93	75.60	77.30	74.80	**79.42**
ImageNet	Base	72.43	74.40	75.99	76.81	**76.93**	**76.93**
	Novel	68.14	68.63	70.39	**70.66**	69.51	69.61
	HM	70.22	71.40	73.08	**73.61**	73.03	73.09
Caltech101	Base	96.84	97.61	97.64	95.61	95.41	**98.18**
	Novel	94.00	93.72	94.52	94.71	94.05	**95.02**
	HM	95.40	95.63	96.05	96.18	95.90	**96.57**
OxfordPets	Base	91.17	95.06	95.56	**95.61**	95.41	95.21
	Novel	97.26	95.02	97.52	97.63	90.56	**97.74**
	HM	94.12	95.04	96.53	**96.61**	92.92	96.46
Stanford Cars	Base	63.37	76.18	70.97	72.49	77.41	**80.44**
	Novel	**74.89**	69.30	73.44	73.46	70.92	74.76
	HM	68.65	72.58	72.18	72.97	74.02	**77.50**
Flowers102	Base	72.08	96.27	93.88	95.49	95.34	**97.73**
	Novel	**77.80**	69.92	72.56	72.55	76.84	76.67
	HM	74.83	81.01	81.85	82.45	85.10	**85.93**
Food101	Base	90.10	90.32	**90.54**	90.50	90.17	89.46
	Novel	91.22	90.10	91.15	91.71	85.53	**91.94**
	HM	74.83	90.21	90.84	**91.10**	87.79	90.68
FGVC Aircraft	Base	27.19	38.87	33.64	36.33	39.01	**42.96**
	Novel	**36.29**	31.95	26.49	32.64	27.77	34.72
	HM	31.09	35.07	29.64	34.39	32.44	**38.40**
SUN397	Base	69.36	76.50	79.86	80.65	**81.35**	81.18
	Novel	75.35	74.60	76.51	**78.33**	69.06	77.08
	HM	72.23	75.54	78.15	**79.47**	74.70	79.08
DTD	Base	53.24	80.46	76.58	79.20	77.45	**82.19**
	Novel	59.90	52.79	53.47	55.01	51.63	**61.21**
	HM	56.37	63.75	62.97	64.92	61.96	**70.17**
EuroSAT	Base	56.48	88.48	86.18	90.38	84.88	**93.48**
	Novel	64.05	67.12	63.04	68.43	56.66	**75.88**
	HM	60.03	76.33	72.82	77.89	67.96	**83.77**
UCF101	Base	70.53	85.81	82.22	84.02	83.82	**86.71**
	Novel	77.50	72.55	73.22	77.62	71.13	**77.77**
	HM	73.85	78.62	77.46	80.69	76.96	**82.00**

been selected for the final results.

C.8 Ablation on α

Table 12 presents the comprehensive results of the ablation study on a fixed α, which is used in Table 1 and Eq. (10). The same α is applied uniformly across all classes and is set as a fixed value for both the visual and text encoders. This is done to determine the correlation between α and the domain, along with its transfer difficulty. Similar to Section C.7, domains with high RTD, such as EuroSAT, require a higher α value to perform well compared to domains with low RTD, like Stanford Cars. These findings support the necessity for an adaptive ensemble that is closely aligned with RTD.

C.9 Shallow Prompt

Although we observe that TPT leads to overfitting, we employ one-layer learnable text prompts to enhance real-world practicality. Table 13 compares the performance of manually optimized prompts (Gao et al., 2023; Zhang et al., 2022), the ensemble of manual prompts (Radford et al., 2021), and shallow prompts. The shallow prompt method outperforms manual prompts, proving its effectiveness. However, manual prompts, particularly when ensembled, also show comparable performance to shallow prompts, suggesting that well-designed manual prompts can be an effective alternative.

Table 9: Extended baselines not presented in Table 2 for comparison between base-to-novel experiments with our method.

Dataset		CLIP	VPT	TPT	VPT + TPT	Prompt -SRC	APEX
Average on 11 datasets	Base	69.34	81.01	82.07	82.93	**84.36**	83.99
	Novel	74.22	73.11	73.90	74.15	75.37	**76.76**
	HM	71.70	76.55	77.51	78.00	79.39	**80.04**
ImageNet	Base	72.43	75.94	76.81	77.18	**77.90**	77.12
	Novel	68.14	68.74	69.45	69.86	70.26	**71.10**
	HM	70.22	72.16	72.94	73.34	73.88	**73.99**
Caltech101	Base	96.84	97.79	97.84	97.98	97.81	**98.18**
	Novel	94.00	93.65	94.29	94.38	93.88	**95.06**
	HM	95.40	95.68	96.03	96.15	95.80	**96.59**
OxfordPets	Base	91.17	95.11	95.48	**95.78**	95.69	95.11
	Novel	97.26	96.57	97.52	**97.65**	97.42	97.27
	HM	94.12	95.83	96.49	**96.71**	96.55	96.18
Stanford Cars	Base	63.37	70.72	75.18	75.75	80.16	**80.53**
	Novel	74.89	72.78	72.73	73.02	74.52	**75.08**
	HM	68.65	71.74	73.93	74.36	77.24	**77.71**
Flowers102	Base	72.08	91.60	96.45	96.26	96.96	**97.47**
	Novel	**77.80**	69.62	74.69	72.62	76.73	77.58
	HM	74.83	79.11	84.19	82.79	85.67	**86.40**
Food101	Base	90.10	90.17	90.30	90.36	**90.60**	89.60
	Novel	91.22	90.94	91.42	91.58	91.38	**92.06**
	HM	90.66	90.55	90.86	90.97	**90.99**	90.81
FGVC Aircraft	Base	27.19	34.70	37.86	38.76	**43.67**	42.69
	Novel	36.29	33.53	34.17	35.08	**36.42**	35.21
	HM	31.09	34.10	35.92	36.83	**39.72**	38.59
SUN397	Base	69.36	79.09	81.70	81.57	**82.94**	81.17
	Novel	75.35	76.85	77.62	77.92	78.37	**78.98**
	HM	72.23	77.95	79.61	79.70	**80.59**	80.06
DTD	Base	53.24	78.67	79.81	80.81	82.21	**82.45**
	Novel	59.90	53.78	55.32	55.64	59.58	**63.80**
	HM	56.37	63.89	65.35	65.90	69.09	**71.94**
EuroSAT	Base	56.48	**94.17**	86.98	92.91	93.06	92.83
	Novel	64.05	73.26	69.16	71.19	71.60	**79.89**
	HM	60.03	82.41	77.05	80.61	80.93	**85.88**
UCF101	Base	70.53	83.10	84.38	84.92	**87.05**	86.74
	Novel	77.50	74.52	76.54	76.75	**78.96**	78.37
	HM	73.85	78.58	80.27	80.63	**82.81**	82.34

C.10 Results on Different VLMs

We validate our approach using different backbones: EVA-CLIP (Sun et al., 2023) and CoCa (Yu et al., 2022). Table 14 displays the results using these two backbones, where we compare our method with both zero-shot and naive prompt tuning approaches that combine VPT and TPT. As observed, **APEX** consistently outperforms the aver-

Table 10: Results for additional ablation study on configurations when combined with adaptive ensemble.

		Average on 11 datasets	ImageNet	Caltech101	OxfordPets	Stanford Cars	Flowers102	Food101	FGVC Aircraft	SUN397	DTD	EuroSAT	UCF101
TPT + VA	Base	83.51	76.43	98.00	94.76	79.68	97.28	89.24	42.27	80.96	81.49	92.27	86.24
	Novel	75.88	69.43	94.49	97.21	**75.77**	77.50	91.50	34.85	78.20	62.05	76.77	76.90
	HM	79.32	72.76	96.21	95.97	77.68	86.27	90.36	38.20	79.56	70.45	83.81	81.30
VPT + TA + TPT	Base	83.56	76.93	98.03	94.77	79.45	**97.51**	89.26	42.14	81.02	81.72	92.11	86.21
	Novel	75.09	**71.30**	94.72	**97.76**	72.98	76.70	91.94	33.80	78.08	58.82	73.11	76.80
	HM	78.85	**74.01**	96.35	**96.24**	76.08	85.86	90.58	37.51	79.52	68.40	81.52	81.23
VPT + TA (APEX)	Base	**83.99**	**77.12**	**98.18**	**95.11**	**80.53**	97.47	**89.60**	**42.69**	**81.17**	**82.45**	**92.83**	**86.74**
	Novel	**76.76**	71.10	**95.06**	97.27	75.08	**77.58**	**92.06**	**35.21**	**78.98**	**63.80**	**79.89**	**78.37**
	HM	**80.04**	73.99	**96.59**	96.18	**77.71**	**86.40**	**90.81**	**38.59**	**80.06**	**71.94**	**85.88**	**82.34**

Table 11: Results for additional ablation study on scaling factor β. Our proposed methods shows robust performance on the selection of β.

β	Average on 11 datasets	ImageNet	Caltech101	OxfordPets	Stanford Cars	Flowers102	Food101	FGVC Aircraft	SUN397	DTD	EuroSAT	UCF101
1.0	75.97	70.62	95.15	97.43	72.15	75.95	91.38	35.07	77.02	63.90	78.36	78.66
2.0	76.51	71.06	95.14	**97.44**	73.95	77.06	91.70	35.35	78.12	63.99	78.89	**78.92**
3.0	76.75	**71.18**	95.15	97.37	74.69	77.61	91.92	**35.46**	78.66	**64.17**	79.35	78.64
4.0 **(APEX)**	**76.76**	71.10	95.06	97.27	75.08	77.58	**92.06**	35.21	**78.98**	63.80	79.89	78.37
5.0	76.72	71.00	**95.16**	97.18	75.10	77.79	91.96	35.05	78.96	63.77	79.88	78.07
6.0	76.66	70.96	**95.16**	97.15	**75.17**	**77.80**	91.98	34.84	78.92	63.54	**80.01**	77.75

age results in terms of harmonic mean, regardless of the model used. Specifically, with EVA-CLIP, our method demonstrates superior performance for both base and novel classes. In the case of the most challenging domain, EuroSAT, our method significantly enhances performance compared to the zero-shot accuracy for novel classes (+18.46%). A similar improvement of 8.85% on EuroSAT is observed with CoCa. However, in terms of novel classes, the average performance of zero-shot tuning is superior for CoCa. This could be attributed to the larger patch size of this backbone, which might increase the risk of overfitting on the vision side when setting two learnable prompts. Nonetheless, our method shows comparable performance on novel classes to zero-shot CoCa, with a significant improvement in base classes. This results in superior performance in harmonic mean, demonstrating our method's effectiveness across various VLMs.

D Details about Observation

D.1 Relative Transfer Difficulty

Here, we report the value of RTD which is defined in Sectionn 3 for 11 transfer datasets. We compute the RTD based on the CLIP-B/16 model.

D.2 Inter- and Intra-class Cosine Similarity

In addition to presenting relative values in Figure 5, we also report the absolute values for both inter- and intra-class similarities. We observe a significant correlation between the RTD and the ratio of intra- to inter-class similarity.

D.3 Results on 6 datasets

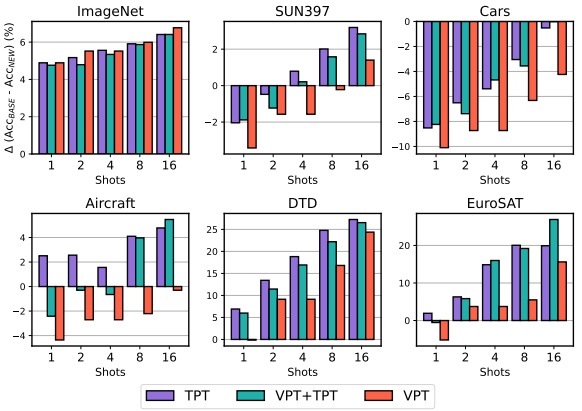

Figure 11: Extended results for Figure 2. All results in different datasets show similar trends that indicate VPT yields a smaller discrepancy in performance between base and novel categories, suggesting a reduced risk of overfitting compared to TPT.

We also present extended results in Figure 11, which include data from three additional datasets: ImageNet, SUN397, and DTD. For ImageNet and SUN397, which already exhibit high class separability, we note that all methods—TPT, VPT, and their combination—yield similar performance differences. However, the results for DTD indicate

Table 12: Extended results for ablation study on hyperparamter α related to Table 1.

α	Average on 11 datasets	ImageNet	Caltech101	OxfordPets	Stanford Cars	Flowers102	Food101	FGVC Aircraft	SUN397	DTD	EuroSAT	UCF101
0.0	75.38	70.80	95.13	97.03	**75.19**	**77.87**	91.94	33.57	78.32	61.68	70.90	76.80
0.1	75.86	71.06	**95.19**	97.19	75.17	77.67	**92.10**	34.34	78.82	62.68	72.74	77.52
0.2	76.10	**71.20**	95.14	97.29	75.04	77.52	91.96	34.75	78.80	63.18	74.10	78.08
0.3	76.27	**71.20**	95.09	97.39	74.67	77.33	91.92	35.16	**78.90**	63.74	75.08	78.54
0.4	**76.34**	71.18	95.14	97.47	74.22	76.96	91.88	35.34	78.68	64.16	75.87	78.84
0.5	76.29	71.04	95.15	**97.50**	73.59	76.56	91.78	**35.45**	78.40	64.32	76.41	**79.01**
0.6	76.13	70.82	95.14	97.47	72.74	76.13	91.64	35.33	78.00	**64.30**	76.95	78.96
0.7	75.88	70.46	95.17	97.39	71.82	75.66	91.44	35.25	77.38	64.23	77.10	78.79
0.8	75.54	70.06	95.07	97.36	70.85	75.09	91.22	34.93	76.56	64.04	77.33	78.39
0.9	75.10	69.62	95.01	97.31	69.63	74.49	90.98	34.53	75.68	63.53	77.44	77.92
1.0	74.61	69.08	94.91	97.24	68.40	73.71	90.70	33.97	74.52	63.05	**77.73**	77.39

Table 13: Comparison of the accuracy of the base, novel, and their harmonic means among the various prompt types on text encoder.

Prompt	Base Acc.	Novel Acc.	HM
Opt. manual prompt (Zhang et al., 2022)	**84.15**	75.24	79.17
Ens. (60 manual prompts (Radford et al., 2021))	84.02	76.17	79.70
Shallow prompt	83.99	**76.76**	**80.04**

a tendency for TPT to overfit to the base classes. This observation is consistent with the findings presented in Figure 2.

E More Related Work

Vision-Language Models VLMs overcome the limitations of vision-only supervised learning with their robustness and flexibility in zero-shot inference through natural language supervision. CLIP (Radford et al., 2021) facilitates this by adopting contrastive learning with a large-scale dataset of 400 million images. ALIGN(Jia et al., 2021) further improves upon this by scaling up the dataset with more noisy image-text pairs. FILIP (Yao et al., 2022) enables finer-grained alignment between two modalities and GLIP (Li et al., 2022b) improves visual grounding and object detection using VLMs. CoCa (Yu et al., 2022) employs both captioning and contrastive losses, thereby integrating the model capabilities of contrastive approaches like CLIP with those of generative methods. CyCLIP (Goel et al., 2022) employs cyclic loss to ensure geometric consistency, while FLIP (Li et al., 2023) enhances VLMs through masking techniques. EVA-CLIP (Sun et al., 2023) implements various training techniques, such as different attention mechanisms and optimizers, to further improve CLIP's performance. Additionally, SigLIP (Zhai et al., 2023) replaces the softmax loss with sigmoid loss, enabling more efficient pretraining with smaller batch sizes.

There is also a line of research focused on encoder-decoder or decoder-only architectures. BLIP (Li et al., 2022a) facilitates both encoding and decoding by training with three objective functions, utilizing synthetic data and data filtering. ALBEF (Li et al., 2021) employs a strategy of alignment before applying cross-attention, combined with a momentum update. Flamingo (Alayrac et al., 2022) enables few-shot inference in vision-language tasks through architectural innovations, using vision-language prompts.

Prompt Tuning Efficient tuning using soft prompts, originating in the domain of natural language processing, has gained a lot of attention (Lester et al., 2021). This approach has also been applied in the vision-language domain to adapt to downstream tasks. CoOp (Zhou et al., 2022b) was the first to apply learnable prompts for CLIP model, replacing manual prompts for each domain. ProDA (Lu et al., 2022) observes that these text prompts can be viewed as a distribution and proposes prompt distributional learning for higher quality results. CoCoOp (Zhou et al., 2022a) conditions text prompts on images to prevent overfitting to base classes. KgCoOp (Yao et al., 2023) regularizes by minimizing the discrepancy between learned and manual prompts. UPT (Zang et al., 2022) examines both VPT (Jia et al., 2022) and text prompts, proposing a unified approach to generate visual and textual prompts from the same architecture. MaPLe (khattak et al., 2023) employs the alignment of visual and text prompts for improvement with deep prompts, while DCP (Liu et al., 2023) uses an attention mechanism for this alignment. There is also a line of research aimed at preventing the forgetting of general knowledge. ProGrad (Zhu et al., 2023a) aligns gradient direc-

Table 14: Accuracy on base-to-novel generalization of **APEX** on EVA-CLIP (Sun et al., 2023) and CoCa (Yu et al., 2022).

Model		EVA-CLIP-B/16			CoCa-B/32		
Dataset		**ZS**	**TPT +VPT**	**APEX**	**ZS**	**TPT +VPT**	**APEX**
Average on 11 datasets	Base	75.28	85.91	**85.93**	70.85	**82.39**	82.09
	Novel	77.68	75.24	**79.34**	**74.29**	71.05	73.98
	HM	76.46	80.22	**82.50**	72.53	76.30	**77.87**
ImageNet	Base	79.20	**81.78**	81.26	67.10	**69.50**	69.46
	Novel	75.60	72.28	**75.83**	**66.60**	62.33	66.46
	HM	77.36	76.74	**78.45**	66.85	65.72	**67.90**
Caltech101	Base	98.60	**98.87**	98.82	96.70	97.86	**98.04**
	Novel	**97.30**	95.05	97.22	**96.30**	94.12	95.98
	HM	97.95	96.92	**98.01**	96.50	95.95	**97.00**
OxfordPets	Base	94.90	**95.52**	95.27	92.30	91.83	**92.44**
	Novel	98.10	**98.34**	97.97	**96.20**	95.07	93.54
	HM	96.47	**96.91**	96.60	**94.21**	93.42	92.99
Stanford Cars	Base	76.90	85.76	**86.16**	84.00	**88.94**	88.87
	Novel	**87.10**	82.49	86.75	**93.00**	90.73	92.57
	HM	81.68	84.09	**86.45**	88.27	89.83	**90.68**
Flowers102	Base	74.20	99.41	**99.50**	69.10	96.33	**96.83**
	Novel	**81.10**	77.32	79.94	**74.70**	65.61	70.09
	HM	77.50	86.98	**88.65**	71.79	78.06	**81.32**
Food101	Base	90.30	**90.34**	90.24	**81.20**	79.87	80.80
	Novel	**91.90**	90.11	91.76	**82.90**	79.30	82.66
	HM	**91.09**	90.22	90.99	**82.04**	79.58	81.72
FGVC Aircraft	Base	28.70	45.52	**46.01**	21.40	**40.71**	39.81
	Novel	**32.50**	26.75	32.12	**25.50**	22.04	25.22
	HM	30.48	33.70	**37.83**	23.27	28.60	**30.88**
SUN397	Base	76.70	**83.10**	82.44	73.70	**78.68**	77.68
	Novel	**80.80**	76.76	80.54	**77.40**	73.50	77.12
	HM	78.70	79.80	**81.48**	75.50	76.00	**77.40**
DTD	Base	62.80	83.78	**84.15**	62.60	83.04	**83.25**
	Novel	63.90	61.32	**64.39**	61.10	58.46	**61.14**
	HM	63.35	70.81	**72.96**	61.84	68.62	**70.50**
EuroSAT	Base	72.30	**95.32**	94.81	62.80	**96.42**	93.87
	Novel	68.30	73.74	**86.76**	71.50	73.90	**80.35**
	HM	70.24	83.15	**90.61**	66.87	83.67	**86.59**
UCF101	Base	73.50	85.58	**86.58**	68.50	**83.13**	82.01
	Novel	77.90	73.43	**79.49**	**72.00**	66.54	69.69
	HM	75.64	79.04	**82.88**	70.21	73.92	**74.76**

tions to preserve general knowledge, and PromptSRC (Khattak et al., 2023) utilizes multiple regularization losses with Gaussian aggregation of model weights to prevent forgetting.

Adapter-style Tuning Adapter-style tuning has been extensively explored as an alternative to prompt tuning. CLIP-Adapter (Gao et al., 2023) was the first proposed method in this area, utilizing a two-layer MLP structure with ReLU nonlinearity in between. Additionally, it incorporates a residual connection to preserve general knowledge. For improved efficiency, Tip-Adapter (Zhang et al., 2022) employs a cache-based model to save the features and labels of few-shot samples, using them to predict test outcomes without further training. This approach also facilitates better fine-tuning by using the cache as initial training points for further refinement. Differently, Task Residual (Yu et al., 2023) adopts a unique strategy by simply adding a residual or bias term vector for each class, reducing reliance on pre-trained features. Zhu et al. (2023b) enhances cache-based models through prior refinement, which involves selecting important features for the cache-based model.

Table 15: The relative transfer difficulty values for all datasets by using Definition 1.

Dataset	**ImageNet**	**Caltech**	**Pets**	**Cars**
RTD	1.4×10^{-3}	1.08×10^{-2}	3.01×10^{-2}	7.7×10^{-3}
Dataset	**Flowers**	**Food**	**Aircraft**	**SUN**
RTD	1.52×10^{-2}	1.15×10^{-2}	4.07×10^{-2}	3.8×10^{-3}
Dataset	**DTD**	**EuroSAT**	**UCF**	
RTD	4.95×10^{-3}	1.84×10^{-1}	1.42×10^{-2}	

Table 16: The averaged cosine similarity value for inter- and intra-class and their relative ratio.

Dataset	**ImageNet**	**Caltech**	**Pets**	**Cars**	**Flowers**	**Food**
Inter	0.551	0.672	0.844	0.564	0.749	0.754
Intra	0.925	0.898	0.910	0.829	0.924	0.853
Ratio	1.680	1.336	1.078	1.470	1.234	1.131
Dataset	**Aircraft**	**SUN**	**DTD**	**EuroSAT**	**UCF**	
Inter	0.746	0.487	0.803	0.896	0.673	
Intra	0.858	0.780	0.823	0.934	0.866	
Ratio	1.150	1.602	1.025	1.042	1.287	

Algorithm 2 Pseudo-Algorithm for Adaptive Inference of **APEX**

Require: Pretrained visual encoder $\mathcal{V}$, Pretrained text encoder $\mathcal{T}$, Learned vision prompts $\hat{\mathbf{P}}$, Learned shallow text prompts $\mathbf{P_0}$,Learned adapter parameterized by matrix $\mathbf{A}$ and $\mathbf{b}$, The C classes for base category $\{1, \ldots, C\}$, The C_{eval} candidate classes for evaluation $\{C + 1, \ldots, C + C_{\text{eval}}\}$,

Require: Initial Trained Text Embeddings $\{\mathbf{W}_{0,j}\}_{j=1}^{C}$, Initial Evaluation Text Embedding $\{\mathbf{W}_{0,\text{eval}}\}_{\text{eval}=C+1}^{C+C_{\text{eval}}}$, Evaluation Image I

$\{\mathbf{t'}_j\}_{j=1}^{C} = \{\mathcal{T}(\mathbf{W}_{0,j})\}_{j=1}^{C}$
for eval $= C + 1, \ldots, C + C_{\text{eval}}$ **do**
  $\mathbf{t'}_{\text{eval}} = \mathcal{T}(\mathbf{W}_{0,\text{eval}})$
  $\tilde{\mathbf{t}}_{\text{eval}} = \mathcal{T}([\mathbf{W}_{0,\text{eval}}, \mathbf{P_0}])$
  $d_{\text{eval}}^{\text{avg}} = 1.0 - \frac{1}{C}\sum_{j=1}^{C} \text{sim}(\mathbf{t'}_{\text{eval}}, \mathbf{t'}_j)$
  $d_{\text{eval}}^{\text{nn}} = 1.0 - \min_{\forall j \in \{1,\ldots,C\}} \text{sim}(\mathbf{t'}_{\text{eval}}, \mathbf{t'}_j)$
  $\alpha_{\text{eval}} = \exp\left(-\beta \cdot (d_{\text{eval}}^{\text{avg}}) \cdot \mathbf{1}_{(d_{\text{eval}}^{\text{nn}} > \epsilon)}\right)$
  $\mathbf{t}_{\text{eval}} = \alpha_{\text{eval}} \cdot (\mathbf{A}^{\top}\tilde{\mathbf{t}}_{\text{eval}} + \mathbf{b}) + (1 - \alpha_{\text{eval}}) \cdot \tilde{\mathbf{t}}_{\text{eval}}$
end for
$\mathbf{E}_0 = \texttt{PathEmbedding}(I)$
$\mathbf{c'}_{L_{\mathcal{V}}} = \mathcal{V}([\mathbf{c}_0, \mathbf{E_0}])$
$\mathbf{z'} \leftarrow \texttt{ImageProj}(\mathbf{c'}_{L_{\mathcal{V}}})$
for $i = 1, \ldots, J_{\mathcal{V}}$ **do**
  $[\mathbf{c}_i, \mathbf{E}_i, \_\_] \leftarrow \mathcal{V}_i([\mathbf{c}_{i-1}, \mathbf{E}_{i-1}, \hat{\mathbf{P}}_{i-1}])$
end for
for $i = J_{\mathcal{V}} + 1, \ldots, L_{\mathcal{V}}$ **do**
  $[\mathbf{c}_i, \mathbf{E}_i, \hat{\mathbf{P}}_i] \leftarrow \mathcal{V}_i([\mathbf{c}_{i-1}, \mathbf{E}_{i-1}, \hat{\mathbf{P}}_{i-1}])$
end for
$\mathbf{z} \leftarrow \texttt{ImageProj}(\mathbf{c}_{L_{\mathcal{V}}})$
$\bar{\alpha} = \frac{1}{C_{\text{eval}}} \sum_{\text{eval}=C+1}^{C+C_{\text{eval}}} \alpha_{\text{eval}}$
$\mathbf{z} = \bar{\alpha} \cdot \mathbf{z'} + (1 - \bar{\alpha}) \cdot \mathbf{z}$
/* Calculate the probability for class i */
Calculate $\Pr(y = i|\mathbf{z}, \mathbf{t}) = \frac{\exp(\text{sim}(\mathbf{z}, \mathbf{t}_i)/\tau)}{\sum_{j=C+1}^{C+C_{\text{eval}}} \exp(\text{sim}(\mathbf{z}, \mathbf{t}_j)/\tau)}$
Predict as $\arg\max_{i \in \{C+1,\ldots,C+C_{\text{eval}}\}} \Pr(y = i|\mathbf{z}, \mathbf{t})$