Adaptive Parameter Selection for Tuning Vision-Language Models

Yi Zhang[1*] Yi-Xuan Deng[3] Meng-Hao Guo[2] Shi-Min Hu[2†]
[1]State Key Laboratory of Virtual Reality Technology and Systems, SCSE, Beihang University
[2]BNRist, Department of Computer Science and Technology, Tsinghua University
[3] Zhili College, Tsinghua University

Abstract

Vision-language models (VLMs) like CLIP have been widely used in various specific tasks. Parameter-efficient fine-tuning (PEFT) methods, such as prompt and adapter tuning, have become key techniques for adapting these models to specific domains. However, existing approaches rely on prior knowledge to manually identify the locations requiring fine-tuning. Adaptively selecting which parameters in VLMs should be tuned remains unexplored. In this paper, we propose CLIP with Adaptive Selective Tuning (CLIP-AST), which can be used to automatically select critical parameters in VLMs for fine-tuning for specific tasks. It opportunely leverages the adaptive learning rate in the optimizer and improves model performance without extra parameter overhead. We conduct extensive experiments on 13 benchmarks, such as ImageNet, Food101, Flowers102, etc, with different settings, including few-shot learning, base-to-novel class generalization, and out-of-distribution. The results show that CLIP-AST consistently outperforms the original CLIP model as well as its variants and achieves state-of-the-art (SOTA) performance in all cases. For example, with the 16-shot learning, CLIP-AST surpasses GraphAdapter and PromptSRC by 3.56% and 2.20% in average accuracy on 11 datasets, respectively. Code will be publicly available.

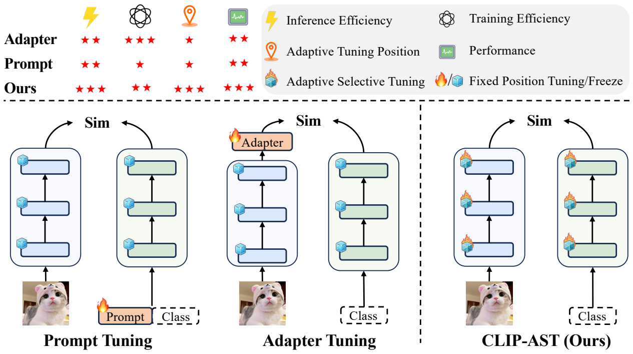

Figure 1. Comparison of fine-tuning methods across key dimensions. Prompt tuning and adapter tuning rely on prior knowledge to define positions for fine-tuning. In contrast, our method performs adaptive selective tuning. It automatically identifies the most critical positions for tuning. Additionally, no extra parameters are introduced, improving efficiency and performance.

1. Introduction

With the rise of large-scale image-text pretraining, vision-language models (VLMs) [8, 29, 38, 44, 45, 54] have shown strong capabilities in multimodal representation. These models learn the distribution between visual and textual features by jointly training on large-scale image and text paired data, which enables them to perform well across specific tasks. For example, CLIP [38], trained with hundreds of millions of image-text pairs using the contrastive learning framework, aligns image and text representations in a high-dimensional feature space, which gives CLIP strong generalization capabilities and makes it widely used in various applications [7, 10, 24, 43, 47, 51, 56, 58, 61].

While CLIP has shown strong general capabilities, its performance often remains suboptimal on specific tasks without task-oriented fine-tuning. However, performing full fine-tuning on large pre-trained models often introduces challenges, including increased computational cost, potential overfitting, and reduced transferability across diverse tasks and domains. Parameter-efficient fine-tuning (PEFT) [2, 4, 15, 27] has been widely adopted to mitigate the computational cost and overfitting risks associated with the full fine-tuning of large pre-trained models.

Although PEFT reduces computational costs and improves results, most existing methods rely on prior knowledge to decide which model components to fine-tune. These methods typically introduce specialized learnable modules at predefined positions in the model. For example, prompt

[*]Work conducted during an internship at Tsinghua University.
[†]Corresponding author.

tuning [41, 59, 60] incorporates learnable prompt vectors at the input layer or specific transformer layers of CLIP, guiding the model's representation learning. In contrast, adapter tuning [15, 32, 37, 46] inserts additional lightweight modules within the transformer layers or at the final layer of CLIP to adapt to new tasks. However, these methods rely on the manual selection of model locations for fine-tuning, which can be suboptimal and often results in extra parameter overhead. Automatically selecting and fine-tuning the most critical parameters remains underexplored. Accurately selecting critical parameters can effectively address these limitations, thereby enhancing model performance without extra parameter costs.

To tackle this challenge, as illustrated in Fig. 1, we propose CLIP with Adaptive Selective Tuning (CLIP-AST), which aims to automatically select and fine-tune the most important parameters in CLIP without inserting extra learnable modules. CLIP-AST leverages the adaptive learning rate feature of the AdamW optimizer [28], where the effective learning rate of a parameter is inversely proportional to the second-moment estimate of its gradients. Parameters with lower second-moment estimates receive higher effective learning rates, allowing them to adapt quickly to new tasks. Conversely, parameters with higher second-moment estimates receive smaller updates, preventing large adjustments due to noisy gradients. By utilizing this property, we use second-moment estimates as importance scores to select and fine-tune the most critical parameters for the task.

Building on the dynamic learning rate property in the optimizer, CLIP-AST introduces adaptive selective fine-tuning. For each transformer layer, we rank the parameters based on importance scores and fine-tune the top K most important ones while freezing the others. This adaptive selective fine-tuning preserves the knowledge of the pre-trained model. It enhances task-specific adaptation without extra parameters during training and inference.

The main contributions are as follows:

- We propose CLIP-AST, a novel method that automatically selects and fine-tunes the most critical parameters in CLIP without extra parameters, thus maintaining the same inference efficiency as the original CLIP model.
- We introduce an automatic parameter identification strategy based on AdamW's adaptive learning rates to select important parameters with lower second-moment estimates of the gradient.
- Extensive experiments on 13 benchmark datasets demonstrate that CLIP-AST consistently outperforms the original CLIP and its variants, achieving state-of-the-art performance.

2. Related Work

Vision-Language Models. Vision-Language Models (VLMs) [18, 29, 38, 44, 54] aim to efficiently handle cross-modal tasks by jointly learning visual and linguistic information. Some methods, such as CLIP [38] and ALIGN [18], utilize hundreds of millions or even billions of image-text pairs for training to bridge the gap between visual and textual data. The vast amount of training data enables VLMs to exhibit exceptional capabilities. Among these models, the most representative is the CLIP model, which employs contrastive learning to map images and texts into the same feature space, thereby achieving efficient zero-shot learning. Due to its powerful cross-modal and zero-shot capabilities, CLIP has been widely used in various downstream tasks [10, 25, 30, 36, 43, 48, 51, 56, 58]. However, a gap exists between most downstream tasks and the original CLIP model, necessitating fine-tuning for effective transfer. Therefore, this work investigates the fine-tuning of CLIP, allowing CLIP to be effectively adapted to specific domains while maintaining its strong performance.

Parameter-Efficient Fine-Tuning for VLMs. Recently, with the rapid development of parameter-efficient fine-tuning (PEFT) [15, 16, 27] in the field of NLP, it has become a common paradigm for fine-tuning VLMs. Common PEFT methods include adapters [4, 11, 15, 22, 32, 37, 46], which introduce additional lightweight trainable module, as well as prompt tuning [19, 59, 60], which optimizes input prompts. In the context of applying PEFT to VLMs, CoOp [60] modifies the static text encoder prompts in the original CLIP model into adaptable, trainable vectors. Tip-Adapter [55] develops a key-value caching mechanism to facilitate efficient knowledge retrieval. PromptSRC [21] implements a self-regulation technique to mitigate overfitting. MMA [52] introduces a multimodal adapter to enhance the alignment between text and visual representations. Unlike previous methods, which choose the fine-tuning locations based on prior knowledge, we automatically find the important parameters in the original model that need to be trained and complete the fine-tuning of the model.

Selective Fine-tuning. Unlike approaches that retrain all model weights or add new trainable modules, selective fine-tuning [12, 40, 53, 57] focuses on training specific parameters within the model to achieve fine-tuning. A notable example is linear probing [3, 13, 38], where only the model's last layer is trained. However, restricting training to the last layer often yields suboptimal results. BitFit [53] introduces bias-only fine-tuning, which adjusts only the bias parameters (or a subset of them), keeping the rest of the model parameters fixed. These methods, however, rely on empirical insights or observations to select parameters for training, like the previous PEFT methods for VLMs. In contrast, BSR [40] introduces a systematic selective fine-tuning approach that identifies critical layers by comparing outputs from pre-trained and fine-tuned models. Layers showing significant output changes are prioritized for

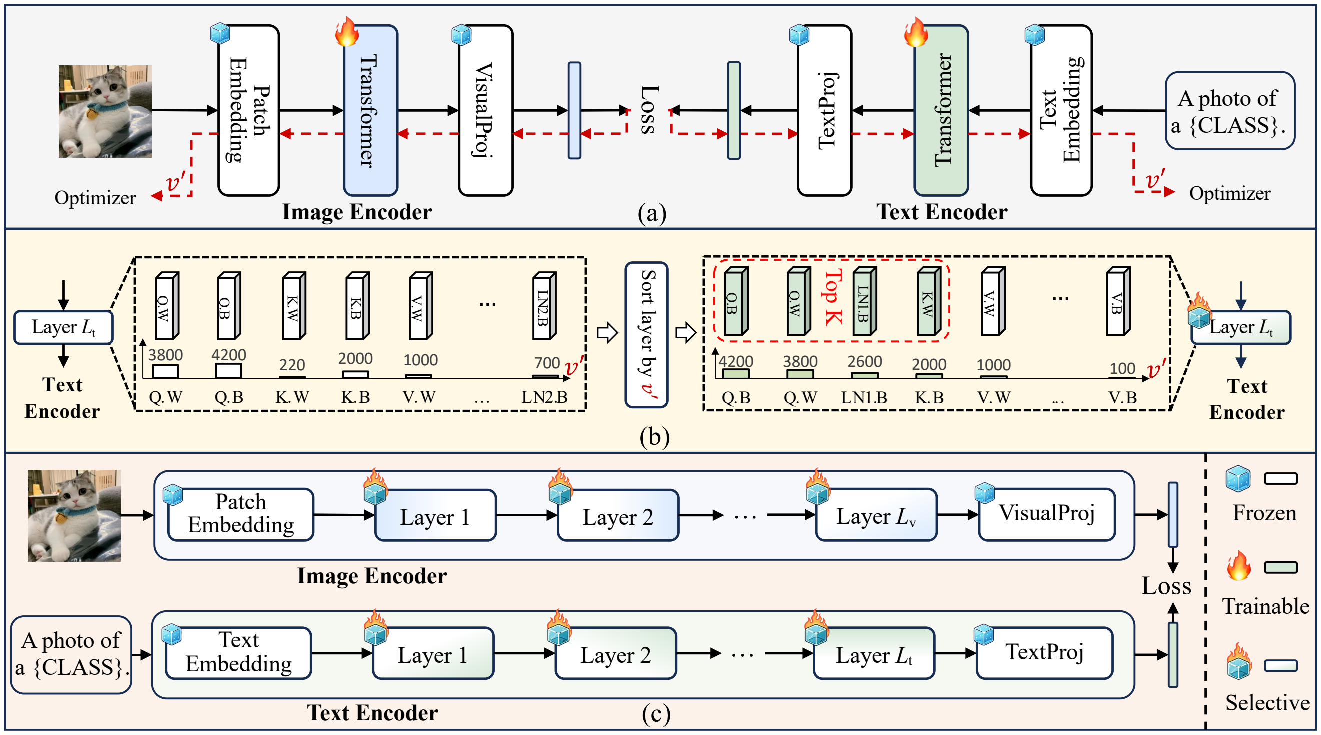

Figure 2. Overview of the proposed CLIP-AST pipeline, consisting of (a) transformer fine-tuning and (b) (c) adaptive selective fine-tuning stages. In (a), we fine-tune the transformer layers of both visual and text encoders to extract parameter importance scores. (b) illustrates the process of ranking sub-layer parameters using importance scores, followed by the adaptive selection of the top K sub-layers. (c) shows the selective fine-tuning process where only the most important sub-layers are updated while other components remain frozen, enhancing efficiency without compromising model performance.

fine-tuning based on task adaptability. SPT [12] takes a data-driven approach by computing parameter sensitivity on downstream tasks. It identifies the most impactful parameters and employs structured fine-tuning methods (e.g., LoRA or Adapter) rather than direct adjustments. Despite these advances, adaptive selective fine-tuning for CLIP remains relatively unexplored. This work addresses this gap by introducing an automated method for identifying the modules in CLIP that benefit most from fine-tuning, advancing the efficiency and effectiveness of fine-tuning for vision-language models.

3. Method

As illustrated in Fig. 2, CLIP-AST includes two stages: transformer fine-tuning and adaptive selective fine-tuning. Initially, we fine-tune the pre-trained layers of the transformer to extract importance scores from the optimizer. The optimizer measures the importance of parameters by estimating the second-order moments of the gradients in each transformer layer, as described in Sec. 3.1. After a few iterations, these scores are ranked to select the parameters requiring training, as specified in Sec. 3.2. Then, the important parameters in each layer of the transformer are fine-tuned to complete the fine-tuning of CLIP.

3.1. Transformer Fine-Tuning

To assess the importance of all the parameters of the transformer layer in the pre-trained CLIP model, as shown in Fig. 2 (a), we first fine-tune the CLIP transformer block. The CLIP model includes a visual encoder $\mathrm{E_v}$ and a textual encoder $\mathrm{E_t}$, both responsible for mapping images and text into a unified feature space. This design facilitates the interaction of features across different modalities. During the fine-tuning of CLIP's transformer, we first sampled a batch of training data from the dataset, which included N images $\mathbf{X} \in \mathbb{R}^{N\times 3\times H\times W}$, corresponding labels $\mathbf{Y} \in \mathbb{R}^{N}$, and the names of C classes $\mathbf{T} = \{\mathbf{T}_k\}, k \in \{1,\dots,C\}$ in the dataset. Subsequently, the visual encoder $\mathrm{E_v}$ extracts visual embeddings $\mathbf{G} \in \mathbb{R}^{N\times D}$ from the $\mathbf{X}$:

$$\begin{aligned} [\mathbf{G_0}, \mathbf{I_0}] &= \mathrm{PatchEmbedding}(\mathbf{X}), \\ [\mathbf{G_L}, \mathbf{I_L}] &= \mathrm{Transformer}([\mathbf{G_0}, \mathbf{I_0}]), \\ [\mathbf{G}, \mathbf{I}] &= \mathrm{VisualProj}([\mathbf{G_L}, \mathbf{I_L}]), \end{aligned} \tag{1}$$

where $\mathbf{G_0}$ and $\mathbf{I_0}$ represent the initial global and visual embeddings generated by the $\mathrm{PatchEmbedding}$ block from the input visual $\mathbf{X}$. The $\mathrm{Transformer}$ block processes these embeddings through multiple layers to refine the features, resulting in $\mathbf{G_L}$ and $\mathbf{I_L}$, which are the final global and visual embeddings at the last layer. Finally, the $\mathrm{VisualProj}$ block projects these embeddings onto a feature space, producing $\mathbf{G}$ and $\mathbf{I}$ as the final visual embeddings.

For the text encoder $\mathrm{E_t}$, the class names T are combined with a predefined template (e.g., ”a photo of [CLASS]”) to generate class descriptions $\mathbf{T_P}$. Then, each class description is sent to the text encoder to obtain the corresponding class embeddings $\mathbf{W} \in \mathbb{R}^{C\times D}$:

$$\begin{aligned}\mathbf{W_0} &= \mathrm{TextEmbedding}(\mathbf{T_P}),\\ \mathbf{W_L} &= \mathrm{Transformer}(\mathbf{W_0}),\\ \mathbf{W} &= \mathrm{TextProj}(\mathbf{W_L}).\end{aligned} \tag{2}$$

Here, $\mathbf{W_0}$ represents the initial text embeddings generated by the $\mathrm{TextEmbedding}$ layer. In contrast, $\mathbf{W_L}$ is the feature produced by passing $\mathbf{W_0}$ through multiple layers of the transformer, extracting semantic information further at each layer. Finally, $\mathrm{TextProj}$ projects the text features into a unified feature space, producing $\mathbf{W}$ as the final text embedding that aligns with the visual features.

After obtaining the $\mathbf{G}$ and $\mathbf{W}$, we first calculate the probability $\hat{y}_{i,j}$ of each image being predicted for each class and then calculate the loss function with the label:

$$\begin{aligned}\hat{y}_{i,j} &= \frac{\exp(\mathrm{sim}(\mathbf{G}_i, \mathbf{W}_j)/\tau)}{\sum_{k=1}^{c}\exp(\mathrm{sim}(\mathbf{G}_i, \mathbf{W}_k)/\tau)},\\ \mathcal{L}_{\text{CE}} &= -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{C} y_{i,j}\log(\hat{y}_{i,j}),\end{aligned} \tag{3}$$

where $\mathrm{sim}(\cdot,\cdot)$ represents the cosine similarity function, τ is a temperature parameter.

Fine-tuning the transformer requires only a few training iterations. As its name implies, only the transformer part of the two encoders is trainable during the training process. After training, we can obtain parameter importance scores v, which are embedded within the AdamW optimizer.

3.2. Adaptive Selective Fine-tuning

After transformer fine-tuning, we introduce the adaptive selective fine-tuning strategy, which uses the square root of the second-moment estimate of the gradient in the AdamW optimizer [28] as importance scores. The parameter update process in the i-th iteration of AdamW optimization can be described as follows. First, we compute the gradient g_i with respect to the model parameters θ:

$$g_i = \nabla_\theta f(\theta_{i-1}). \tag{4}$$

Next, we update the first-moment estimate m_i and the second-moment estimate v_i:

$$\begin{aligned}m_i &= \beta_1 \cdot m_{i-1} + (1-\beta_1)\cdot g_i,\\ v_i &= \beta_2 \cdot v_{i-1} + (1-\beta_2)\cdot g_i^2,\end{aligned} \tag{5}$$

where β_1 and β_2 are the hyperparameters controlling the exponential decay rates for the moment estimates. To correct the bias introduced in the early stages of optimization, we compute the bias-corrected estimates:

$$\hat{m}_i = \frac{m_i}{1-\beta_1^i}, \quad \hat{v}_i = \frac{v_i}{1-\beta_2^i}. \tag{6}$$

The final parameter update rule is then expressed as:

$$\theta_i = (\theta_{i-1} - \alpha\cdot\lambda\cdot\theta_{i-1}) - \frac{\alpha}{\sqrt{\hat{v}_i}+\epsilon}\cdot\hat{m}_i, \tag{7}$$

where α denotes the learning rate, λ represents the weight decay coefficient that controls the strength of regularization applied to the model parameters, and ϵ is a small constant added for numerical stability to prevent division by zero.

In our adaptive selective fine-tuning, the importance of each parameter is assessed using the square root of the second-moment estimate of gradient $\hat{v}_i$, which has the same dimensionality as the parameter itself. Specifically, we compute the scalar average importance score v' of the transformer layer sub-layer parameter i as follows:

$$v'_i = \mathrm{Avg}\left(\frac{1}{\sqrt{\hat{v}_i}}\right). \tag{8}$$

This score measures the magnitude of parameter updates during training. A higher v'_i means the parameter receives larger updates, suggesting it is more actively involved in learning and adapting to the task-specific data. Such parameters will likely capture essential features and contribute significantly to model predictions. Conversely, a lower v'_i implies that the parameter receives smaller updates, possibly because it requires less adjustment or is less sensitive to the current data.

By focusing on parameters with higher importance scores, we can adaptively and selectively fine-tune the model, prioritizing the most influential for the specific task. This selective fine-tuning strategy maintains model performance while reducing computational costs, as it concentrates updates on the most critical components without compromising the overall model integrity.

During the adaptive selective fine-tuning, shown in Fig. 2 (b) and (c), we select the most important components at the sub-layer level (e.g., weights or biases of the $\mathrm{linear_q}$ layer). For each layer in the transformer module, we select the top K important parameters of the sub-layer for training while freezing all others. This adaptive selective fine-tuning strategy enhances the efficiency and effectiveness of the model, allowing for focused updates that improve task-specific adaptation without compromising overall model integrity.

3.3. Loss Function

We utilize the self-consistency loss (SCL) [21] to mitigate overfitting during selective fine-tuning. SCL employs an additional frozen CLIP model to extract original visual embeddings $\tilde{\mathbf{G}}$ and textual embeddings $\tilde{\mathbf{W}}$ from the training data. By using an L1 loss, SCL enforces consistency between the features produced by the frozen model and the features produced by the fine-tuned model:

$$\begin{aligned}\mathcal{L}_{\text{SCL-image}} &= \sum_{i=1}^{d} |\tilde{\mathbf{G}} - \mathbf{G}|, \\ \mathcal{L}_{\text{SCL-text}} &= \sum_{i=1}^{d} |\tilde{\mathbf{W}} - \mathbf{W}|.\end{aligned} \tag{9}$$

In addition, SCL introduces logit-level consistency regularization, further strengthening the constraints and maximizing the alignment of features before and after training. This is achieved by minimizing the Kullback-Leibler divergence to align the trained logit distribution with the logits generated by the frozen CLIP model, as expressed in the following formula:

$$\mathcal{L}_{\text{SCL-logits}} = \mathcal{D}_{\mathcal{KL}}(\text{sim}(\tilde{\mathbf{G}}, \tilde{\mathbf{W}}), \text{sim}(\mathbf{G}, \mathbf{W})). \tag{10}$$

Finally, we incorporate the cross-entropy loss discussed in Sec. 3.1. The total loss our model aims to minimize is a combination of these types of losses, represented by the equation:

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda_1 \cdot \mathcal{L}_{\text{SCL-image}} + \lambda_2 \cdot \mathcal{L}_{\text{SCL-text}} + \lambda_3 \cdot \mathcal{L}_{\text{SCL-logits}}, \tag{11}$$

where λ_1, λ_2 and λ_3 are weights that balance the contributions of $\mathcal{L}_{\text{SCL-image}}$, $\mathcal{L}_{\text{SCL-text}}$ and $\mathcal{L}_{\text{SCL-logits}}$, respectively.

4. Experiments

4.1. Experimental Settings

Few-shot learning. We aim to evaluate the model's ability to generalize with limited labeled examples in the few-shot learning setting. We simulate a low-shot scenario by providing only a few labeled samples per class, typically 1, 2, 4, 8, 16. In this setting, we use the 11 commonly used classification datasets introduced in Sec. 4.2 to reflect the few-shot tasks in the real world. The performance under this setting is measured by top-1 accuracy.

Base-to-novel class generalization. The base-to-novel class generalization setting evaluates the model's adaptability to novel classes it has not seen during fine-tuning. In this setting, the model is trained on a base set of classes with 16-shot per class and is then tested on novel classes, assessing its capability to transfer knowledge from the base classes to novel, unseen classes. Following the previous work [21, 60], we partition the dataset into base and novel classes, ensuring no overlap between the two. The metrics include the top-1 accuracy of the base class (Base) and the novel class (Novel), as well as the harmonic mean (HM) of the two.

Out-of-distribution setting. In the out-of-distribution setting, we test the model's robustness against data that differs from the training distribution. This setting aims to assess the model's ability to handle unexpected inputs. We train on ImageNet as an in-distribution dataset and test on out-of-distribution datasets. The performance is evaluated for average accuracy on both in-distribution and out-of-distribution samples.

4.2. Datasets

For few-shot learning and base-to-novel class generalization, we conduct experiments on 11 commonly used classification benchmarks, including Caltech101 [9], DTD [5], EuroSAT [14], FGVC Aircraft [31], Flowers102 [33], Food101 [1], ImageNet [6], OxfordPets [34], StanfordCars [23], SUN397 [50], UCF101 [42]. For the out-of-distribution setting, We use ImageNet-Sketch [49] and ImageNetV2 [39] as out-of-distribution datasets.

4.3. Implementation details

For the adaptive selective fine-tuning stage, the preprocessing steps during training include randomly cropping the input images, with the crop size ranging from 0.5 to 1.0 times the original image size. The cropped images are then resized to a fixed size of 224 × 224 pixels. Next, each image undergoes horizontal flipping with a probability of 0.5. Finally, each channel of the images is normalized. We use the AdamW optimizer for training. For the few-shot learning and out-of-distribution settings, we train for 30 epochs, and for the base-to-novel class generalization setting, we train for 20 epochs. To ensure a fair comparison, we follow previous methods [21, 52] and use the ViT-B/16 CLIP model. For the transformer fine-tuning stage, we train for 1 epoch, with other settings being the same as adaptive selective fine-tuning. All experiments are implemented using the Jittor [17] and PyTorch [35] frameworks.

4.4. Comparison with State-of-the-art

Few-shot learning. In this setting, we benchmark CLIP-AST against state-of-the-art (SOTA) few-shot learning methods, including CoOp [60], Tip-Adapter [55], PromptSRC [21], and GraphAdapter [26]. As shown in Tab. 1, CLIP-AST consistently achieves the highest average accuracy across the ImageNet dataset as the shot count increases, demonstrating that our proposed method is on par with, or even surpasses, previously introduced methods such as prompt tuning and adapters. To further clarify the comparison, we present the overall and detailed results for 11 datasets at varying shot levels in Fig. 3. Notably, CLIP-

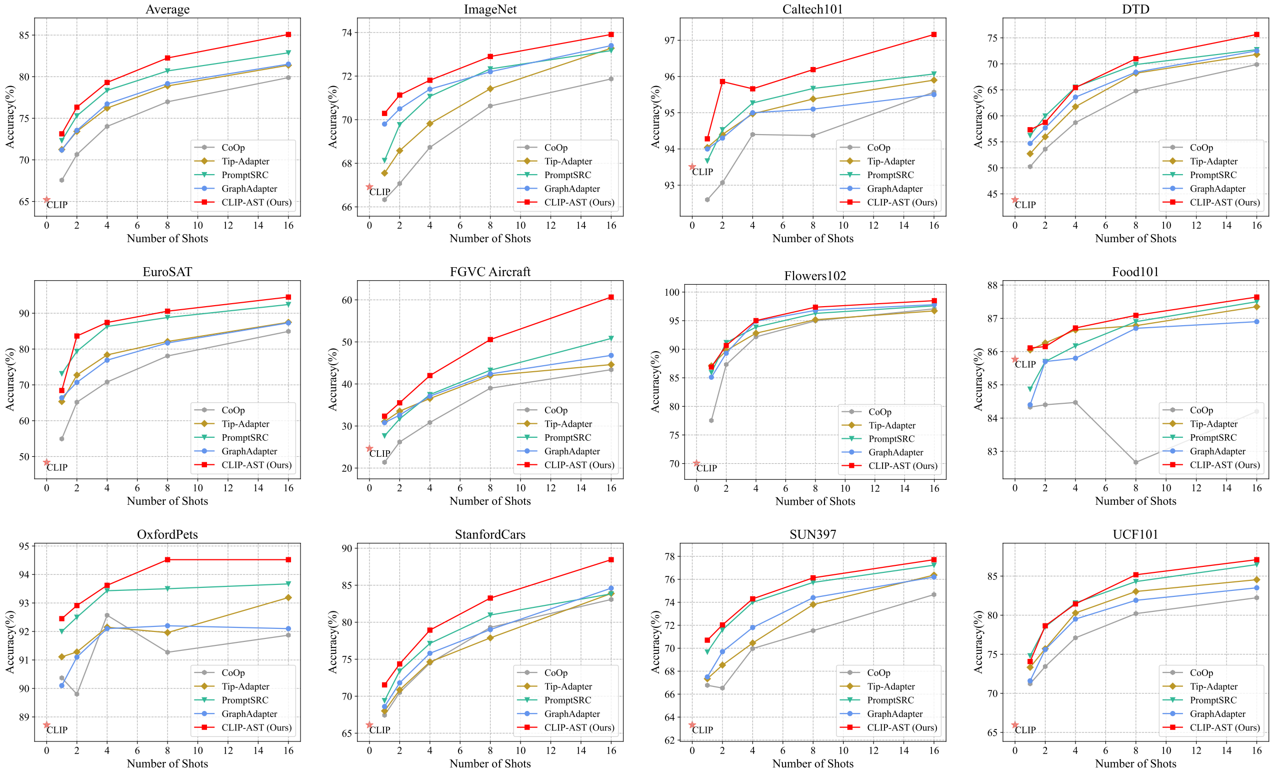

Figure 3. Comparison with previous SOTA across 11 datasets under 1, 2, 4, 8, and 16-shot settings demonstrates that our approach consistently outperforms existing methods, setting a new SOTA.

AST achieves an average accuracy improvement of 2.21% over the SOTA across 11 datasets. Across all datasets, our method outperforms previous SOTA approaches, with particularly substantial improvements on certain datasets. For instance, on FGVC Aircraft, CLIP-AST achieves approximately 10% higher accuracy with 16-shot, surpassing other methods such as PromptSRC and CoOp, which are specifically designed for prompt optimization. Moreover, as the sample count increases, the performance advantage of our model over prior methods also becomes more obvious. The consistent progress achieved by CLIP-AST on different levels under different samples demonstrates that our proposed fine-tuning method is a simple yet highly effective strategy in the few-shot learning setting.

Base-to-novel class generalization. In this setting, we evaluate the ability of CLIP-AST to generalize from base to novel classes by comparing it with state-of-the-art methods in Tab. 2, where CLIP-AST consistently performs favorably in both base and novel class recognition tasks. CLIP-AST achieves the highest harmonic mean (HM) across multiple datasets, indicating robust performance in both base and novel classes. For example, on the EuroSAT dataset, our method achieves an HM improvement of approximately 4.37% compared to MMA. This demonstrates the capability of CLIP-AST to maintain high accuracy across varying class distributions without overfitting to base classes. Our results illustrate that selective fine-tuning in a base-to-new setting allows CLIP-AST to balance learning across class distributions effectively. By avoiding overfitting to base classes, our approach maintains strong generalization to novel classes. The gains across both base and novel categories prove that CLIP-AST can manage the trade-off between base and novel class performance.

Method	Number of Shot				
	1	2	4	8	16
CLIP [38]	Zero-shot 66.92				
CoOp [60]	66.33	67.07	68.73	70.63	71.87
CoCoOp [59]	69.43	69.78	70.39	70.63	70.83
Tip-Adapter [55]	67.55	68.58	69.82	71.42	73.28
PromptSRC [21]	68.13	69.77	71.07	72.33	73.17
GraphAdapter [26]	69.80	70.50	71.40	71.40	73.40
CLIP-AST (Ours)	**70.29**	**71.13**	**71.81**	**72.90**	**73.91**

Table 1. Compared with the previous SOTA methods in the ImageNet dataset with the few-shot learning setting.

Out-of-distribution setting. In this setting, we assess the out-of-distribution robustness of CLIP-AST by comparing it with leading methods in Tab. 3. CLIP-AST achieves su-

Dataset		CLIP [38]	CoCoOp [59]	MaPLe [20]	PromptSRC [21]	MMA [52]	CLIP-AST (Ours)
Average	Base	69.34	80.47	82.28	84.26	83.20	**85.94**
	Novel	74.22	71.69	75.14	76.10	76.80	**76.99**
	HM	71.70	75.83	78.55	79.97	79.87	**81.06**
Caltech101	Base	96.84	97.96	97.74	98.10	98.40	**98.71**
	Novel	94.00	93.81	**94.36**	94.03	94.00	94.00
	HM	95.40	95.84	96.02	96.02	96.15	**96.30**
DTD	Base	53.24	77.01	80.36	83.37	83.20	**84.03**
	Novel	59.90	56.00	59.18	62.97	**65.63**	65.34
	HM	56.37	64.85	68.16	71.75	73.38	**73.52**
EuroSAT	Base	56.48	87.49	94.07	92.90	85.46	**95.90**
	Novel	64.05	60.04	73.23	73.90	**82.34**	81.72
	HM	60.03	71.21	82.35	82.32	83.87	**88.24**
FGVC Aircraft	Base	27.19	33.41	37.44	42.73	40.57	**48.98**
	Novel	36.29	23.71	35.61	37.87	36.33	**38.21**
	HM	31.09	27.74	36.50	40.15	38.33	**42.93**
Flowers102	Base	72.08	94.87	95.92	98.07	97.77	**97.91**
	Novel	77.80	71.75	72.46	76.50	75.93	**77.73**
	HM	74.83	81.71	82.56	85.95	85.48	**86.66**
Food101	Base	90.10	90.70	90.71	**90.67**	90.13	90.57
	Novel	91.22	91.29	92.05	**91.53**	91.30	91.11
	HM	90.66	90.99	91.38	**91.10**	90.71	90.84
ImageNet	Base	72.43	75.98	76.66	77.60	77.31	**78.44**
	Novel	68.14	70.43	70.54	70.73	**71.00**	70.22
	HM	70.22	73.10	73.47	74.01	74.02	**74.10**
OxfordPets	Base	91.17	95.20	95.43	95.33	95.40	**96.23**
	Novel	97.26	97.69	97.76	97.30	**98.07**	97.37
	HM	94.12	96.43	96.58	96.30	96.72	**96.80**
Stanford Cars	Base	63.37	70.49	72.94	78.27	78.50	**84.21**
	Novel	74.89	73.59	74.00	**74.97**	73.10	74.05
	HM	68.65	72.01	73.47	76.58	75.70	**78.80**
SUN397	Base	69.36	79.74	80.82	82.67	82.27	**83.05**
	Novel	75.35	76.86	78.70	78.47	**78.57**	78.12
	HM	72.23	78.27	79.75	80.52	80.38	**80.51**
UCF101	Base	70.53	82.33	83.00	87.10	86.23	**87.38**
	Novel	77.50	73.45	78.66	78.80	**80.03**	79.12
	HM	73.85	77.64	80.77	82.74	82.20	**83.05**

Table 2. Compared with the previous SOTA methods in the base-to-novel class generalization setting, where the model is trained on the base class with 16-shot and evaluated on the base class and novel class.

Method	Source	Target		Avg Acc(%)
	ImageNet	-Sketch	-V2	
CLIP [38]	66.73	46.15	60.83	57.90
CoOp [60]	71.51	47.99	64.20	61.23
CoCoOp [59]	71.02	48.75	64.07	61.28
Tip-Adapter [55]	73.23	46.82	65.01	61.68
MaPLe [20]	70.72	49.15	64.07	61.31
PromptSRC [21]	71.27	**49.55**	64.35	61.72
MMA [52]	71.00	49.13	64.33	61.48
CLIP-AST (Ours)	**73.87**	48.37	**66.37**	**62.87**

Table 3. Compared with the previous SOTA methods in the out-of-distribution setting, where the model is trained on the ImageNet dataset with 16-shot and evaluated on the ImageNet-V2 and ImageNet-Sketch benchmarks.

Method	Tuning Type	Train time(s)	Inference FPS	Acc(%)
CLIP [38]	-	**0**	**1323**	66.12
CoCoOp [59]	Prompt	2280	54	72.00
Tip-Adapter [55]	Adapter	30	1280	83.09
PromptSRC [21]	Prompt	630	1236	74.70
MMA [52]	Adapter	135	1113	83.60
CLIP-AST (Ours)	Adaptive Selection	100	**1323**	**85.64**

Table 4. Comparison of training time, inference efficiency, and accuracy across different methods. All methods are compared using a single NVIDIA 4090 GPU, trained for 10 epochs on the StanfordCars dataset.

perior performance on both in-distribution (ImageNet) and out-of-distribution target datasets (ImageNet-Sketch and ImageNet-V2). CLIP-AST achieves the highest average accuracy across these datasets, notably achieving a 2.04% improvement over the next-best method on ImageNet-V2. This robust performance on challenging out-of-distribution tasks highlights the strength of our method for maintaining model generalization beyond the source distribution. These results demonstrate that CLIP-AST can effectively mitigate the domain shift, balancing in-distribution and out-of-distribution accuracy.

4.5. Ablation Study

In our ablation study using the few-shot learning setting, we first analyzed our selected approach's training and inference efficiency compared to previous fine-tuning paradigms. Then, we conducted an ablation study across 11 datasets. We extensively evaluated different approaches to selective fine-tuning and then performed a detailed ablation test on the hyperparameters of the CLIP-AST.

Training and inference efficiency analysis. We compare the training and inference efficiency of various methods in Tab. 4. Our method achieves the highest accuracy of 85.64% while maintaining competitive training and inference efficiency. All methods were trained for 10 epochs on the StanfordCars dataset. Our method requires only 100 seconds of training time, less than needed for prompt tuning methods. The inference speed is also the same as the original CLIP model and faster than both prompt and adapter tuning methods. This indicates that our selective fine-tuning method improves performance and maintains high efficiency during the training and inference stages.

Adaptive selective fine-tuning schemes. Our ablation study, summarized in Tab. 5, evaluates the impact of different selective fine-tuning schemes on model performance. We start with a baseline where neither the image encoder nor the text encoder is fine-tuned, resulting in an average accuracy of 65.21%. Fine-tuning only the image encoder or only the text encoder using the adaptive selection strategy yields average accuracies of 84.04% and 81.95%, respectively, indicating significant contributions from both encoders. Next, We compared BitFit [53], which utilizes a priori fixed selection of bias as a training parameter, and found that the method performs poorly. We also compared the global granularity for adaptive selection, where the top K important types of sub-layers are trained across the entire transformer, with the layer granularity selection, where the top K important layers are trained within each layer. These results highlight the effectiveness of finer-grained layer granularity selection.

Hyperparameters in adaptive selection. In Fig. 4, we per-

Trainable Image Encoder	Trainable Text Encoder	Scheme for Selecting Trainable Layers	Avg Acc(%)
✗	✗	-	65.21
✓	✗	layer granularity	84.04
✗	✓	layer granularity	81.95
✓	✓	fixed bias	79.96
✓	✓	global granularity	84.25
✓	✓	layer granularity	**85.05**

Table 5. Ablation of selective fine-tuning schemes. Fixed bias trains the bias parameters, global granularity chooses the most important top K sub-layers across the model, and layer granularity picks the most important top K sub-layers within each transformer layer. Avg Acc is the average accuracy across 11 datasets in the few-shot learning setting with 16-shot.

formed an ablation on the hyperparameters of adaptive selection. First, we ablated the number of epochs for transformer fine-tuning used to obtain the importance scores and found that the number of epochs had little impact on the results. Next, we ablated the number of training layers K. We found that smaller or larger values of K did not lead to optimal results, possibly due to underfitting and overfitting. Ultimately, we chose 6 sub-layers as the hyperparameter for the few-shot learning setting.

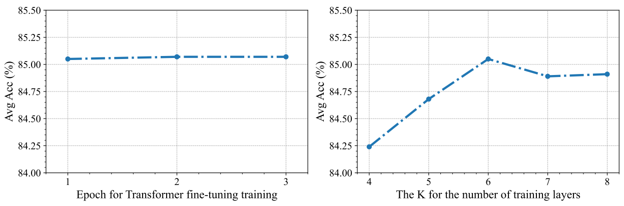

Figure 4. Ablation of epoch for transformer fine-tuning and the number of training layers in adaptive selective fine-tuning.

4.6. Visualization

Statistics of training parameter selection. In Fig. 5, we visualize the parameter selection results in the few-shot setting for 16-shot across 11 datasets. Firstly, we observe that the sub-layers chosen for both the image and text encoders are similar but not identical. Notably, some sub-layers are consistently selected across different layers, indicating that these sub-layers might be responsible for extracting general features. Additionally, the sub-layers selected by the transformer layer are also different, indicating that the functions responsible for shallow and deep layers are not the same.

Image feature analysis before and after training. In Fig. 6, we visualize the distribution of image features extracted by the model before and after training using T-SNE [38]. On the left, the features are scattered with significant overlap between different classes, indicating that the model has not yet effectively distinguished visual features. After training (right side), the features form compact and distinct clusters, demonstrating that the model has successfully learned to extract discriminative features.

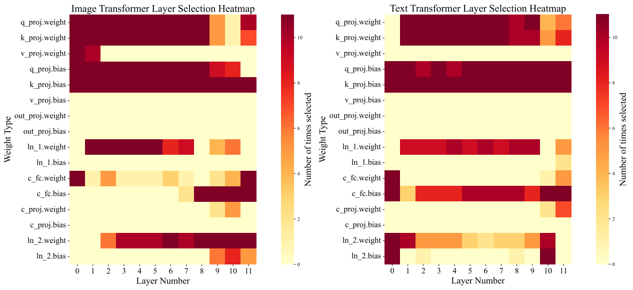

Figure 5. Heatmaps of adaptive parameter selection for image (left) and text (right) encoders across 11 datasets in the 16-shot setting.

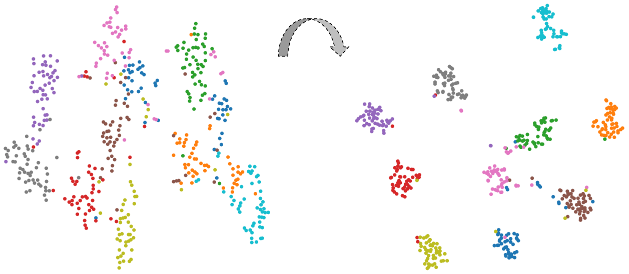

Figure 6. T-SNE visualization of image features before (left) and after (right) training. After training, features form clearer clusters, showing improved class separation.

5. Conclusion

We introduce CLIP-AST, an adaptive selective fine-tuning method for VLMs that addresses the limitations of existing PEFT approaches for CLIP. Unlike traditional methods that rely on the manual selection of adaptation positions, CLIP-AST uses the adaptive learning rate mechanism of the AdamW optimizer to automatically select critical parameters with lower second-moment estimates of their gradients. This enables selective fine-tuning of key sub-layers, preserving pre-trained knowledge, reducing overfitting, and minimizing computational costs. Extensive experiments show that CLIP-AST outperforms the original CLIP model and achieves exceptional performance across numerous benchmarks. Overall, it provides a robust framework for efficient fine-tuning of VLMs, with potential for application in other multimodal architectures and further enhancements in parameter selection strategies. Future work could explore extending this approach to other multimodal architectures and further refine the parameter selection criteria for even greater performance gains.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (project Nos. 62220106003, 623B2057), the Research Grant of Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 - mining discriminative components with random forests. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, pages 446–461. Springer, 2014. 5

[2] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 1

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 1597–1607. PMLR, 2020. 2

[4] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 1, 2

[5] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 3606–3613. IEEE Computer Society, 2014. 5

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. 5

[7] Sinuo Deng, Lifang Wu, Ge Shi, Lehao Xing, Meng Jian, Ye Xiang, and Ruihai Dong. Learning to compose diversified prompts for image emotion classification. *Comput. Vis. Media*, 10(6):1169–1183, 2024. 1

[8] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T. Toshev, and Vaishaal Shankar. Data filtering networks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 1

[9] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2004, Washington, DC, USA, June 27 - July 2, 2004*, page 178. IEEE Computer Society, 2004. 5

[10] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *Int. J. Comput. Vis.*, 132(2):581–595, 2024. 1, 2

[11] Meng-Hao Guo, Yi Zhang, Tai-Jiang Mu, Sharon X. Huang, and Shi-Min Hu. Tuning vision-language models with multiple prototypes clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):11186–11199, 2024. 2

[12] Haoyu He, Jianfei Cai, Jing Zhang, Dacheng Tao, and Bohan Zhuang. Sensitivity-aware visual parameter-efficient fine-tuning. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 11791–11801. IEEE, 2023. 2, 3

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. Computer Vision Foundation / IEEE, 2020. 2

[14] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 12(7):2217–2226, 2019. 5

[15] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 2790–2799. PMLR, 2019. 1, 2

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 2

[17] Shi-Min Hu, Dun Liang, Guo-Ye Yang, Guo-Wei Yang, and Wen-Yang Zhou. Jittor: a novel deep learning framework with meta-operators and unified graph execution. *Sci. China Inf. Sci.*, 63(12), 2020. 5

[18] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pages 4904–4916. PMLR, 2021. 2

[19] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIII*, pages 709–727. Springer, 2022. 2

[20] Muhammad Uzair Khattak, Hanoona Abdul Rasheed, Muhammad Maaz, Salman H. Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 19113–19122. IEEE, 2023. 7

[21] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model

adaptation without forgetting. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 15144–15154. IEEE, 2023. 2, 5, 6, 7, 1

[22] Sungyeon Kim, Boseung Jeong, Donghyun Kim, and Suha Kwak. Efficient and versatile robust fine-tuning of zero-shot models. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XVII*, pages 440–458. Springer, 2024. 2

[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, pages 554–561. IEEE Computer Society, 2013. 5

[24] Seung Hyun Lee, Sieun Kim, Wonmin Byeon, Gyeongrok Oh, Sumin In, Hyeongcheol Park, Sang Ho Yoon, Sung-Hee Hong, Jinkyu Kim, and Sangpil Kim. Audio-guided implicit neural representation for local image stylization. *Comput. Vis. Media*, 10(6):1185–1204, 2024. 1

[25] Jiaao Li, Yixiang Huang, Ming Wu, Bin Zhang, Xu Ji, and Chuang Zhang. CLIP-SP: vision-language model with adaptive prompting for scene parsing. *Comput. Vis. Media*, 10(4): 741–752, 2024. 2

[26] Xin Li, Dongze Lian, Zhihe Lu, Jiawang Bai, Zhibo Chen, and Xinchao Wang. Graphadapter: Tuning vision-language models with dual knowledge graph. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 5, 6

[27] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35, 2023. 1, 2

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 2, 4

[29] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23, 2019. 1, 2

[30] Hao Ma, Ming Li, Jingyuan Yang, Or Patashnik, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Clip-flow: Decoding images encoded in CLIP space. *Comput. Vis. Media*, 10(6):1157–1168, 2024. 2

[31] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151, 2013. 5

[32] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 4296–4304. AAAI Press, 2024. 2

[33] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Sixth Indian Conference on Computer Vision, Graphics &amp; Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008*, pages 722–729. IEEE Computer Society, 2008. 5

[34] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 3498–3505. IEEE Computer Society, 2012. 5

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas K&quot;opf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. 5

[36] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2065–2074. IEEE, 2021. 2

[37] Jonas Pfeiffer, Aishwarya Kamath, Andreas R&quot;uckl&apos;e, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 487–503. Association for Computational Linguistics, 2021. 2

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pages 8748–8763. PMLR, 2021. 1, 2, 6, 7, 8

[39] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5389–5400. PMLR, 2019. 5

[40] Sreetama Sarkar, Souvik Kundu, Kai Zheng, and Peter A. Beerel. Block selective reprogramming for on-device training of vision transformers. In *IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024, pages 8094–8103. IEEE, 2024. 2

[41] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 2

[42] K Soomro. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[43] Sanjay Subramanian, William Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach. Reclip: A strong zero-shot baseline for referring expression comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5198–5215. Association for Computational Linguistics, 2022. 1, 2

[44] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: improved training techniques for CLIP at scale. *CoRR*, abs/2303.15389, 2023. 1, 2

[45] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-clip: A CLIP model focusing on wherever you want. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13019–13029. IEEE, 2024. 1

[46] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. VL-ADAPTER: parameter-efficient transfer learning for vision-and-language tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5217–5227. IEEE, 2022. 2

[47] Vidit Vidit, Martin Engilberge, and Mathieu Salzmann. CLIP the gap: A single domain generalization approach for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 3219–3229. IEEE, 2023. 1

[48] Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: semantically-aware object sketching. *ACM Trans. Graph.*, 41(4):86:1–86:11, 2022. 2

[49] Haohan Wang, Songwei Ge, Zachary C. Lipton, and Eric P. Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 10506–10518, 2019. 5

[50] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3485–3492. IEEE Computer Society, 2010. 5

[51] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. SAN: side adapter network for open-vocabulary semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12):15546–15561, 2023. 1, 2

[52] Lingxiao Yang, Ru-Yuan Zhang, Yanchen Wang, and Xiaohua Xie. MMA: multi-modal adapter for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 23826–23837. IEEE, 2024. 2, 5, 7

[53] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9. Association for Computational Linguistics, 2022. 2, 7, 1

[54] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 11941–11952. IEEE, 2023. 1, 2

[55] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of CLIP for few-shot classification. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXV*, pages 493–510. Springer, 2022. 2, 5, 6, 7, 1

[56] Yi Zhang, Meng-Hao Guo, Miao Wang, and Shi-Min Hu. Exploring regional clues in CLIP for zero-shot semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 3270–3280. IEEE, 2024. 1, 2

[57] Zhi Zhang, Qizhe Zhang, Zijun Gao, Renrui Zhang, Ekaterina Shutova, Shiji Zhou, and Shanghang Zhang. Gradient-based parameter selection for efficient fine-tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 28566–28577. IEEE, 2024. 2, 1

[58] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVIII*, pages 696–712. Springer, 2022. 1, 2

[59] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16795–16804. IEEE, 2022. 2, 6, 7

[60] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *Int. J. Comput. Vis.*, 130(9):2337–2348, 2022. 2, 5, 6, 7

[61] Wenyang Zhou, Lu Yuan, and Taijiang Mu. Multi3d: 3d-aware multimodal image synthesis. *Comput. Vis. Media*, 10 (6):1205–1217, 2024. 1