Connecting the Dots: Collaborative Fine-tuning for Black-Box Vision-Language Models

Zhengbo Wang [1 2] **Jian Liang** [2 3] **Ran He** [2 3] **Zilei Wang** [1] **Tieniu Tan** [2 3 4]

Abstract

With the emergence of pretrained vision-language models (VLMs), considerable efforts have been devoted to fine-tuning them for downstream tasks. Despite the progress made in designing efficient fine-tuning methods, such methods require access to the model's parameters, which can be challenging as model owners often opt to provide their models as a black box to safeguard model ownership. This paper proposes a **C**ollabo**ra**tive **F**ine-**T**uning (**CraFT**) approach for fine-tuning black-box VLMs to downstream tasks, where one only has access to the input prompts and the output predictions of the model. CraFT comprises two modules, a prompt generation module for learning text prompts and a prediction refinement module for enhancing output predictions in residual style. Additionally, we introduce an auxiliary prediction-consistent loss to promote consistent optimization across these modules. These modules are optimized by a novel collaborative training algorithm. Extensive experiments on few-shot classification over 15 datasets demonstrate the superiority of CraFT. The results show that CraFT achieves a decent gain of about 12% with 16-shot datasets and only 8,000 queries. Moreover, CraFT trains faster and uses only about 1/80 of the memory footprint for deployment, while sacrificing only 1.62% compared to the white-box method. Our code is publicly available at https://github.com/mrflogs/CraFT.

[1]University of Science and Technology of China [2]NLPR & MAIS, Institute of Automation, Chinese Academy of Sciences [3]School of Artificial Intelligence, University of Chinese Academy of Sciences [4]Nanjing University. Correspondence to: Jian Liang <liangjian92@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

In recent years, large-scale pretrained vision-language models have garnered much attention. By establishing a link between images and natural language, these models exhibit impressive zero-shot capabilities and remarkable transfer ability (Radford et al., 2021; Jia et al., 2021; Alayrac et al., 2022; Li et al., 2022), demonstrating potential in learning open-world concepts. One of the most successful large-scale pretrained vision-language models is CLIP (Radford et al., 2021). After pretraining, CLIP (Radford et al., 2021) can perform zero-shot recognition by merely providing the class names. The classification weights are generated by the language encoder through prompting (Liu et al., 2023).

Besides its remarkable zero-shot ability, recent studies have found that CLIP (Radford et al., 2021) also possesses astonishing transfer ability (Zhou et al., 2022b; Zhang et al., 2022; Lu et al., 2022). For example, CoOp (Zhou et al., 2022b) can achieve a 15% improvement compared to zero-shot CLIP (Radford et al., 2021) with only 16 samples per class by fine-tuning a mere 16k parameters. However, these methods assume we have access to the model parameters, which is unrealistic in the current era. Training large vision-language models typically requires extensive computational resources and data, thus leading to high training costs. Therefore, model owners seldom release the model and the weights to protect the model ownership. Typically, model owners deploy the models as a service, such as GPT-4 (OpenAI, 2023), where we can only obtain the input and output. Thus, it is crucial to explore ways to fine-tune powerful vision-language models in the black-box scenario.

To address the aforementioned challenge, we propose **C**ollabo**ra**tive **F**ine-**T**uning (**CraFT**), a parameter- and data-efficient fine-tuning approach for black-box VLMs. The CraFT framework comprises three key components. Firstly, it incorporates a prompt generation module designed to learn global text prompts tailored to downstream datasets. Given the unavailability of gradients from the black-box model, we leverage derivative-free optimization (DFO) for the module, inspired by prior works (Sun et al., 2022b;a). The DFO method assumes that the module's parameters adhere to a parameterized distribution. By sampling solutions from this distribution and calculating the corresponding loss values,

it iteratively updates the distribution parameters. Later, we can obtain the module's parameter from the distribution. To accelerate the optimization process, the text prompts are projected into a lower-dimensional subspace using a random matrix, as Aghajanyan et al. (2021) demonstrates a low-dimensional subspace can be as effective as the full parameter space for fine-tuning.

Secondly, CraFT introduces a prediction refinement module aimed at enhancing the VLM's output predictions. This module builds upon the predictions of black-box models which thus can be optimized through gradient descent. It consists of a three-layer MLP that learns the prediction's residual, where the residual connection plays a pivotal role in the collaborative training algorithm discussed below.

Thirdly, CraFT develops a novel collaborative training algorithm to optimize the aforementioned modules jointly. Given that the prompt generation module and the prediction refinement module are optimized using different optimizers (derivative-free and derivative-based), their joint training poses a challenge. To address this, we demonstrate that the model with residual connections can be reframed as the addition of outputs of each layer, enabling the modules to be optimized alternately. Fortunately, both VLMs and the prediction refinement module incorporate shortcut connections, facilitating this iterative optimization. To improve training stability, we introduce a prediction-consistent loss that penalizes deviations between the black-box model's output and the refinement module's output.

Our main contributions are summarized as follows:

- This paper is among the pioneering works in exploring efficient fine-tuning methods for black-box vision-language models, providing a new framework for fine-tuning black-box VLMs.
- CraFT comprises a prompt generation module and a prediction refinement module, which are designed to learn the text prompts and refine the output predictions, respectively. In addition, a collaborative training algorithm and a prediction-consistent loss are proposed to train these modules jointly and collaboratively.
- CraFT significantly outperforms black-box baselines on 15 datasets on few-shot classification. Compared to the white-box method, CraFT trains faster and requires only 1/80 of the memory footprint for deployment.

2. Related Work

Vision-Language Models. In recent years, vision-language models (VLMs) have gained popularity as fundamental models that aim to connect the modalities of vision and language. These models are pretrained on large-scale image-text datasets, which endows them with powerful transferable abilities such as zero-shot learning, few-shot adaptation, and in-context learning (Radford et al., 2021; Kim et al., 2021; Lu et al., 2019; Su et al., 2020; Jia et al., 2021; Wang et al., 2023a; Chen et al., 2023a; Wen et al., 2023). Contrastive-based methods have become the mainstream approach in this field. These methods, including CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), are trained on large-scale web-based noisy image-text pairs. They employ a language encoder and a vision encoder to encode the texts and images, respectively, and learn to align their representations through contrastive loss.

Efficient Fine-tuning for VLMs. Inspired by the prior works in NLP, recent researches focus on developing efficient fine-tuning methods for VLMs on downstream tasks (Zhou et al., 2022b;a; Zhang et al., 2022; Gao et al., 2024; Lu et al., 2022; Chen et al., 2023b; Derakhshani et al., 2023; Wang et al., 2023b; 2024). Existing efficient fine-tuning methods can be classified into two categories: prompt tuning (Zhou et al., 2022b;a; Lu et al., 2022; Chen et al., 2023b) and adapter-style tuning (Gao et al., 2024; Zhang et al., 2022). Prompt tuning methods propose to learn soft text prompts for downstream tasks through back-propagation on few-shot datasets. For instance, CoOp (Zhou et al., 2022b) proposes to learn soft text prompts through back-propagation on few-shot datasets. Adapter-style tuning methods, on the other hand, maintain the original zero-shot classifier but refine the output representation. CLIP-Adapter (Gao et al., 2024) proposed to add MLPs to refine the visual and text features via a residual connection. Although these methods have achieved satisfactory results on downstream datasets, they all assume that the entire parameters of VLMs are available. However, to safeguard the model ownership, it's difficult to obtain the parameters and architecture of the models. Therefore, it is necessary to investigate ways to fine-tune black-box VLMs.

Black-Box Optimization. Recently, influenced by developments in NLP, there has been a growing focus on addressing the challenge of fine-tuning VLMs in black-box scenarios (Oh et al., 2023; Sun et al., 2022a;b; Yu et al., 2023; Malladi et al., 2023; Guo et al., 2023; Ouali et al., 2023). The methods typically can be split into two categories: zeroth-order optimization and evolutionary algorithms. Zeroth-order methods (Oh et al., 2023; Guo et al., 2023) address the black-box optimization by estimating the gradients, while the evolutionary methods (Sun et al., 2022b; Yu et al., 2023) solve it by generating candidate solutions through a parameterized distribution. Though there are some works in black-box VLMs, the underlying assumptions of these methods may be deemed somewhat unreasonable. For example, LFA (Ouali et al., 2023) assumes that the VLM is dual-towered and we can obtain the corresponding features, which restricts its applicability. And Yu et al. (2023) assumes they can access the architecture of VLMs, but update

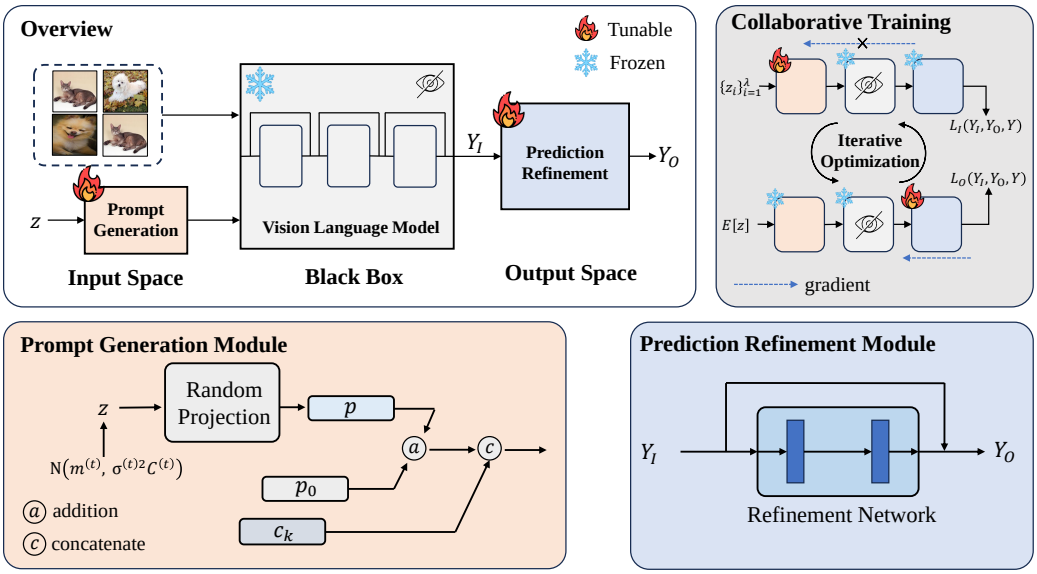

Figure 1: **The architecture of our proposed method.** Our proposed method consists of two modules: a prompt generation module and a prediction refinement module. The prompt generation module utilizes the CMA-ES optimizer to learn the text prompts. Specifically, given $z \in \mathbb{R}^{d_0}$ from the CMA-ES optimizer, we project it into the prompt space using a random matrix $A \in \mathbb{R}^{nd \times d_0}$ and add it to initial prompt embeddings $p_0 \in \mathbb{R}^{n \times d}$ (e.g., "a photo of a"). We then concatenate it with the class name embedding c_k to obtain the final prompts $p_k = [p_0 + Az, c_k]$, where $k = 1, 2, \ldots, K$ for K classes. The prediction refinement module refines the output of the black-box model using a refinement network in residual style. It can be optimized using gradient descent. Since the modules use different optimizers, we propose a novel algorithm to train them collaboratively. For more details on the training process, please refer to section 3.3.

it without gradients. In our setting, we assume the model is invisible, and no assumptions are made about its architecture. Detailed comparison of black-box setting can refer to appendix A.

3. Method

As depicted in Figure 1, we divide the model into three distinct parts: the input space, the black-box vision-language model, and the output space. Since we lack access to the parameters of the black-box model, we can solely integrate learnable modules in the input and output spaces. In the input space, we propose a prompt generation module, which learns global text prompts for downstream tasks using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen et al., 2003). In the output space, we propose a prediction refinement module to refine the output prediction of the black-box model in residual style. Later, we propose a collaborative training algorithm to train them jointly, despite they utilize different optimizers.

3.1. Prompt Generation Module

We consider a black-box vision-language model, denoted by f. Given the model, we can only obtain its input and output prediction $f(\{t_k\}_{k=1}^K, \{i_n\}_{n=1}^N) \in \mathbb{R}^{N \times K}$. Here, $\{i_n\}_{n=1}^N$ refers to the N images that are uploaded to the black-box vision-language model, and $\{t_k\}_{k=1}^K$ denotes the K class text prompts, where each prompt t_k consists of a predefined text embedding p_0 (e.g., "a photo of a") and a corresponding class name c_k. Specifically, we have $t_k = [p_0, c_k]$, $k = 1, 2, \ldots, K$ for K classes.

As shown in Figure 1, we propose to learn global prompts $p \in \mathbb{R}^{n \times d}$ for the black-box model, where n and d represent the length of the prompts and their dimension, respectively. Previous work (Aghajanyan et al., 2021) reveals that a low-dimensional subspace can be as effective as the full parameter space for fine-tuning, we further reduce the search space for fast training by mapping the prompts p into a low-dimensional subspace using a random matrix, i.e., $p = Az$. Here, $A \in \mathbb{R}^{nd \times d_0}$ is a random matrix sampled from a Gaussian distribution, and $d_0 \ll nd$ is the dimension of the subspace. Next, we add the prompts to the initial prompt embeddings p_0. Thus, the optimization can be formulated as follows:

$$\min_z \mathcal{L}(f(\{[p_0 + Az, c_k]\}_{k=1}^K, \{i_n\}_{n=1}^N), Y), \quad (1)$$

where $\mathcal{L}$ is the cross-entropy, and Y denotes the ground-truth. Since the model's gradients are not accessible, we solve this problem using a DFO method, CMA-ES (Hansen et al., 2003).

CMA-ES (Hansen et al., 2003) is a parameterized search distribution model that uses a multivariate normal distribution. At each iteration, CMA-ES generates a population of new query solutions by sampling from the multivariate normal distribution:

$$z_i \sim m^{(t)} + \sigma^{(t)}\mathcal{N}(0, C^{(t)}), \quad i = 1, 2, \ldots, \lambda. \quad (2)$$

Here, i denotes the index of the sampled solution, λ is the population size, $m^{(t)}$ represents the distribution mean, $\sigma^{(t)} \geq 0$ is the step-size, and $C^{(t)}$ denotes the covariance matrix of the distribution. The parameters $m^{(t)}, \sigma^{(t)}, C^{(t)}$ are updated in each iteration to minimize the loss of the sample solutions.

3.2. Prediction Refinement Module

Besides learning text prompts, we further build a refinement network on top of the output of the black-box vision-language model, which learns to refine the output prediction in residual style.[1] Specifically, given the original output $Y_I \in \mathbb{R}^K$ of the black-box model, the refinement network learns to generate the residual $R(Y_I)$, which is added to the original output to obtain the final result:

$$Y_O = Y_I + R(Y_I). \quad (3)$$

The refinement network is trained to minimize the cross-entropy. As the refinement network is built on top of the black-box model and does not require gradients from it, we utilize gradient descent to optimize the refinement network.

3.3. Collaborative Training Algorithm

As shown in Figure 1, the prompt generation module and prediction refinement module uses different optimizers (CMA-ES and AdamW, respectively). Therefore, optimizing the modules jointly becomes a challenge. To address this issue, we propose a collaborative training algorithm for them.

Previous works (Mei et al., 2016; Kandasamy et al., 2015; Sun et al., 2022a) have shown that networks with shortcut connections can be decomposed into some additive form. Therefore, different layers can be optimized separately. For example, considering the optimization of the first and third layers in a three-layer model with residual connections (similar to our scenario), this optimization problem can be decomposed as follows:

$$\begin{aligned}
\min_{\theta_1,\theta_3} f(x) &= \min_{\theta_1,\theta_3} f_3(x_3) + x_3 \\
&= \min_{\theta_1,\theta_3} f_3(x_3) + f_2(x_2) + x_2 \\
&= \min_{\theta_1,\theta_3} f_3(x_3) + f_2(x_2) + f_1(x_1) + x_1 \\
&= \min_{\theta_3} f_3(x_3) + f_2(x_2) + \min_{\theta_1} f_1(x_1) + x_1,
\end{aligned} \quad (4)$$

[1]The residual connection is necessary for collaborative training algorithm 3.3. Table 7 shows its ablation.

Algorithm 1 Collaborative Training for CLIP

Require: Budget of API calls $\mathcal{B}$, Population size λ, Dataset size $|\mathcal{D}|$, Batch size B, Refinement network R with residual connections.

Initialize random projections A
Initialize parameters $m^{(0)}, \sigma^{(0)}, C^{(0)}$
for $i = 1$ to $\mathcal{B}/\lambda$ **do**
  # Optimize prompt generation module
  Sample λ solutions $z_i \sim m^{(t)} + \sigma^{(t)}\mathcal{N}(0, C^{(t)})$
  Compute the fitnesses using Equation (5)
  Update $m^{(t)}, \sigma^{(t)}, C^{(t)}$ using the CMA-ES
  # Optimize prediction refinement module
  for $j = 1$ to $|\mathcal{D}|/B$ **do**
    Sample batch (Y_I, Y)
    Compute the refined output $Y_O = Y_I + R(Y_I)$
    Compute the loss using Equation (6)
    Update refinement network R using AdamW
  end for
end for
return prompts $p = \mathbb{E}_z[p_0 + Az]$ and network R

where $f_i(\cdot)$ denotes the i-th layer and its parameter is θ_i, and x_i is the input of the i-th layer.

Thus, based on Eq. (4), with residual connections, f_3 and f_1 can be optimized independently. Fortunately, vision-language models typically have residual connections, and the prediction refinement network also comprises a shortcut connection. Therefore, we can iteratively optimize the prompt generation and the prediction refinement module.

Moreover, to enhance training stability, we further propose a prediction-consistent loss. Specifically, we use an additional Kullback–Leibler divergence to constrain the output of the black-box model and the refinement module. Thus, the loss for the CMA-ES optimizer is formulated as follows:

$$\mathcal{L}_I = CE(Y_I, Y) + \lambda_I * KL(Y_I \| Y_O), \quad (5)$$

where λ_I is a hyper-parameter, Y_I denotes the output of the black-box model, Y_O is the output of the refinement network, Y represents the ground-truth label, CE is the cross-entropy loss, and KL is the Kullback-Leibler divergence. Similarly, during the optimization of the prediction refinement network, we also add a KL divergence loss, which serves as the regularization term, for training stabilization. The objective for the prediction refinement module can be written as follows:

$$\mathcal{L}_O = CE(Y_O, Y) + \lambda_O * KL(Y_O \| Y_I), \quad (6)$$

where λ_O is a hyper-parameter. Thus, the term "collaboratively" implies the algorithm can jointly optimize two modules while ensuring they work together through consistency loss rather than interfering with each other due to the sequential nature of the modules.

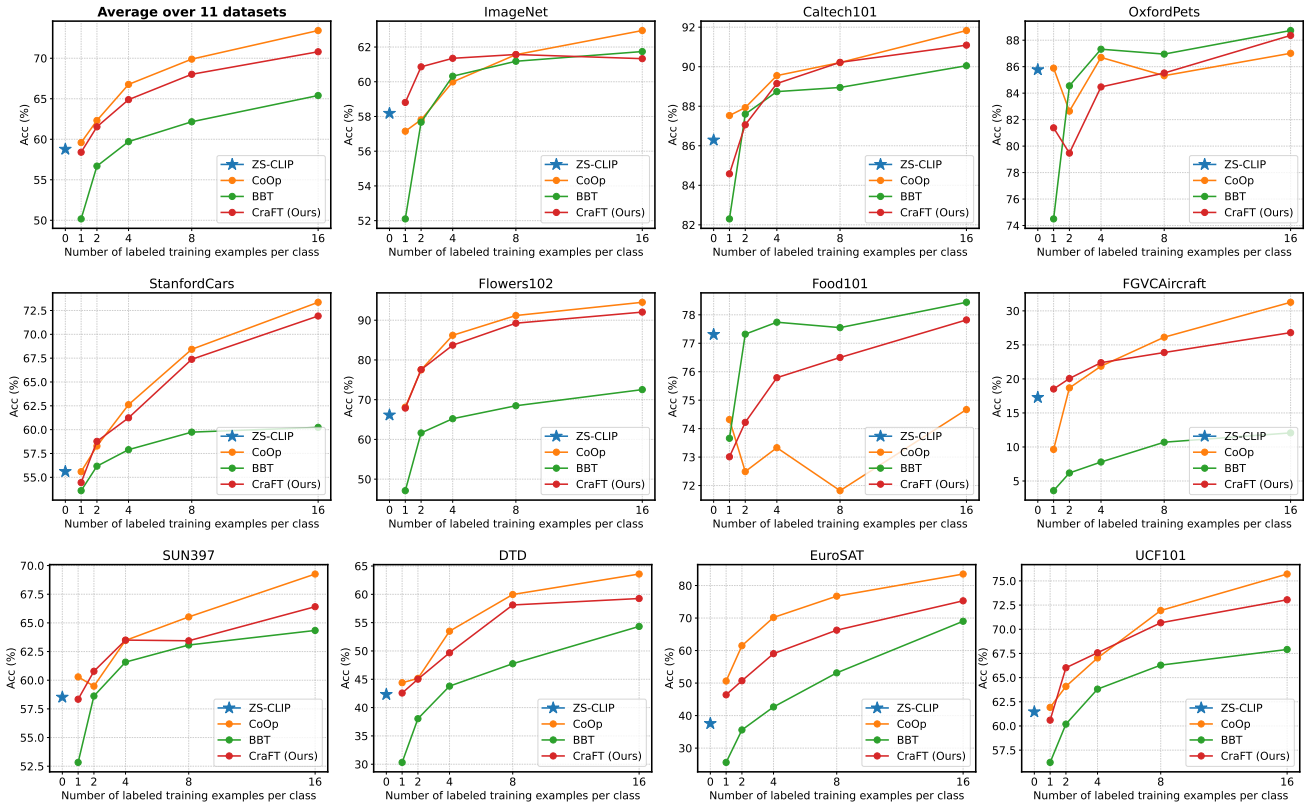

Figure 2: **Results of few-shot learning on the 11 datasets.** Here, CoOp is a white-box method that works as the upper bound. Our proposed method greatly surpasses the black-box baseline methods, BBT, and zero-shot CLIP.

4. Experiments

4.1. Setup

Datasets. In accordance with CoOp (Zhou et al., 2022b), we adopt 11 distinct image classification datasets to investigate few-shot learning. These datasets encompass various domains of image classification, including generic object recognition with ImageNet (Deng et al., 2009) and Caltech101 (Li et al., 2004), fine-grained image recognition with OxfordPets (Parkhi et al., 2012), StanfordCars (Krause et al., 2013), Flowers102 (Nilsback & Zisserman, 2008), Food101 (Bossard et al., 2014) and FGVCAircraft (Maji et al., 2013), satellite image classification with EuroSAT (Helber et al., 2019), action classification with UCF101 (Soomro et al., 2012), texture classification with DTD (Cimpoi et al., 2014), and scene recognition with SUN397 (Xiao et al., 2010). Moreover, we utilize 4 additional datasets to investigate the robustness of the model to distribution, including ImageNetV2 (Recht et al., 2019), ImageNet-Sketch (Wang et al., 2019), ImageNet-A (Hendrycks et al., 2021b), and ImageNet-R (Hendrycks et al., 2021a). ImageNetV2 is a reproduced test set using different sources while following ImageNet's data collection process. ImageNet-Sketch contains sketch images belonging to the same 1,000 ImageNet classes. Both ImageNet-A and -R contains 200 classes derived from a subset of ImageNet's 1,000 classes.

Evaluation Protocol. To evaluate the performance of few-shot learning models, we have followed the evaluation protocol proposed in CLIP (Radford et al., 2021). Specifically, we have trained models using 1, 2, 4, 8, and 16 shots and evaluated them on the full test sets. Additionally, we have assessed the robustness of the models to distribution shift by training CraFT on ImageNet (Deng et al., 2009) with 16 shots and evaluating it on target datasets ImageNetV2 (Recht et al., 2019), ImageNet-Sketch (Wang et al., 2019), ImageNet-A (Hendrycks et al., 2021b), and ImageNet-R (Hendrycks et al., 2021a).

Training Details. We utilize CLIP (Radford et al., 2021) as our black-box vision-language model, with ResNet-50 (He et al., 2016) and transformer (Vaswani et al., 2017) serving as the vision and language encoders, respectively. These encoders are initialized with CLIP's pretrained weights and kept frozen and unseen during training. To optimize the text prompts in the prompt generation module, we used the CMA-ES algorithm and set the prompt length to 4. The text prompts are projected into a subspace of dimension 512 using a random matrix sampled from a Gaussian distribution $\mathcal{N}(0, 0.02)$. The population size is set to 40, with a budget of 8,000 API calls. For the prediction refinement module,

Table 1: **Results of different architectures on 11 datasets.** The models are trained on the 16-shot setting datasets. **Bold** denotes the best results of black-box methods.

Backbone	Method	Pets	Flo	FGVC	DTD	Euro	Cars	Food	SUN	Cal	UCF	IN	Avg.
ResNet-50	ZS-CLIP (Radford et al., 2021)	85.77	66.14	17.28	42.32	37.56	55.61	77.31	58.52	86.29	61.46	58.18	58.77
	BBT (Sun et al., 2022b)	**88.73**	72.53	12.07	54.33	69.01	60.24	**78.44**	64.34	90.05	67.91	**61.74**	65.40
	CraFT	88.36	**92.04**	**26.80**	**59.26**	**75.30**	**71.92**	77.82	**66.41**	**91.09**	**73.05**	61.33	**71.22**
ResNet-101	ZS-CLIP (Radford et al., 2021)	86.75	64.03	18.42	38.59	32.59	66.23	80.53	58.96	89.78	60.96	61.62	59.86
	BBT (Sun et al., 2022b)	**89.44**	75.48	24.02	54.79	64.73	67.84	**81.77**	**65.52**	92.97	70.74	**64.35**	68.33
	CraFT	89.20	**89.61**	**28.05**	**58.85**	**67.16**	**71.57**	81.37	64.79	**93.21**	**75.16**	63.68	**71.15**
ViT-B/32	ZS-CLIP (Radford et al., 2021)	87.49	66.95	19.23	43.97	45.19	60.55	80.50	61.91	90.87	62.01	62.05	61.88
	BBT (Sun et al., 2022b)	**89.77**	74.14	18.72	55.85	69.67	63.21	**81.44**	68.08	**94.20**	72.67	**65.18**	68.45
	CraFT	88.11	**90.43**	**26.24**	**60.40**	**70.07**	**70.45**	77.89	**68.57**	93.79	**75.72**	62.76	**71.31**
ViT-B/16	ZS-CLIP (Radford et al., 2021)	89.21	71.34	24.72	44.39	47.60	65.32	86.06	62.50	92.94	66.75	66.73	65.23
	BBT (Sun et al., 2022b)	**92.70**	82.41	29.49	59.26	70.48	70.19	**86.42**	70.33	**94.75**	70.48	**70.15**	72.42
	CraFT	91.94	**93.92**	**36.89**	**63.28**	**72.07**	**78.11**	83.66	**70.97**	94.48	**79.78**	68.21	**75.76**

we use a three-layer MLP with a hidden dimension of 512 as the refinement network. We set the hyper-parameters λ_I and λ_O to 0.1 divided by the number of classes by default. The prediction refinement module is optimized using the AdamW optimizer with a learning rate of 0.001, and we set the batch size as 256 during training. Results are reported with average accuracy. All experiments are conducted on a single NVIDIA GeForce RTX 3090. We conducted three runs with different seeds and averaged the results to obtain a reliable estimate of model performance.

Baseline Methods. To evaluate the effectiveness of CraFT, we compare it with three baseline methods. **(1) ZS-CLIP:** Our first baseline method is zero-shot CLIP (Radford et al., 2021). This method requires handcrafted prompts, which we set to be the same as those used in previous works (Zhou et al., 2022b;a) to ensure a fair comparison. **(2) CoOp:** Our second baseline method is CoOp (Zhou et al., 2022b). CoOp is a white-box method that proposes learning the global text prompts through gradient descent. We use the best version of CoOp (Zhou et al., 2022b), setting the length of text prompts to 16, for comparison. **(3) BBT:** Our third baseline method is Black-Box Tuning (BBT) (Sun et al., 2022b). BBT is a black-box method for NLP tasks that proposes optimizing the soft prompts with the CMA-ES algorithm. We implement BBT in the black-box VLM, and we set its hyper-parameters the same as CraFT.

4.2. Results of Few-Shot Classification

Figure 2 illustrates the performance of our proposed method, CraFT, in comparison to three baseline methods: CoOp (Zhou et al., 2022b), BBT (Sun et al., 2022b), and ZS-CLIP (Radford et al., 2021), across 11 downstream datasets, accompanied by their respective average results. Our proposed approach demonstrates a significant superiority over the other black-box methods, i.e., ZS-CLIP and BBT. Specifically, under the 16-shot setting, CraFT achieves a substantial accuracy improvement of 12.45% and 5.82% when compared to ZS-CLIP and BBT, respectively.

Our proposed CraFT surpasses the black-box baseline BBT (Sun et al., 2022b) on most datasets except OxfordPets (Parkhi et al., 2012) and Food101 (Bossard et al., 2014). OxfordPets (Parkhi et al., 2012) and Food101 (Bossard et al., 2014) are fine-grained datasets and therefore sensitive to the fine-tuning process. Since gradient-based optimization estimates the gradient on batch inputs, it can lead to unstable training and hurt the good properties of the pretrained model. Therefore, on OxfordPets (Parkhi et al., 2012) and Food101 (Bossard et al., 2014) datasets, BBT, which is optimized without gradient, performs significantly better than gradient-based methods (CoOp and our CraFT).

4.3. Effectiveness of Different Architectures

We further evaluate the effectiveness of our proposed method on the 11 datasets with different visual architectures of CLIP (Radford et al., 2021), containing both CNNs (He et al., 2016) and ViTs (Dosovitskiy et al., 2021). Table 1 shows the results of our methods and two black-box baselines, ZS-CLIP and BBT, with different model architectures. These methods are trained on downstream 16-shot datasets.

On average, our proposed CraFT method outperformed zero-shot CLIP (Radford et al., 2021) by 12.45%, 11.29%, 9.43%, and 10.53% on ResNet-50, ResNet-101, ViT-B/32, and ViT-B/16-based CLIP, respectively. Additionally, CraFT outperforms BBT (Sun et al., 2022b) by 5.82%, 2.82%, 2.86%, and 3.34% on average of the 11 datasets on ResNet-50, ResNet-101, ViT-B/32, and ViT-B/16 based CLIP, respectively. These results demonstrate the effectiveness of CraFT across different black-box model architectures.

4.4. Robustness to Distribution Shift

We further conduct experiments to evaluate the robustness of CraFT to distribution shift. Specifically, we trained the models using the 16-shot ImageNet (Deng et al., 2009) dataset and subsequently transferred them to target domain shift datasets. These included ImageNetV2 (Recht et al., 2019), ImageNet-Sketch (Wang et al., 2019), ImageNet-

Table 2: **Robustness to distribution shift.** We compare our method with CLIP and CoOp (prompt length $L = 4$ and $L = 16$). And the models are trained on 16-shot datasets with different architectures. **Bold** and Underline denote the highest and second highest results.

Backbone	Method	Black-Box	Source	Target				
			ImageNet	-V2	-Sketch	-A	-R	Avg.
ResNet-50	ZS-CLIP (Radford et al., 2021)	✓	58.18	51.34	33.32	21.65	56.00	40.58
	CoOp (L=4) (Zhou et al., 2022b)	✗	**63.33**	**55.40**	**34.67**	23.06	56.60	**42.43**
	CoOp (L=16) (Zhou et al., 2022b)	✗	62.95	55.11	32.74	22.12	54.96	41.23
	CraFT	✓	61.33	53.92	34.01	**23.13**	**58.23**	42.32
ResNet-101	ZS-CLIP (Radford et al., 2021)	✓	61.62	54.81	38.71	28.05	64.38	46.49
	CoOp (L=4) (Zhou et al., 2022b)	✗	65.98	58.60	**40.40**	29.60	64.98	**48.40**
	CoOp (L=16) (Zhou et al., 2022b)	✗	**66.60**	**58.66**	39.08	28.89	63.00	47.41
	CraFT	✓	63.68	56.95	39.50	**30.40**	**65.94**	48.20
ViT-B/32	ZS-CLIP (Radford et al., 2021)	✓	62.05	54.79	40.82	29.57	65.99	47.79
	CoOp (L=4) (Zhou et al., 2022b)	✗	66.34	**58.24**	**41.48**	**31.34**	65.78	**49.21**
	CoOp (L=16) (Zhou et al., 2022b)	✗	**66.85**	58.08	40.44	30.62	64.45	48.40
	CraFT	✓	62.76	56.55	40.26	31.27	**66.46**	48.63
ViT-B/16	ZS-CLIP (Radford et al., 2021)	✓	66.73	60.83	46.15	47.77	73.96	57.18
	CoOp (L=4) (Zhou et al., 2022b)	✗	71.73	**64.56**	**47.89**	**49.93**	75.14	**59.38**
	CoOp (L=16) (Zhou et al., 2022b)	✗	**71.92**	64.18	46.71	48.41	74.32	58.41
	CraFT	✓	68.21	62.78	47.17	49.73	**75.52**	58.80

A (Hendrycks et al., 2021b), and ImageNet-R (Hendrycks et al., 2021a).

Table 2 reports the results of our method and two other baseline methods: ZS-CLIP (Radford et al., 2021) and CoOp (Zhou et al., 2022b) (prompt length $L = 4$ and $L = 16$). Our proposed CraFT outperforms ZS-CLIP on all datasets and architectures, with improvements of 1.74%, 1.71%, 0.84%, and 16.2% observed for ResNet-50, ResNet-101, ViT-B/32, and ViT-B/16-based CLIP, respectively. These results illustrate that CraFT enhances the robustness of CLIP (Radford et al., 2021). Moreover, our method CraFT achieves comparable performance with the white-box prompt tuning method, CoOp. Compared with the CoOp ($L = 16$) variant, which performs well on few-shot classification, our CraFT achieves improvements of 1.09%, 0.79%, 0.23%, and 0.39% for each architecture, yielding the effectiveness of our method.

4.5. Ablation Study

Effectiveness of Components. In this part, we analyze the efficacy of the components of CraFT. Table 3 displays the average results obtained from 11 downstream datasets for different shot settings. In the table, "PG." refers to the prompt generation module, "PR." indicates the prediction refinement module, and "Co." stands for the collaborative training algorithm.

The results demonstrate that using either the prompt generation module or the prediction refinement module in isolation achieves superior performance compared to ZS-CLIP (58.77%) in downstream tasks. This indicates the effectiveness of both the prompt generation module and the prediction refinement module. However, when optimizing them iteratively without using the collaborative training algorithm, the model performs even worse than using the prediction refinement module alone. After incorporating the collaborative training algorithm, the models exhibit better performance compared to the other settings, indicating the effectiveness of this component. Therefore, it can be concluded that both the prompt generation module and prediction refinement module are effective, and they work best when optimized together using the collaborative training.

Table 3: We ablate the components of CraFT with different shots. PG. denotes the prompt generation module. PR. denotes the prediction refinement module. Co. denotes the collaborative training algorithm. Results are average over 11 dataset.

PG.	PR.	Co.	1	2	4	8	16
✓	✗	✗	50.17	56.69	59.71	62.16	65.40
✗	✓	✗	54.54	59.07	63.13	66.07	69.29
✓	✓	✗	51.54	56.93	60.70	64.47	68.23
✓	✓	✓	**59.49**	**61.87**	**65.26**	**68.44**	**71.22**

Ablation of the Prediction Refinement Network. Furthermore, we investigate the best architecture of the prediction refinement module. In Table 7, we ablate the effectiveness of the residual connection and the architecture of the refinement network R. As shown in Table 7, the performance of CraFT drops dramatically if we delete the residual connection (-30.91% on the 1-shot setting), indicating its effectiveness. Additionally, changing the MLP refinement network to a linear mapping also results in a significant performance drop. As a result, we implement the prediction refinement module using the MLP as the refinement network together with a shortcut connection.

Ablation of Hyper-parameters. Moreover, we ablate the

Table 4: Comparison of deployment efficiency, the viability of black-box, test accuracy, training time, and memory footprint of user and server. Models are trained using RN50 CLIP with the 16-shot ImageNet.

Method	Black-Box	Test Accuracy	Training Time	Mem. (User)	Mem. (Server)
ZS-CLIP (Radford et al., 2021)	✓	58.18	0	0	244.7 MB
CoOp (Zhou et al., 2022b)	✗	62.95	2h 3min	395.7 MB	0
CraFT	✓	61.33	1h 44min	5.0 MB	244.7 MB

Table 5: **Ablation of hyper-parameter** λ_I. Models are trained using RN50 CLIP in EuroSAT.

λ_I	0.00	0.01	0.03	0.05	0.07	0.09
CraFT	72.51	75.30	73.83	76.86	72.83	70.72

Table 6: **Ablation of hyper-parameter** λ_O. Models are trained using RN50 CLIP in EuroSAT.

λ_O	0.00	0.01	0.03	0.05	0.07	0.09
CraFT	74.45	75.30	74.92	75.91	74.83	75.49

sensitivity of the hyper-parameters λ_I and λ_O. The default values for λ_I and λ_O are set to 0.1 divided by the number of classes, which is 0.01 in this case. As shown in Table 5 and Table 6, the model performance does not exhibit a significant correlation with the values of λ_I and λ_O. Nevertheless, it consistently outperforms the case where lambda is set to 0, indicating the effectiveness of the consistency loss. Consequently, we maintain the default values of λ_I and λ_O across all experimental settings and datasets.

Comparison of Efficiency. We further evaluate the efficiency of CraFT and compare it with CoOp (Zhou et al., 2022b) and ZS-CLIP (Radford et al., 2021) on the 16-shot ImageNet dataset, based on deployment efficiency, black-box viability, test accuracy, training time, and memory footprint. Table 4 shows that CraFT trains faster and has a significantly smaller memory footprint, using only 1/80 of the memory footprint of the white-box method CoOp, and CraFT incurs only a marginal loss in test accuracy.

We further report the query efficacy of our method. For the black-box vision-language model, we first upload the image dataset to the server, and we query the model with the text prompts to obtain its output $f(\{t_i\}_{i=1}^K, \{i_j\}_{j=1}^N) \in \mathbb{R}^{N\times K}$. Figure 3 depicts the relationship between the model's performance and the number of queries on the ImageNet dataset. The results indicate that our method achieves a high degree of efficacy in terms of query efficiency. Our method outperforms the ZS-CLIP approach with a mere 1,000 queries. These results suggest its potential usefulness in a range of applications involving black-box vision-language models.

Table 7: We ablate the components of the prediction refinement module. Arch. denotes the architecture of the refinement network. **Bold** denotes the highest result.

Residual	Arch.	shots 1	2	4	8	16
✗	MLP	28.58	44.44	52.58	59.25	63.10
✓	Linear	47.99	52.70	56.74	62.84	65.88
✓	MLP	**59.49**	**61.87**	**65.26**	**68.44**	**71.22**

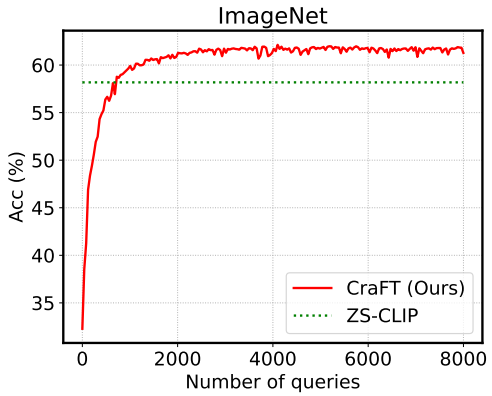

Figure 3: The relation of performance and number of queries on 16-shot ImageNet. The models utilize RN50 CLIP.

5. Conclusion

In this paper, we present **C**ollabo**ra**tive **F**ine-**T**uning (**CraFT**), a novel approach for fine-tuning black-box VLMs. CraFT comprises a prompt generation module and a prediction refinement module, designed to learn the text prompts and refine the black-box output prediction, respectively. Moreover, we developed a novel collaborative training algorithm capable of optimizing both modules jointly and mitigating conflicts between them. We demonstrate the effectiveness of CraFT over 15 datasets, as well as its robustness to distribution shifts and different architectures. Moreover, without the need for access to the parameters of vision-language models, CraFT improves its performance with marginal deployment cost and training costs. These results demonstrate the effectiveness of our method.

Acknowledgements

This work was funded by the Beijing Nova Program under Grant Z211100002121108, the National Natural Science Foundation of China under Grant 62276256, and the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Aghajanyan, A., Gupta, S., and Zettlemoyer, L. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *ACL*, 2021.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *ECCV*, 2014.

Chen, F.-L., Zhang, D.-Z., Han, M.-L., Chen, X.-Y., Shi, J., Xu, S., and Xu, B. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023a.

Chen, G., Yao, W., Song, X., Li, X., Rao, Y., and Zhang, K. Prompt learning with optimal transport for vision-language models. In *ICLR*, 2023b.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *CVPR*, 2014.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Derakhshani, M. M., Sanchez, E., Bulat, A., da Costa, V. G. T., Snoek, C. G., Tzimiropoulos, G., and Martinez, B. Bayesian prompt learning for image-language model generalization. *ICCV*, 2023.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

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

Guo, Z., Wei, Y., Liu, M., Ji, Z., Bai, J., Guo, Y., and Zuo, W. Black-box tuning of vision-language models with effective gradient approximation. In *Findings of EMNLP*, 2023.

Hansen, N., Müller, S. D., and Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evol. Comput.*, 11(1):1–18, 2003.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Helber, P., Bischke, B., Dengel, A., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *J-STARS*, 12(7):2217–2226, 2019.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021a.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In *CVPR*, 2021b.

Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., Lv, T., Cui, L., Mohammed, O. K., Liu, Q., et al. Language is not all you need: Aligning perception with language models. In *NeurIPS*, 2023.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

Kandasamy, K., Schneider, J., and Póczos, B. High dimensional bayesian optimisation and bandits via additive models. In *ICML*, 2015.

Kim, W., Son, B., and Kim, I. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *ICCVW*, 2013.

Li, F.-F., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPRW*, 2004.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. C. H. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

Liang, J., Hu, D., Feng, J., and He, R. Dine: Domain adaptation from single and multiple black-box predictors. In *CVPR*, 2022.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

Lu, J., Batra, D., Parikh, D., and Lee, S. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

Lu, Y., Liu, J., Zhang, Y., Liu, Y., and Tian, X. Prompt distribution learning. In *CVPR*, 2022.

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

Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J. D., Chen, D., and Arora, S. Fine-tuning language models with just forward passes. In *NeurIPS*, 2023.

Mei, Y., Omidvar, M. N., Li, X., and Yao, X. A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization. *ACM Trans. Math. Softw.*, 42(2):1–24, 2016.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

Oh, C., Hwang, H., Lee, H.-y., Lim, Y., Jung, G., Jung, J., Choi, H., and Song, K. Blackvip: Black-box visual prompting for robust transfer learning. In *CVPR*, 2023.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ouali, Y., Bulat, A., Matinez, B., and Tzimiropoulos, G. Black box few-shot adaptation for vision-language models. In *CVPR*, 2023.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *CVPR*, 2012.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

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

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., and Dai, J. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020.

Sun, T., He, Z., Qian, H., Zhou, Y., Huang, X.-J., and Qiu, X. Bbtv2: Towards a gradient-free future with large language models. In *EMNLP*, 2022a.

Sun, T., Shao, Y., Qian, H., Huang, X., and Qiu, X. Black-box tuning for language-model-as-a-service. In *ICML*, 2022b.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019.

Wang, X., Chen, G., Qian, G., Gao, P., Wei, X.-Y., Wang, Y., Tian, Y., and Gao, W. Large-scale multi-modal pre-trained models: A comprehensive survey. *Machine Intelligence Research*, 20(4):447–482, 2023a.

Wang, Z., Liang, J., He, R., Xu, N., Wang, Z., and Tan, T. Improving zero-shot generalization for clip with synthesized prompts. In *ICCV*, 2023b.

Wang, Z., Liang, J., Sheng, L., He, R., Wang, Z., and Tan, T. A hard-to-beat baseline for training-free clip-based adaptation. In *ICLR*, 2024.

Wen, J.-R., Huang, Z., and Zhang, H. Editorial for special issue on large-scale pre-training: Data, models, and fine-tuning. *Machine Intelligence Research*, 20(2):145–146, 2023.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

Yang, J., Peng, X., Wang, K., Zhu, Z., Feng, J., Xie, L., and You, Y. Divide to adapt: Mitigating confirmation bias for domain adaptation of black-box predictors. In *ICLR*, 2023.

Yu, L., Chen, Q., Lin, J., and He, L. Black-box prompt tuning for vision-language model as a service. In *IJCAI*, 2023.

Zhang, R., Zhang, W., Fang, R., Gao, P., Li, K., Dai, J., Qiao, Y., and Li, H. Tip-adapter: Training-free adaption of clip for few-shot classification. In *ECCV*, 2022.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Conditional prompt learning for vision-language models. In *CVPR*, 2022a.

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

A. Comparison of Black-Box Setting in VLMs

Table 8: **Comparision of black-box settings.** We compare the black-box assumption in this paper with assumptions in other methods in three aspects: VLM requirements, Network requirements, and Access Permission.

Method	Black-Box Assumption		
	VLM requirements	Network Requirements	Access Permission
LFA (Ouali et al., 2023)	**Dual-towered**	**No requirements**	**Features**
CBBT (Guo et al., 2023)	**Dual-towered**	**No requirements**	**Features**
Yu et al. (2023)	**No requirements**	**Vision Transformer**	**Predictions**
CraFT (Ours)	**No requirements**	**No requirements**	**Predictions**

To illustrate the validity of our black-box assumption, we compared our black-box assumption with black-box assumptions in previous methods. As shown in Table 8, we are comparing them in terms of VLM requirements, Network requirements, and Access Permission.

In comparison to LFA (Ouali et al., 2023) and CBBT (Guo et al., 2023), our approach does not impose specific requirements on VLM, while LFA and CBBT need it to be dual-towered. Notably, some VLMs are not dual-towered, and they have complex interactions between language and vision in the model, such as ALBEF (Li et al., 2021) and KOSMOS-1 (Huang et al., 2023). The setup in these methods may not be effective for these models. Moreover, these methods require the black-box VLM to return features of each modality, potentially compromising model ownership. Granting such access to users poses a significant risk, as it opens the door for model theft through distillation. Therefore, many black-box methods (Yang et al., 2023; Liang et al., 2022) opt for returning predictions instead of features.

In comparison to Yu et al. (2023), a crucial assumption in their work is that the vision encoder employed is a vision transformer. This enables the incorporation of learnable prompts in the visual encoder, and the prompts can be updated through CMA-ES. However, it is worth noting that many vision encoders in the realm of VLM are not based on the ViT architecture, such as the ResNet-based CLIP. Consequently, this assumption imposes constraints on the applicability of their methodology. Furthermore, in Table 9 presented below, a comparison between our CraFT and Yu et al. (2023) in the same architecture reveals that our method outperforms theirs, underscoring the efficacy of our approach.

In summary, we believe our black-box definition for VLM is more accurate and applicable to a wider range of scenarios.

B. More Experimental Results

To further verify the effectiveness of our method, we compare our method with two more black-box optimization methods, BlackVIP (Oh et al., 2023) and Yu et al. (2023). We do not include the results in the main text due to the time-intensive nature of training these methods. Additionally, the black-box framework assumed in Yu et al. (2023) is not consistent with ours. Their framework presupposes access to the visual architecture of VLMs, assuming it to be a vision transformer. In contrast, we assume the model is invisible, and no assumptions are made about its architecture in our setting. We believe our black-box setting is more reasonable and has broader applicability.

The comparison results are shown in Table 9. We trained these methods with ViT-B/16 CLIP on 16-shot datasets. To align with Yu et al. (2023), we train a variant of our method where we also incorporate visual prompts (vp) in the black-box optimization. As shown in the table, our method achieves the best results on average over the 11 datasets, even without including the vision prompts. It surpasses BBT (Sun et al., 2022b), BlackVIP (Oh et al., 2023), and Yu et al. (2023) by 3.34%, 7.54%, and 2.04%. Moreover, after incorporating the vision prompts in optimization, our method achieves an additional gain of 2.64%, demonstrating that our method is compatible with visual prompts.

Moreover, we report the training time of these methods and our method in Table 10. The results show that our method is quite efficient compared with the baselines, which takes only about 2 days for training but achieves the best performance. Moreover, we notice that including optimizing visual prompts will greatly increase the training time. There may be two reasons contributing to this issue. Firstly, it increases the search dimensions of CMA-ES, which consequently decelerates the optimization algorithm. Secondly, for each image, multiple insertions of visual prompts are required (i.e., different sampled solutions in CMA-ES), followed by forward computations to obtain visual representations. In contrast, when visual prompts

are not optimized, a single forward pass is sufficient to obtain the image representation in the whole training process.

In summary, the results show that our method is effective and efficient, and can be extended with visual prompts.

Table 9: **Comparing with more black-box methods.** The methods are trained with ViT-B/16 CLIP on 16-shot datasets. 'vp' denotes that we include the same visual prompts for optimization as Yu et al. (2023) for a fair comparison.

Method	Pets	Flo	FGVC	DTD	Euro	Cars	Food	SUN	Cal	UCF	IN	Avg.
BBT	92.70	82.41	29.49	59.26	70.48	70.19	86.42	70.33	94.75	70.48	**70.15**	72.42
BlackVIP	89.70	70.60	25.00	45.20	73.10	65.60	86.60	64.70	93.70	69.10	67.10	68.22
Yu et al. (2023)	93.43	82.87	30.08	59.24	82.33	68.39	**86.94**	69.91	94.42	74.09	69.24	73.72
CraFT	91.94	93.92	36.89	63.28	72.07	78.11	83.66	70.97	94.48	79.78	68.21	75.76
CraFT + vp	**93.12**	**95.40**	**41.17**	**66.86**	**82.40**	**80.95**	83.75	**72.60**	**95.29**	**80.83**	70.05	**78.40**

Table 10: **The time cost of training these methods.** We report the total training time of BBT (Sun et al., 2022b), BlackVIP (Oh et al., 2023), Yu et al. (2023), and our methods on the 11 16-shot datasets with 3 random seeds. The training time is computed on a single RTX 3090.

	BBT (Sun et al., 2022b)	BlackVIP (Oh et al., 2023)	Yu et al. (2023)	CraFT	CraFT+vp
Training Time	$\sim$ 2 days	$\sim$ 1 month	$\sim$ 2 weeks	$\sim$ 2 days	$\sim$ 2 weeks

C. Dataset Statistics

In the experiments, we selected a set of 15 public datasets to assess the efficacy of our method and the baseline methods. A comprehensive overview of the datasets, including detailed statistics, is presented in Table 11.

Table 11: Detailed statistics of datasets used in experiments.

Dataset	# Classes	# Training	# Test	Task
OxfordPets	37	2,944	3,669	fine-grained pets recognition
Flowers102	102	4,093	2,463	fine-grained flowers recognition
FGVCAircraft	100	3,334	3,333	fine-grained aircraft recognition
DTD	47	2,820	1,692	Textural recognition
EuroSAT	10	13,500	8,100	Satellite image recognition
StanfordCars	196	6,509	8,041	Fine-grained car recognition
Food101	101	50,500	30,300	Fine-grained food recognition
Sun397	397	15,880	19,850	Scene recognition
Caltech101	100	4,128	2,465	Object recognition
UCF101	101	7,639	3,783	Action recognition
ImageNet	1,000	1.28M	50,000	Object recognition
ImageNetV2	1,000	-	10,000	Robustness of collocation
ImageNet-Sketch	1,000	-	50,889	Robustness of sketch domain
ImageNet-A	200	-	7,500	Robustness of adversarial
ImageNet-R	200	-	30,000	Robustness of rendition styles