ZIP: AN EFFICIENT ZEROTH-ORDER PROMPT TUNING FOR BLACK-BOX VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent research has introduced various approaches for prompt-tuning black-box vision-language models, referred to as black-box prompt-tuning (BBPT). While BBPT has demonstrated considerable potential, it is often found that many existing methods require an excessive number of queries (*i.e.*, function evaluations), which poses a significant challenge in real-world scenarios where the number of allowed queries is limited. To tackle this issue, we propose Zeroth-order Intrinsic-dimensional Prompt-tuning (ZIP), a novel approach that enables efficient and robust prompt optimization in a purely black-box setting. The key idea of ZIP is to reduce the problem dimensionality and the variance of zeroth-order gradient estimates, such that the training is done fast with far less queries. We achieve this by re-parameterizing prompts in low-rank representations and designing intrinsic-dimensional clipping of gradients. We evaluate ZIP on 13+ vision-language tasks in standard benchmarks and show that it achieves an average improvement of approximately 6% in few-shot accuracy and 48% in query efficiency compared to the best-performing alternative BBPT methods, establishing a new state of the art. Our ablation analysis further shows that the proposed clipping mechanism is robust and nearly optimal, without the need to manually select the clipping threshold, matching the result of expensive hyperparameter search.

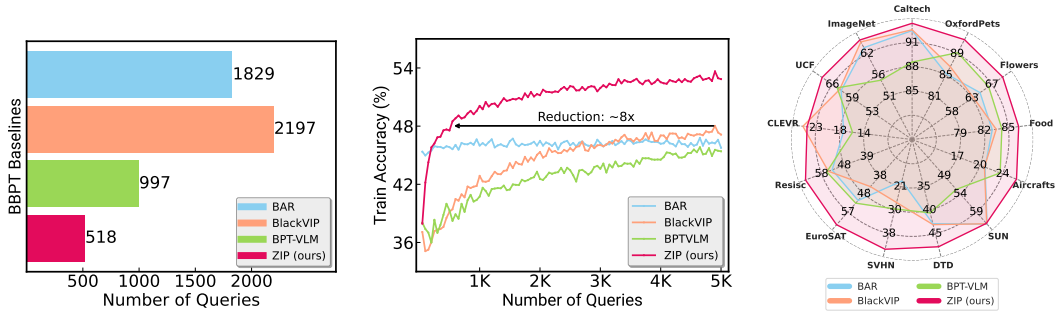
1 INTRODUCTION

Foundation models pre-trained on a vast amount of data are creating tremendous success across a wide range of applications in various domains (Ramesh et al., 2021; Radford et al., 2021; Jia et al., 2021; Singh et al., 2022; Copet et al., 2024; Liu et al., 2024). A notable example is CLIP (Radford et al., 2021) which learns visual concepts from natural language supervision and works zero-shot at inference.

In fact, these models are fine-tuned for specific downstream tasks at deployment to create yet more performance refinement in practice (Liu et al., 2022). However, fine-tuning these models is not only computationally expensive, but also requires full access to model specifications. The complication here is that many high-performing foundation models are provided only as a software-as-a-service (OpenAI, 2023; Google, 2023) without model details due to commercial interests and security concerns.

To overcome this challenge, recent works have suggested to fine-tune such *black-box* models via so-called black-box prompt-tuning (BBPT) (Sun et al., 2022b; Diao et al., 2023; Oh et al., 2023; Yu et al., 2023); *i.e.*, by parameterizing input prompts and only optimizing them via classic derivative-free optimization methods such as evolutionary strategies (Hansen & Ostermeier, 2001; Hansen et al., 2003) and zeroth-order optimization (Spall, 1992; 1997; Ghadimi & Lan, 2013), it enables fine-tuning without access to the model details or back-propagation.

Nevertheless, it still remains a major challenge to secure query-efficiency in existing BBPT methods. Specifically, we find that many existing approaches require excessive model evaluations (*i.e.*, queries), often spanning several tens of thousands times (Tsai et al., 2020; Oh et al., 2023), and moreover, they result in significant performance drop when they are given a limited query budget. This is quite critical in many practical scenarios where large models are provided in the form of prediction APIs, and users can only make use of it with a limited budget.



(a) Query efficiency (b) Training progress (c) Generalization performance

Figure 1: ZIP performance summary: (a) number of queries required to reach a given target accuracy, (b) training accuracy vs. number of queries, and (c) test set errors, measured on 13 standard vision-language tasks; the first two are arithmetic mean. ZIP shows its outstanding performance compared to other black-box prompt-tuning methods (*e.g.*, BAR (Tsai et al., 2020), BLACKVIP (Oh et al., 2023), BPTVLM (Yu et al., 2023); the details are provided in Appendix D.4) in terms of both training and generalization performances.

In this work, we propose a new method called ZIP: Zeroth-order Intrinsic-dimensional Prompt-tuning to tackle this challenge. The key idea is to reduce the dimensionality of the problem (hence the term "intrinsic") and the variance of zeroth-order gradients, such that the training is done fast with far less queries, and subsequently, improves generalization. In essence, we achieve this by re-parameterizing prompts in low-rank representations in effective forms and designing intrinsic-dimensional clipping of zeroth-order gradients (Section 4).

Fundamentally, we are inspired by a line of previous works that hint at the dimension dependency of zeorth-order methods (Spall, 1992; Ghadimi & Lan, 2013; Duchi et al., 2015) and noise in stochastic methods (Bottou et al., 2018) causing optimization difficulty when it comes to large models. Indeed, we find this pertinent to BBPT in our empirical analysis, in which we show that a naive zeroth-order method suffers from increased dimensionality unlike the first-order counterpart (Section 3).

Our extensive experimental results show that ZIP is extremely efficient and robust, setting a new state of the art. To be specific, we evaluate ZIP on 13+ datasets for various vision-language tasks in standard benchmarks including few-shot learning, base-to-new generalization, cross-dataset transfer, and out-of-distribution generalization tasks, and across all, we find that ZIP consistently outperforms existing BBPT methods by substantial margins in terms of prediction accuracy, while demanding far less number of queries (Section 5). We provide a summary of how ZIP performs in Figure 1.

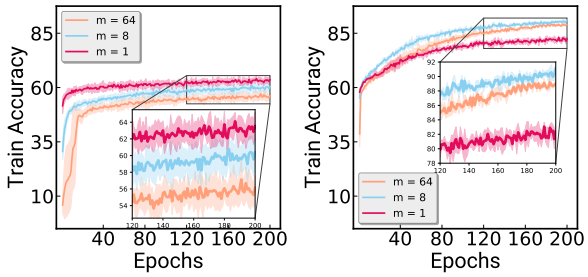
2 BACKGROUND

Prompt-tuning is an emerging paradigm to update large pre-trained models before utilizing them for various downstream tasks (Lester et al., 2021; Liu et al., 2023). It works by prepending learnable context tokens to the input prompts embedding and training them on some data for a target task. Specifically, we can formulate it as an optimization problem as follows:

$$\min_{\theta} f(\theta, \omega; \mathcal{D}) \tag{1}$$

where $\theta \in \mathbb{R}^d$ refers to the learnable parameters where $d = p \times m$ denotes the problem dimensionality with p and m being the word embedding dimensions and number of context tokens, respectively, and ω refers to the pre-trained model; also, f refers to the loss function, which is the cross-entropy for vision-language tasks in our case, and $\mathcal{D}$ is a given dataset.

However, prompt-tuning is not directly applicable to *black-box* models since back-propagation is not allowed, and thus, the optimization must be done derivative-free. To this end, many approaches have been developed to address this issue for namely derivative-free optimization (Larson et al., 2019). In particular, a series of evolution strategies such as covariance matrix adaptation (Hansen, 2016) has been used for black-box prompt tuning or BBPT (Sun et al., 2022b;a; Yu et al., 2023).



(a) Zeroth-order results (b) First-order results

Figure 2: Zeroth-order vs. first-order methods for prompt tuning under varying number of prompt parameters. The training progresses are plotted measured on the Flowers102 dataset; where we observe the same trend that the zeroth-order method suffers from increasing prompt dimensionality.

An alternative approach to prompt-tuning in a black-box situation is by *zeroth-order* optimization which leverages approximate gradients estimated from only function evaluations. In particular, one of the foundational techniques is the simultaneous perturbation stochastic approximation (SPSA) (Spall, 1992; 1997), by which the gradient estimate is computed as follows:

$$\widehat{\nabla} f(\theta; \mathcal{B}) = \frac{1}{N} \sum_{i=1}^{N} \frac{f(\theta + cz_i; \mathcal{B}) - f(\theta - cz_i; \mathcal{B})}{2c} (z_i)^{-1}, \tag{2}$$

where z_i refers to the perturbation vector randomly drawn from a probability distribution with zero mean and finite inverse moment, $(\cdot)^{-1}$ denotes element-wise reciprocal, and N indicates the number of perturbation vector samples used for one gradient estimate; also, c and $\mathcal{B}$ refer to a small positive scalar controlling the perturbation magnitude and the mini-batch of data points, respectively; *i.e.*, SPSA estimates gradients by simultaneously perturbing all dimensions in θ. Then, the optimization is done iteratively using the zeroth-order stochastic gradient descent (ZO-SGD) (Ghadimi & Lan, 2013) with (2) as follows:

$$\theta_{t+1} = \theta_t - \eta_t \widehat{\nabla} f(\theta_t; \mathcal{B}_t), \tag{3}$$

where η_t denotes the step size at iteration t.

This approach has been demonstrated to be effective for BBPT tuning of vision-language models (Oh et al., 2023; Tsai et al., 2020), and yet, in theory, zeroth-order methods can suffer from high variance and slow convergence, especially for high-dimensional problems (Spall, 1992; Ghadimi & Lan, 2013; Duchi et al., 2015). This means that one needs to query the model a high number of times, which we find is critical to secure the feasibility of BBPT in practice. As we show throughout this work, our key idea to fix this issue is to reduce the dimensionality of θ and the variance of $\widehat{\nabla} f$.

3 Motivation

In this section, we motivate our work by disclosing that naively applying a zeroth-order method can lead to a failure of BBPT. Precisely, we show that its performance deteriorates as the number of the context tokens (*i.e.*, dimension of trainable parameters) increases. This is rather unexpected because increasing their dimensions, in fact, is found to improve performance when optimized with a first-order method for prompt tuning.

To be specific, we optimize a vision-language model using both the basic zeroth-order or ZO (3) and first-order or FO (*i.e.*, SGD) methods, with varying the number of context token (1, 8, 64), and measure their training progress. We used CLIP (Radford et al., 2021), a representative pre-trained vision-language model widely used in the literature. The results are plotted in Figure 2. These experiments clearly demonstrate the influence of zeroth-order optimization on training speed and overall performance in the prompt tuning framework, leading to two key observations:

- General prompt tuning can benefit from increased parameters, achieving higher expressive power and performance, especially with a moderate number of context tokens (*e.g.*, 8 tokens).
- In contrast, prompt tuning with zeroth-order optimization suffers in both training speed and performance as the number of context tokens increases.

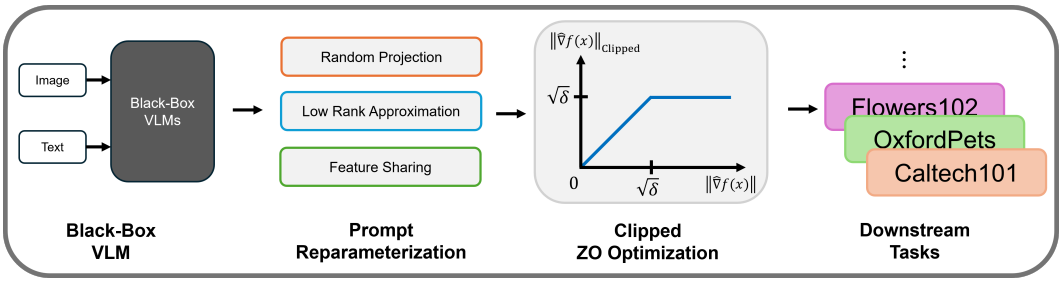


Figure 3: Overview of ZIP framework.

These findings highlight a fundamental limitation of zeroth-order optimization in prompt tuning: Although zeroth-order enables black-box scenario with prompt tuning, there exist noticeable dissonance between zeroth-order and prompt tuning, which prevent prompt tuning with zeroth-order benefits from increased number of context tokens.

Our research aims to alleviate this dissonance. This approach represents a significant departure from previous studies on BBPT, which have primarily focused on applying derivative-free optimization to specific domains, such as LLMs (Sun et al., 2022b) or vision-language models (Yu et al., 2023; Oh et al., 2023). By focusing on narrowing the behavior between general prompt tuning and zeroth-order optimization, our work addresses the fundamental challenges of zeroth-order optimization in prompt tuning, offering a novel direction that enhances both efficiency and effectiveness in BBPT.

Additionally, we prove the convergence rate of ZO-SGD (3) with the version (2) to show its dependency on the problem dimensionality d, highlighting the limitation of the zeroth-order approach. The result is provided in Theorem 1 of Appendix B.2.

4 ZIP: ZEROTH-ORDER INTRINSIC-DIMENSIONAL PROMPT-TUNING

In this section, we introduce ZIP to address inherent discrepancy between zeroth-order optimization and prompt tuning. Specifically, ZIP enables to use more context tokens without the drawbacks from query inefficiency and reduced performance associated with zeroth-order methods. To this end, we reduce the number of learnable parameters by leveraging the concept of intrinsic dimension (Li et al., 2018; Aghajanyan et al., 2021) which suggests that training with a low-dimensional reparameterization can be as effective as using the full parameter space. We then further reparameterization with low-rank approximation style with diagonal term to reduce more parameters, (Section 4.1) and introduce feature sharing technique to improve model expressibility (Section 4.2). In addition, we introduce a gradient clipping technique tailored for zeroth-order optimization (Section 4.3), to further enhance the efficiency and effectiveness of ZIP. An overview of the ZIP framework is provided in Figure 3.

4.1 PROMPT TUNING IN LOWER DIMENSIONAL SPACES

As discussed in previous sections, the performance of zeroth-order optimization methods depends heavily on the problem dimensionality (Ghadimi & Lan, 2013; Duchi et al., 2015). To improve the query efficiency of BBPT, we propose reducing the optimization space dimensionality.

Inspired by intrinsic dimensionality (Li et al., 2018; Aghajanyan et al., 2021), we project learnable parameters of each context tokens $\theta_i \in \mathbb{R}^p$ onto a lower-dimensional space $\boldsymbol{v}_i \in \mathbb{R}^q$ using a Fastfood transform[1] (Le et al., 2014) matrix $\mathbf{M}_i \in \mathbb{R}^{p \times q}$. With it, the total number of trainable parameters are reduced from $d = p \times m$, to $d' = q \times m$ with $d' \ll d$. Each vector indexed by i corresponds to the i-th trainable context token where m indicates the total number of context tokens. The random projection can then be expressed as follows:

$$\theta_i = \theta_{0,i} + \mathbf{M}_i \boldsymbol{v}_i \tag{4}$$

where $\theta_{0,i}$ is initial parameters. With this reparameterization, we can project learnable parameter to much lower dimension, from d to d'.

[1]We use this transformation for computational efficiency, as in Li et al. (2018); Aghajanyan et al. (2021).

Further, we apply additional reparameterization with a low-rank approximation style as below, to get trainable parameter matrix $\mathbf{W}$:

$$\mathbf{W} = [\boldsymbol{v}'_1|\boldsymbol{v}'_2|\cdots|\boldsymbol{v}'_m] = \mathbf{U}\text{diag}(\boldsymbol{s})\mathbf{V}^T \tag{5}$$

We initialized $\mathbf{U}$ with standard normal distribution, $\mathbf{V}$ to zeros, and $\boldsymbol{s}$ to ones, ensuring $\mathbf{W}$ starts at zero. Through this reparameterization, the trainable parameters transform to $\mathbf{U} \in \mathbb{R}^{q\times r}$, $\mathbf{V} \in \mathbb{R}^{r\times m}$, and $\boldsymbol{s} \in \mathbb{R}^r$. We argue that this reparameterization allows to reduce number of trainable parameters while perserving key characteristics of original parameter as possible, which can effectively improve our method (see Section 6.2).

Unlike conventional low-rank approximation style (Hu et al., 2022), we interpose a diagonal matrix, which can effectively adjust importance of dimension by letting diagonal parameters directly scaling each dimension independently. We compare our approach with naive low rank decomposition with two square matrices in Appendix C.3.

4.2 Enhancing expressiveness with feature sharing

While reducing the number of parameters can accelerate training speed with zeroth-order, it may also reduce the model expressive power. To address this, we introduce a feature sharing technique, which incorporate a vector $\boldsymbol{u} \in \mathbb{R}^q$ within $\mathbf{W}$, which can serve as a common base across the partitioned vectors. The vector $\boldsymbol{u}$ is integrated into original vectors v'_i, by using a outer product with a vector $\mathbb{1} \in \mathbb{R}^m$, forming the final trainable parameter matrix Ξ:

$$\begin{aligned}\Xi &= \mathbf{W} + \boldsymbol{u} \otimes \mathbb{1} \\ &= \mathbf{U}\text{diag}(\boldsymbol{s})\mathbf{V}^T + \boldsymbol{u} \otimes \mathbb{1} \\ &= [\boldsymbol{v}'_1 + \boldsymbol{u}|\boldsymbol{v}'_2 + \boldsymbol{u}|\cdots|\boldsymbol{v}'_m + \boldsymbol{u}] \\ &= [\boldsymbol{w}_1|\boldsymbol{w}_2|\cdots|\boldsymbol{w}_m]\end{aligned} \tag{6}$$

where each $\boldsymbol{w}_i \in \mathbb{R}^q$ is a mixed vector that blends the original components $\boldsymbol{v}_i$ with the feature sharing by $\boldsymbol{u}$. With the incorporation of feature sharing, the updated parameters for context tokens are then computed as:

$$\theta_i = \theta_{0,i} + \mathbf{M}_i\boldsymbol{w}_i \tag{7}$$

With the feature sharing, we argue that the capability of the model to capture complex patterns enhances, leading to improved performance. This techinique can be done by adding neligible amount of learnable parameters. We will empirically validate the importance of the feature sharing in Section 6.3.

4.3 Reducing variance with intrinsic-dimensional clipping

Through a series of reparameterization schemes, we obtained the final trainable parameters matrix Ξ, in which there are $\delta = r(q + m + 1) + q$ parameters in total. Now, the problem (1) reduces to

$$\min_{\Xi} \; f(\Xi, \omega; \mathcal{D}). \tag{8}$$

One can consider employing ZO-SGD (3) to solve this problem, and yet, as demonstrated in Section 3, it can still cause slow convergence in practice due to its large variance.

To address this issue, we propose a simple yet robust zeroth-order method based on what we call intrinsic-dimensional gradient clipping mechanism defined as follows

$$\Xi_{t+1} = \Xi_t - \eta_t\alpha_t\widehat{\nabla}f(\Xi_t, \omega; \mathcal{B}), \tag{9}$$

where α_t is a scaling factor defined as follows

$$\alpha_t = \min\left(\frac{\sqrt{\delta}}{\sqrt{\sum_{i=1}^{\delta}\widehat{\nabla}f(\Xi_t, \omega; \mathcal{B})_i^2}}, 1\right), \tag{10}$$

where δ refers to the problem dimensionality as mentioned above; *i.e.*, it clips the zeroth-order stochastic gradient estimates $\widehat{\nabla}f$ if its norm exceeds $\sqrt{\delta}$ as a threshold, while iteratively updating Ξ_t. There are several interesting aspects of this method as described below.

First, the immediate advantage of this approach is that there is no need to manually select the clipping threshold (which is prone to be suboptimal) or perform an expensive hyperparameter search. Considering that gradient clipping can accelerate the optimization process in general (Zhang et al., 2019), and yet, that an appropriate choice of the threshold value is required, this advantage is certainly nontrivial. We validate this adaptivity by showing that the threshold chosen based on (10) is, quite surprisingly, nearly optimal across diverse training workloads in Section 6.1.

Also, the threshold being expressed in terms of the problem dimensionality δ, in particular, $\sqrt{\delta}$, should be reasonably inspiring, considering research results in the literature. Specifically, we can interpret that previous work suggests setting the clipping threshold (for general first-order methods) to be the standard deviation of estimated gradients (Zhang et al., 2020a; Pascanu et al., 2012; Zhang et al., 2019; 2020b). While this is again not quite practical to compute, we notice that it can be done relatively straightforwardly for zeroth-order optimization, since the variance of zeroth-order gradients is inherently bounded in terms of the problem dimensionality δ, thus the standard deviation being $\sqrt{\delta}$. We explicitly show this in Lemma 2 of Appendix B.1.

5 Evaluations

5.1 Experimental setup

Datasets and Tasks. To assess the query efficiency and performance of ZIP, we conduct evaluations on standard generalization tasks following the protocols of Zhou et al. (2022a;b); Oh et al. (2023). These tasks include few-shot learning, base-to-new generalization, cross-dataset transfer, and out-of-distribution (OOD) generalization. For few-shot learning, base-to-new generalization, and cross-dataset transfer, we evaluate ZIP across 13 diverse image classification tasks: ImageNet (Deng et al., 2009), Caltech101 (Fei-Fei et al., 2004), OxfordPets (Parkhi et al., 2012), Flowers102 (Nilsback & Zisserman, 2008), Food101 (Bossard et al., 2014), FGVCAircraft (Maji et al., 2013), SUN397 (Xiao et al., 2010), Resisc45 (Cheng et al., 2017), DTD (Cimpoi et al., 2014), SVHN (Netzer et al., 2011), EuroSAT (Helber et al., 2019), CLEVR (Johnson et al., 2017), and UCF101 (Soomro et al., 2012). For evaluating OOD generalization, we employ four established OOD datasets to measure ZIP's robustness under distribution shifts: ImageNetV2 (Recht et al., 2019), ImageNet-Sketch (Wang et al., 2019), ImageNet-A (Hendrycks et al., 2021b), and ImageNet-R (Hendrycks et al., 2021a).

Baselines. To thoroughly evaluate the performance of ZIP, we compare it against a variety of baselines: (1) manual prompt, where manually composed prompts to conduct the evaluation (human-written prompts are detailed in Table 8); (2) state-of-the-art BBPT approaches for VLMs including BAR (Tsai et al., 2020), BLACKVIP (Oh et al., 2023) and BPTVLM (Yu et al., 2023). For all baselines, we follow the standardized few-shot evaluation protocol across datasets, consistent with Zhou et al. (2022b); Oh et al. (2023), which includes specific few-shot splits to ensure a fair comparison.

Implementation details. We mainly experiment using the CLIP (Radford et al., 2021) model with vision transformer (Dosovitskiy et al., 2021), keeping the CLIP model frozen. We consistently set the number of context tokens m as 8 for ZIP and use 5,000 queries across all tasks for all BBPT baselines. The number of the intrinsic dimensionality d' is set to 500, and the rank of low-rank matrices $r = 5$, resulting in a total of 417 learnable parameters δ with the formula $r(\lfloor d'/m \rfloor + m + 1) + \lfloor d'/m \rfloor$. Following the previous works for transfer learning (Zhou et al., 2022a;b; Oh et al., 2023), we initialize soft prompts from prompts derived from source tasks. We use the official code to reproduce BBPT baselines, and the results are averaged over three different random seeds.

5.2 Generalization performance

We present empirical evidence showcasing the effectiveness and robustness of our proposed method, ZIP, across 13+ vision-language tasks. Our results, summarized in Table 1, 2, and 3, cover evaluations on few-shot accuracy, base-to-new generalization, and cross-dataset transfer with out-of-distribution generalization. The experiments reveal two main insights: (i) ZIP consistently outperforms other BBPT baselines across various tasks; (ii) ZIP achieves better robustness to unseen data distribution compared to existing BBPT methods; Detailed analyses of these findings are provided below.

Table 1: Few-shot performance on 13 vision-language tasks. All the results are based on 16-shots per class. The **bold numbers** denote the highest accuracy of all baselines on each dataset, and the underlined values indicate the second. ZIP clearly outperforms other BBPT baselines.

Method	#Params	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	SUN397	DTD	SVHN	EuroSAT	Resisc45	CLEVR	UCF101	ImageNet	**Average**
Manual Prompt	0k	93.0	89.1	**70.6**	85.9	24.8	**62.6**	44.1	18.8	48.1	58.1	14.5	67.5	**66.7**	57.2
BAR	37.6k	92.5	85.6	65.0	83.0	21.6	62.4	42.9	19.8	51.6	53.9	18.1	63.5	64.0	55.7
BLACKVIP	9.9k	92.6	86.9	63.5	83.5	21.5	62.3	43.1	27.5	44.4	55.5	**25.9**	64.0	65.5	56.6
BPTVLM	4.0k	88.6	89.4	66.9	84.2	24.0	53.2	40.6	29.8	53.0	56.2	16.4	64.8	55.5	55.6
ZIP	0.4k	**93.4**	**91.7**	70.0	**86.3**	**26.6**	62.2	**47.8**	**44.2**	**64.2**	**65.2**	25.1	**69.8**	66.0	**62.5**

Table 2: Base-to-new generalization performance. H represents the harmonic mean, providing a balanced measure of accuracy across seen and unseen classes (Xian et al., 2017). ZIP consistently outperforms BAR, BLACKVIP, and BPTVLM across base, new, and harmonic mean evaluations.

Method	Set	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	SUN397	DTD	SVHN	EuroSAT	Resisc45	CLEVR	UCF101	ImageNet	**Average**
BAR	Base	96.5	87.3	67.5	87.5	25.6	69.2	51.2	23.5	60.2	66.1	27.5	66.4	69.6	61.4
BLACKVIP		**96.6**	87.7	67.9	87.6	25.8	69.0	51.8	26.4	66.4	69.9	38.9	67.0	70.3	63.5
BPTVLM		93.2	90.6	66.9	88.7	29.1	65.3	53.2	45.4	70.3	72.0	41.5	68.3	66.3	65.4
ZIP		**96.6**	**94.9**	**72.1**	**89.9**	**29.8**	**70.3**	**61.7**	**52.9**	**84.0**	**81.6**	**50.1**	**75.1**	**72.1**	**71.6**
BAR	New	**94.5**	94.9	73.2	88.9	29.2	**74.6**	**55.8**	27.3	**72.1**	62.3	27.1	**73.3**	64.9	64.5
BLACKVIP		93.2	90.9	**74.5**	89.4	30.9	73.9	55.4	21.8	48.8	61.2	28.0	72.6	**66.8**	62.1
BPTVLM		92.7	95.8	72.7	85.4	**32.3**	64.8	45.3	40.1	47.0	61.3	**28.4**	65.0	55.2	60.5
ZIP		93.2	**97.0**	73.4	**90.0**	32.0	71.5	51.0	**45.8**	64.4	**65.2**	26.8	69.5	65.6	**65.0**
BAR	Harmonic	**95.5**	90.9	70.2	88.2	27.3	**71.8**	53.4	25.3	65.6	64.1	27.3	67.7	67.2	62.9
BLACKVIP		94.9	89.3	71.0	88.5	28.1	71.4	53.5	23.9	56.3	65.3	32.6	69.7	68.5	62.8
BPTVLM		92.9	93.1	69.7	87.0	30.6	65.0	48.9	42.6	56.3	66.2	33.7	66.6	60.2	62.9
ZIP		94.9	**95.9**	**72.7**	**89.9**	**30.9**	70.9	**55.8**	**49.1**	**72.9**	**72.5**	**34.9**	**72.2**	**68.7**	**68.2**

Table 3: Cross-dataset transfer and out-of-distribution generalization performance. After training on ImageNet (*i.e.*, source) with 16-shot data per class, ZIP is evaluated on 12 target datasets for CDT and 4 ImageNet variants for OOD. ZIP consistently demonstrates better transferability and generalizability, outperforming BAR, BLACKVIP, and BPTVLM.

	Source	**CDT Target**													**OOD Target**				
Method	ImageNet	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	SUN397	DTD	SVHN	EuroSAT	Resisc45	CLEVR	UCF101	**Average**	ImageNet-A	ImageNetV2	ImageNet-R	ImageNet-Sketch	**Average**
BAR	64.0	92.3	84.3	64.3	83.1	20.8	61.0	42.2	20.0	**49.6**	50.6	14.5	63.0	53.8	40.2	57.5	72.0	43.8	53.4
BLACKVIP	65.5	**92.5**	**86.2**	64.9	**83.6**	**22.3**	**62.0**	**43.3**	18.7	40.5	**55.7**	**15.2**	**64.1**	54.1	42.5	59.2	73.1	44.6	54.9
BPTVLM	55.5	80.7	77.7	50.3	77.6	16.3	43.8	30.8	15.5	34.6	37.7	12.4	54.8	44.4	32.7	46.7	61.7	33.5	43.7
ZIP	**66.0**	90.4	85.6	**65.6**	**83.6**	20.5	60.6	40.9	**27.0**	42.3	55.6	14.5	63.6	**54.2**	**47.8**	**59.5**	**74.7**	**45.4**	**56.9**

Few-shot performance. As represented in Table 1, our experimental result indicates that ZIP consistently outperforms state-of-the-art BBPT approaches, including BAR, BLACKVIP, and BPTVLM, across 9 out of 13 datasets. On average, ZIP achieves accuracy gains of +6.8%, +5.9%, and +6.9% over BAR, BLACKVIP, and BPTVLM, respectively, demonstrating notable effectiveness in few-shot learning. In particular, ZIP excels on datasets requiring coarse semantic understanding, with improvements of +3.7% on DTD, +11.2% on EuroSAT, and +7.1% on Resisc45 compared to the second-best method. Additionally, ZIP shows remarkable performance in digit recognition, surpassing the next best method by +14.4% on the SVHN dataset, further highlighting its capability in few-shot learning.

Base-to-new generalization. Table 2 presents the base-to-new generalization results, where models are trained on base classes and evaluated on both base and new classes across 13 datasets. ZIP consistently outperforms all BBPT baselines, achieving the highest base, new, and harmonic mean scores. By leveraging its lower parameter count and the robustness of zeroth-order optimization, ZIP

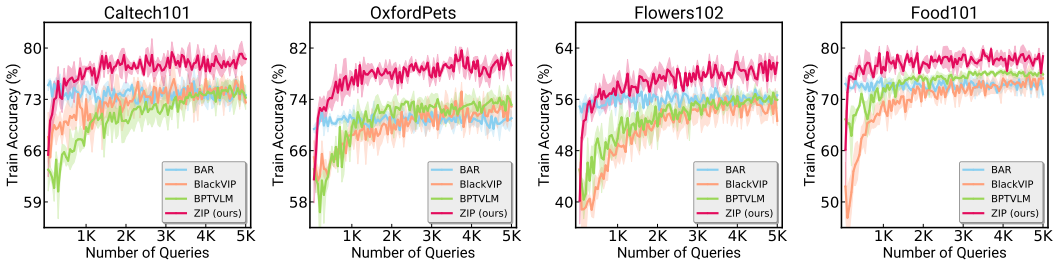


Figure 4: Training performance measured on Caltech101, OxfordPets, Flowers102, and Food101. We provide more results on other datasets in Figure 14.

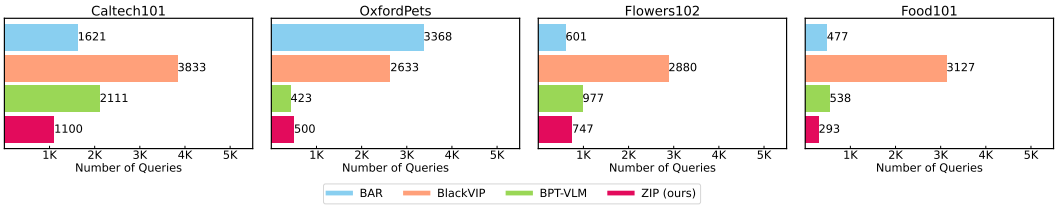


Figure 5: Number of queries to reach a target accuracy. For the datasets Caltech101, OxfordPets, Flowers102, Food101, and FGVCAircraft, ZIP requires fewer API calls to reach target accuracy thresholds in most cases. This demonstrates considerable query efficiency when compared to other BBPT methods. We provide more results on other datasets in Figure 15.

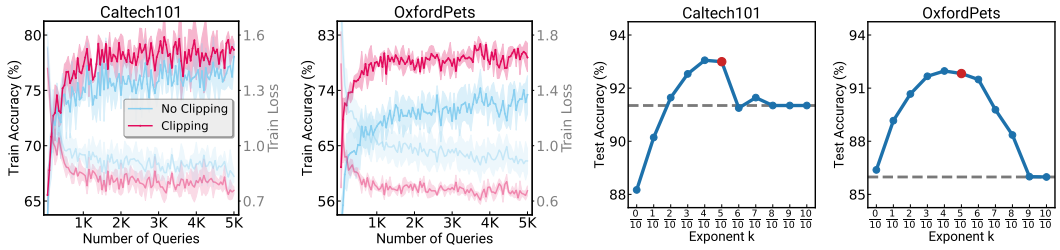
effectively mitigates overfitting, as its reduced model capacity and rough gradient estimates help avoid fitting to noisy outliers, resulting in better generalization.

Cross-dataset transfer & Out-of-distribution generalization. To assess robustness in challenging scenarios, we evaluate ZIP for cross-dataset transfer (CDT) and out-of-distribution (OOD) generalization. As shown in Table 3, ZIP, trained on ImageNet (*i.e.*, source), demonstrates competitive generalization capabilities in the CDT setting, achieving slight improvements of 0.4% over BAR and 0.1% over BLACKVIP, with a more notable gain of 9.8% over BPTVLM across 12 diverse target datasets. More significantly, in the OOD evaluations on four ImageNet variants, ZIP consistently outperforms all baselines, achieving substantial gains of 2.5% over BAR, 2.0% over BLACKVIP, and a remarkable 13.2% improvement over BPTVLM. These results highlight ZIP's exceptional robustness and adaptability in handling domain shifts, making it particularly effective for real-world applications where OOD generalization is critical.

5.3 QUERY EFFICIENCY

This section provides empirical evidence highlighting the query efficiency of ZIP. We start by tracking the training progress of different BBPT methods, ensuring all operate within the same computational budget. For a fair comparison, ZIP and other baselines are allocated a budget of 5,000 queries. This query budget was chosen to reflect a more practical scenario, as many existing methods often require thousands of epochs (Oh et al., 2023), which is unrealistic for real-world applications with strict API query limitations. By setting a more feasible budget, we aim to evaluate efficiency of each methods under conditions that closely resemble practical deployment settings.

Figure 4 shows that ZIP consistently achieves faster training speed and higher accuracy than other BBPT methods under identical query budget constraints. This efficiency is attributed to the effective combination of low-rank approximation with diagonal matrix and our specialized threshold for zeroth-order optimization, which accelerates training, while the feature sharing and compactness of the low-rank representation enhance overall performance. These design elements work synergistically, allowing ZIP to achieve rapid training progress and improved accuracy. To further analyze query efficiency, we evaluate the number of queries required to reach target accuracy, which is determined as the minimum of the maximum accuracy achieved by all methods. As shown in Figure 5, ZIP demonstrates strong query efficiency, achieving the best performance in datasets like Caltech101 and Food101, and maintaining competitive efficiency in OxfordPets and Flowers102, even when not the absolute best. The overall results, summarized in Figure 1a, show that ZIP achieves over a 48%



(a) Effects of our clipping threshold (in Section 6.1) (b) clipping method varying threshold δ^k

Figure 7: Effects of gradient clipping and optimal threshold. (a) Training accuracy comparison with and without gradient clipping. (b) Test accuracy with varying thresholds δ^k. The red point indicates ZIP's chosen threshold, which consistently achieves near-optimal accuracy.

Table 4: Benefits of low-rank approximation with diagonal matrix. Comparing our method against standard dimensionality reduction, demonstrating notable test accuracy improvements.

Method	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	SUN397	DTD	SVHN	EuroSAT	Resisc45	CLEVR	UCF101	ImageNet	**Average**
Standard	90.9	88.1	**67.5**	84.6	23.8	57.9	43.2	31.5	56.5	58.3	18.3	65.3	62.3	57.6
Ours	**93.1**	**90.8**	67.1	**86.0**	**25.2**	**59.0**	**44.4**	**40.9**	**60.6**	**63.3**	**20.2**	**67.4**	**64.8**	**60.2**

improvement in query efficiency compared to the second-best BBPT method. This indicates that ZIP utilizes its query budget effectively, making it particularly suited in resource-constrained scenarios.

We also compare ZIP 's query efficiency with first-order and naive zeroth-order optimization, using 8 tokens and 5,000 queries across all methods. As shown in Figure 6, ZIP bridges the gap between first-order and zeroth-order optimization, achieving training speeds similar to first-order on the OxfordPets dataset. While zeroth-order methods typically exhibit dependence on d for training speed, ZIP 's efficient design allows it to match first-order optimization behavior. This demonstrates ZIP's enhanced query efficiency, making it highly suitable for practical applications where efficient resource utilization is critical. Further details on query efficiency across additional datasets can be found in Figure 14, 15 and 16.

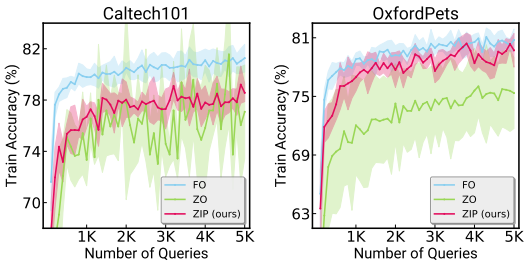


Figure 6: Training curves of first-order (FO), zeroth-order (ZO) and ZIP. ZIP effectively bridges the gap between FO and ZO with notably faster training and high accuracy.

6 ABLATIONS

6.1 INTRINSIC-DIMENSIONAL CLIPPING

In this section, we evaluate the effectiveness of our clipping method, with setting threshold as $\sqrt{\delta}$. We begin by tracking the training progress of ZIP with gradient clipping and the one without. As shown in Figure 7a, ZIP with our clipping threshold consistently achieves faster training speeds and higher accuracy, indicating its efficiency in enhancing zeroth-order optimization. This improvement is largely due to the variance-reducing nature of clipping, which results in more stable gradient estimates and consequently accelerates the training process.

To further validate the effectiveness of gradient clipping with our threshold, we compared $\sqrt{\delta}$ threshold against various alternative values to ensure its optimality. As shown in Figure 7b, the $\sqrt{\delta}$ threshold consistently achieved near-optimal performance on Caltech101 and OxfordPets, outperforming other clipping settings ranging from 1 ($= \delta^{0/10}$) to δ ($= \delta^{10/10}$). The gray dashed line, representing no

Table 5: Benefits of feature sharing over unshared. Integrating shared features consistently boosts model expressive power and accuracy across diverse tasks, demonstrating improved performance.

Method	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	SUN397	DTD	SVHN	EuroSAT	Resisc45	CLEVR	UCF101	ImageNet	**Average**
Unshared	93.1	90.8	67.1	86.0	25.2	59.0	44.4	40.9	60.6	63.3	20.2	67.4	65.2	60.2
Shared	**93.5**	**91.8**	**70.6**	**86.2**	**26.3**	**62.2**	**46.5**	**43.8**	**66.2**	**65.6**	**24.4**	**69.0**	**65.5**	**62.4**

clipping, further underscores the advantage of $\sqrt{\delta}$ threshold. These results highlight the effectiveness of the $\sqrt{\delta}$ threshold, demonstrating its capability as an efficient clipping strategy for zeroth-order optimization without requiring extensive hyperparameter tuning. Additional validation results on other datasets are available in Figure 20 and 21.

6.2 Low rank approximation with diagonal matrix

The low-rank approximation with a diagonal matrix is pivotal in enhancing both the efficiency and performance of our method. Unlike naive lower-dimensional projections, this approach effectively preserves the most crucial components of the parameter space, allowing for accelerated training without compromising the model's expressive power.

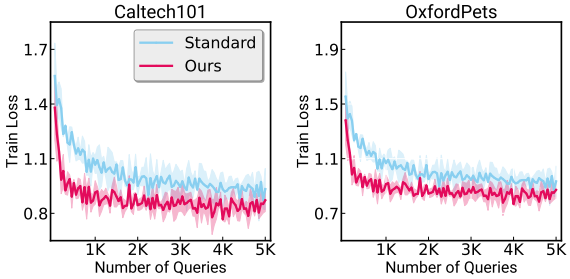


Figure 8: Effects of low-rank approximation with diagonal matrix. Our method improves training efficiency compared to standard dimensionality reduction.

As shown in Figure 8 and Table 4, this approach not only accelerates the training process but also improves model accuracy. For instance, the average accuracy across datasets increased from 57.6% to 60.2% with the application of low-rank approximation using a diagonal matrix. These gains highlight the technique's effectiveness in enhancing training efficiency and overall model performance, making it particularly advantageous for optimizing zeroth-order based prompt tuning compared to more straightforward projection methods. Additional results on other datasets further validating this improvement can be found in Figure 18.

6.3 Feature sharing

To evaluate the expressive power of feature sharing, we compared the performance of models with and without feature sharing. As shown in Table 5, models utilizing feature sharing consistently achieved higher accuracy, increasing the overall average score from 60.2% to 62.4%. These consistent gains across diverse datasets highlight the effectiveness of features sharing in retaining model expressiveness and improving performance, even when parameters are reduced. This confirms that feature sharing is a valuable technique for maintaining model accuracy while optimizing for efficiency.

7 Conclusion

In this paper, we propose Zip, a new method for prompt-tuning black-box vision-language models. Extensive experiments show that Zip outperforms state-of-the-art BBPT methods in generalization performance while offering faster training with far less number of queries.

Reproducibility statement

To ensure reproducibility, we provide detailed information on our experimental setup in Appendix D.3, including training and evaluation procedures. All datasets used are publicly available. The source code, along with implementation details and hyper-parameter settings, will be released in a public repository upon publication.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *IJCNLP*, 2021.

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

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 2018.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *NeurIPS*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Shizhe Diao, Zhichao Huang, Ruijia Xu, Xuechun Li, Yong Lin, Xiao Zhou, and Tong Zhang. Black-box prompt learning for pre-trained language models. *TMLR*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 2015.

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

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 2013.

Google. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 2001.

Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 2003.

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 Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021b.

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 *ICLR*, 2022.

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

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

Jeffrey Larson, Matt Menickelly, and Stefan M Wild. Derivative-free optimization methods. *Acta Numerica*, 2019.

Quoc Viet Le, Tamás Sarlós, and Alexander Johannes Smola. Fastfood: Approximate kernel expansions in loglinear time. *arXiv preprint arXiv:1408.3060*, 2014.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2024.

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 Computing Surveys*, 2023.

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

Yurii Evgen'evich Nesterov. A method of solving a convex programming problem with convergence rate o\bigl(k^2\bigr). In *Doklady Akademii Nauk*, 1983.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop*, 2011.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

Changdae Oh, Hyeji Hwang, Hee-young Lee, YongTaek Lim, Geunyoung Jung, Jiyoung Jung, Hosik Choi, and Kyungwoo Song. Blackvip: Black-box visual prompting for robust transfer learning. In *CVPR*, 2023.

OpenAI. ChatGPT. https://openai.com/blog/chatgpt/, 2023.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *ArXiv*, 2012.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *CVPR*, 2022.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012.

James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 1992.

James C. Spall. A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, 1997.

Tianxiang Sun, Zhengfu He, Hong Qian, Yunhua Zhou, Xuan-Jing Huang, and Xipeng Qiu. Bbtv2: towards a gradient-free future with large language models. In *EMNLP*, 2022a.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *ICML*, 2022b.

Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *ICML*, 2020.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *NeurIPS*, 2019.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *CVPR*, 2017.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

Lang Yu, Qin Chen, Jiaju Lin, and Liang He. Black-box prompt tuning for vision-language model as a service. In *IJCAI*, 2023.

Bohang Zhang, Jikai Jin, Cong Fang, and Liwei Wang. Improved analysis of clipping algorithms for non-convex optimization. *NeurIPS*, 2020a.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*, 2019.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *NeurIPS*, 2020b.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022b.

Table 6: Few-shot performance on 13 vision-language tasks with varying combinations of the proposed modules (*e.g.*, diagonal matrix, feature sharing (FS), and intrinsic-dimensional clipping). All the results are based on 16-shots per class. The **bold numbers** denote the highest accuracy of all baselines on each dataset, and the underlined values indicate the second.

Number	Diagonal	FS	Clipping	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	SUN397	DTD	SVHN	EuroSAT	Resisc45	CLEVR	UCF101	ImageNet	**Average**
1	✓	✗	✗	91.2	82.3	56.9	83.4	13.2	56.8	41.0	38.9	58.8	58.5	23.1	64.4	61.5	56.2
2	✗	✓	✗	90.1	89.3	65.3	84.6	22.7	60.6	42.4	38.4	59.3	59.8	18.9	66.7	63.4	58.6
3	✗	✗	✓	90.7	89.3	68.1	85.0	23.7	57.4	43.9	36.0	59.2	57.1	21.2	65.2	62.6	58.4
4	✓	✓	✗	91.3	86.0	59.7	83.4	16.6	58.9	46.1	**44.9**	61.2	59.2	23.0	64.8	59.0	58.0
5	✗	✓	✓	89.8	89.5	66.4	85.3	25.1	58.5	44.7	38.3	61.0	58.9	18.9	65.9	63.4	58.9
6	✓	✗	✓	93.1	90.8	67.1	86.0	25.2	59.0	44.4	40.9	60.6	63.3	20.2	67.4	64.8	60.2
7	✓	✓	✓	**93.4**	**91.7**	**70.0**	**86.3**	**26.6**	**62.2**	**47.8**	44.2	**64.2**	**65.2**	**25.1**	**69.8**	**66.0**	**62.5**

A ADDITIONAL MATERIALS FOR REVIEWER CLARIFICATIONS

During the rebuttal process, we have included additional figures, tables, and discussions to address specific questions and concerns raised by the reviewers. These materials are temporarily placed in Appendix A for clarity and ease of reference during this period. After the rebuttal process, they will be integrated into the appropriate sections of the main manuscript.

We appreciate the reviewers' insightful feedback, which has guided these additions to further clarify and substantiate our work. Please refer to the detailed captions accompanying each figure and table for an explanation of how they address the specific points raised.

A.1 DISCUSSION

In this section, we present key findings, discuss their implications, and propose potential directions for future research.

Analysis of module combinations. We evaluate all combinations of the proposed modules, including diagonal matrix, feature sharing (FS), and intrinsic-dimensional clipping. The results are presented in Table 6. First, we observe that using all the proposed modules together results in significantly better performance compared to using individual modules or pairs of modules. This demonstrates that each component works harmoniously to contribute to the generation of effective results. Additionally, from the transitions $1 \rightarrow 6$, $4 \rightarrow 7$ and $5 \rightarrow 7$, we find that combining the low-rank approximation with diagonal matrix with intrinsic dimensional clipping yields more pronounced performance improvements (+4%, +4.5%, +3.6%) compared to other combinations. These findings suggest that while each component is effective on its own, their combination creates a complementary synergy that maximizes overall performance. In future work, we plan to conduct an in-depth analysis to uncover the underlying mechanisms behind this synergy. This will provide deeper insights into its practical utility, paving the way for its application to a broader range of tasks.

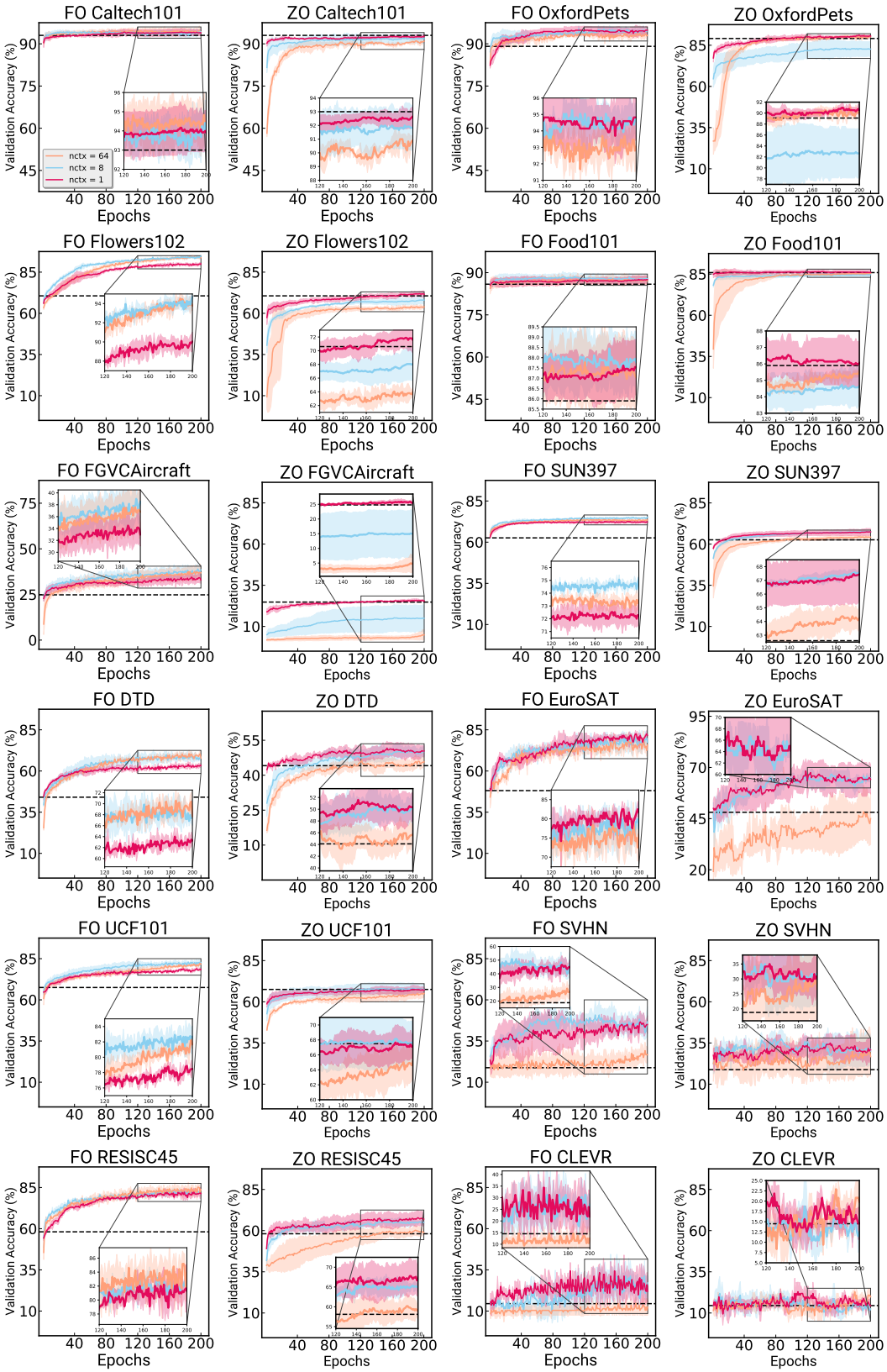


Figure 9: Validation curves illustrating the performance of different optimization methods across various vision-language tasks. The black dotted line represents the manual prompt ($m = 0$).

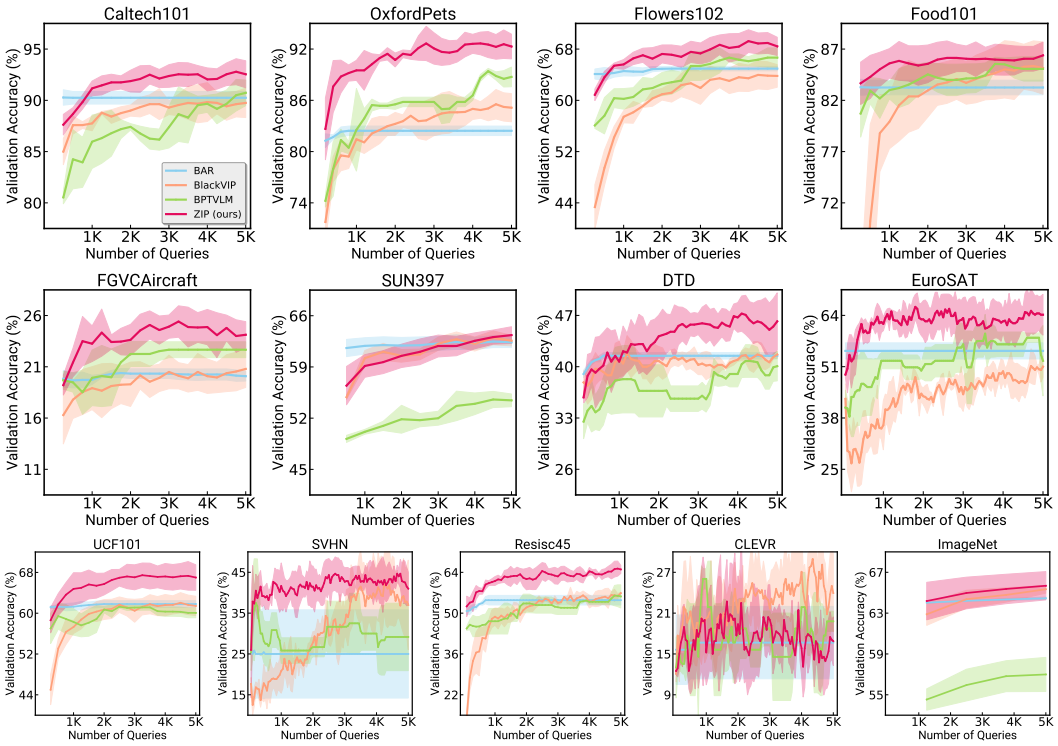


Figure 10: Validation curves with 5,000 query budgets across various vision-language tasks.

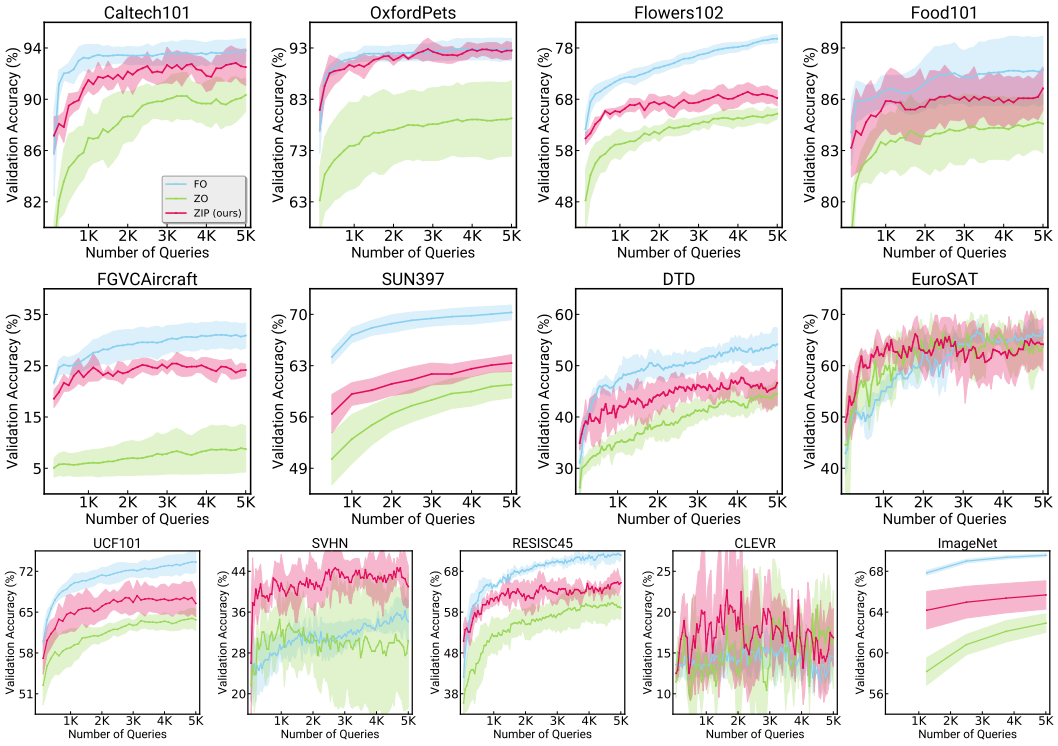


Figure 11: Validation curves of first-order, zeroth-order and ZIP across various vision-language tasks.

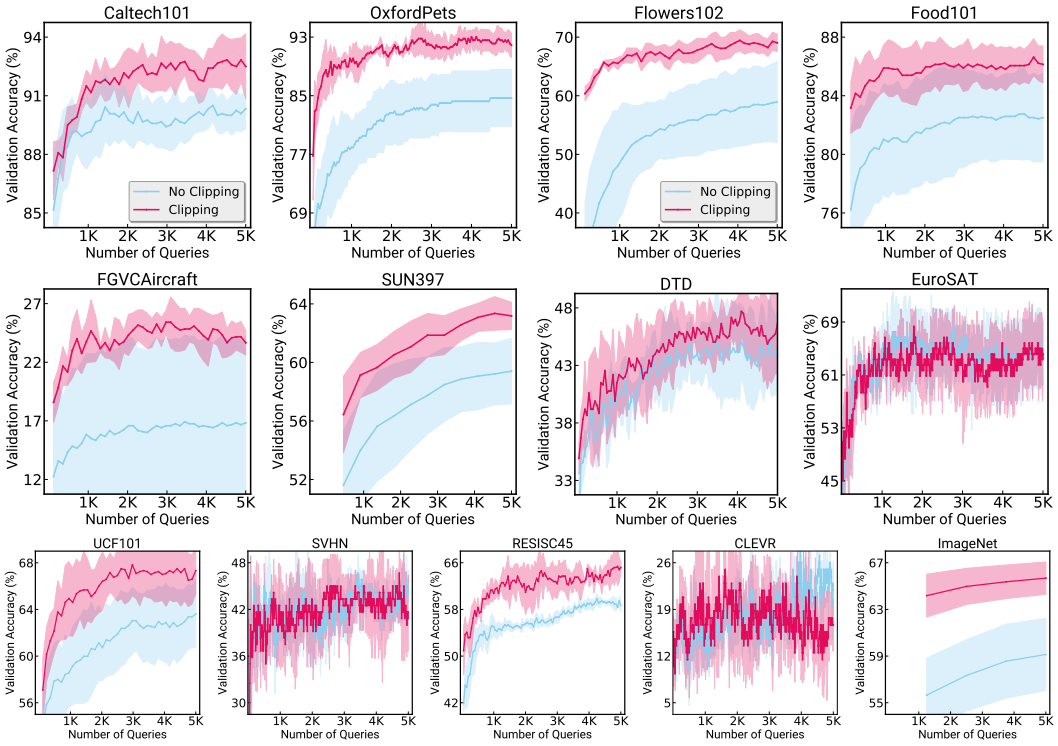


Figure 12: Validation curves showing the impact of zeroth-order gradient clipping across various vision-language tasks.

B THEORETICAL ANALYSIS

B.1 ASSUMPTION & LEMMA

Assumption 1. *On the function $f(\cdot)$, there exists some $L > 0$ such that for all x, y, we have $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$*

Lemma 1. *(Unbiasdness of ZO-SGD) In the $c_t \to 0$ limit, ZO-SGD is a unbiased estiamtor of FO-SGD in terms of random perturbation vector, which follows a Bernoulli distribution of two different values with equal absolute value and probability. That is,*

$$\mathbb{E}_{\{z_n\}_{n=1}^N}(\widehat{\nabla} f(\theta_t; \mathcal{B}_t)) = \nabla f(\theta_t; \mathcal{B}_t) \tag{11}$$

Proof of Lemma 1. Note that as $c_t \to 0$ limit, we have

$$\widehat{\nabla} f(\theta_t; \mathcal{B}_t) = \frac{1}{N}\sum_{i=1}^{N}(z_i)^{-1}(z_i)^\top \nabla f(\theta_t; \mathcal{B}_t)$$

let $A^k \in R^{d\times d}$ be a matrix of $(z_k)^{-1}(z_k)^\top$, then we get

$$A_{ij}^k = \begin{cases} 1 & \text{if } i = j \\ \frac{z_{kj}}{z_{ki}} & \text{otherwise} \end{cases}$$

Note that z_{ni} is a i-th element for the vector z_n. By taking expectation in terms of z_n over matrix A, we can get

$$\mathbb{E}_{\{z_n\}_{n=1}^N}(A_{ij}^n) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

since z_t^n have zero inverse moment and zero mean as we assumed. Therefore,

$$\mathbb{E}_{\{z_n\}_{n=1}^N}(\widehat{\nabla} f(\theta_t; \mathcal{B}_t)) = \nabla f(\theta_t; \mathcal{B}_t)$$

as desired. □

Lemma 2. *(Second moment of ZO-SGD) In the $c_t \to 0$ limit, second moment of ZO-SGD in terms of random perturbation vector, which follows a Bernoulli distribution of two different values with equal absolute value and probability. That is,*

$$\mathbb{E}_{\{z_n\}_{n=1}^N}(\|\widehat{\nabla} f(\theta_t; \mathcal{B}_t)\|^2) = \frac{d}{N}\|\nabla f(\theta_t; \mathcal{B}_t)\|^2 \tag{12}$$

Proof of Lemma 2. Starting from Lemma 1, zeroth-order gradient can be represented as below.

$$\widehat{\nabla} f(\theta_t; \mathcal{B}_t) = \frac{1}{N}\sum_{n=1}^{N} A^n \nabla f(\theta_t; \mathcal{B}_t)$$

Therefore, the second moment of zeroth-order gradient

$$\mathbb{E}_{\{z_n\}_{n=1}^N}(\|\widehat{\nabla} f(\theta_t; \mathcal{B}_t)\|^2) = \mathbb{E}_{\{z_n\}_{n=1}^N}(\frac{1}{N}\sum_{n=1}^{N} \nabla f(\theta_t; \mathcal{B}_t)^\top (A^n)^\top A^n \nabla f(\theta_t; \mathcal{B}_t))$$

let $B^n \in R^{d\times d}$ be a result of $(A^n)^\top A^n$, we can get

$$B_{ij}^n = \begin{cases} d & \text{if } i = j \\ \sum_{k=1}^{d} \frac{(z_{ni})^2}{z_{nj} z_{ni}} & \text{otherwise} \end{cases}$$

Taking expectation over matrix B^n, we can get

$$\mathbb{E}_{\{z_n\}_{n=1}^N}(B_{ij}^n) = \begin{cases} d & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

By plugging above results, the second moment of zeroth-order gradient is

$$\mathbb{E}_{\{z_n\}_{n=1}^N}(\|\widehat{\nabla} f(\theta_t; \mathcal{B}_t)\|^2) = \frac{d}{N}\|\nabla f(\theta_t; \mathcal{B}_t)\|^2.$$

as desired.

□

Lemma 3. *With assumption 1, for any unbiased gradient estimate* $\widehat{\nabla} f(\theta_t; z, \mathcal{B}_t)$,

$$\mathbb{E}(f(\theta_{t+1})|x_t) \leq f(\theta_t) - \eta \|\nabla f(\theta_t)\|^2 + \frac{L}{2}\eta^2 \left\| \widehat{\nabla} f(\theta_t; z, \mathcal{B}_t) \right\|^2$$

B.2 Convergence analysis of zeroth-order optimization

The high variance of zeroth-order gradient estimates stems from the estimation process involving random perturbations, introducing an additional problem dimension (d) related terms in convergence compared with corresponding first-order (FO) methods. Although the convergence rate was originally proven by Ghadimi & Lan (2013), we have also confirmed similar convergence behavior using Spall (1992) approach. Note that we assumed z_i has zero inverse moment, as it is sampled from a Bernoulli distribution of two different values with equal absolute value and probability in practice. The convergence rate of ZO-SGD using (2) is as follows:

Theorem 1 (Convergence rate of ZO-SGD)**.** *Under Assumption 1, in the* $c_t \to 0$ *limit, when* $\eta = \sqrt{\frac{2NF}{LGd}}\sqrt{\frac{1}{T}}$ *where* $F := f(x_0) - f(x_*)$ *and sampling the* z_n *from a Bernoulli distribution of two different values with equal absolute value and probability convergence rate of ZO-SGD is*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_{t,\{z_n\}_{n=1}^N} \|\widehat{\nabla} f(\theta_t; \mathcal{B}_t)\|_2^2 = \mathcal{O}\left(\sqrt{\frac{d}{T}}\right). \tag{13}$$

Proof of Theorem 1. With Lemma 1, we can start from Lemma 3. By assuming that FO-SGD has finite variance bound as $\mathbb{E}_t\left[\left\|\widetilde{\nabla} f(x_t)\right\|_2^2\right] \leq G$ and reformulate Lemma 3 then we get :

$$\|\nabla f(x_t)\|_2^2 \leq \frac{1}{\eta}\mathbb{E}_{t,\{z_n\}}\left[f(x_t) - f(x_{t+1})\right] + \frac{Ld}{2N}\eta G.$$

Summing over from $t = 0$ to $t = T$:

$$\sum_{t=0}^{T-1} \|\nabla f(x_t)\|_2^2 \leq \frac{1}{\eta}\left[f(x_0) - \mathbb{E} f(x_T)\right] + \frac{Ld}{2N}\eta G T.$$

Remind that f is lower bounded with f_* and divide with T :

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(x_t)\|_2^2 \leq \frac{f(x_0) - f_*}{\eta T} + \frac{Ld}{2N}\eta G.$$

Let $\eta = \mathcal{O}\left(\sqrt{\frac{1}{dT}}\right)$ then,

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(x_t)\|_2^2 = \mathcal{O}\left(\sqrt{\frac{d}{T}}\right).$$

□

C Further analysis

We conducted additional supplementary experiments to further validate and gain deeper insights into our proposed method, Zip. To ensure a comprehensive analysis, we extended our evaluation to include all remaining standard classification tasks mentioned in Section 5 and 6. This extended evaluation provides a more detailed understanding of the performance of Zip across a diverse range of datasets.

C.1 Impact of optimization methods varying context token counts

We conduct a series of experiments to examine how varying the number of context tokens affects both first-order and zeroth-order optimization methods across multiple datasets, as illustrated in Figure 2 and 13.

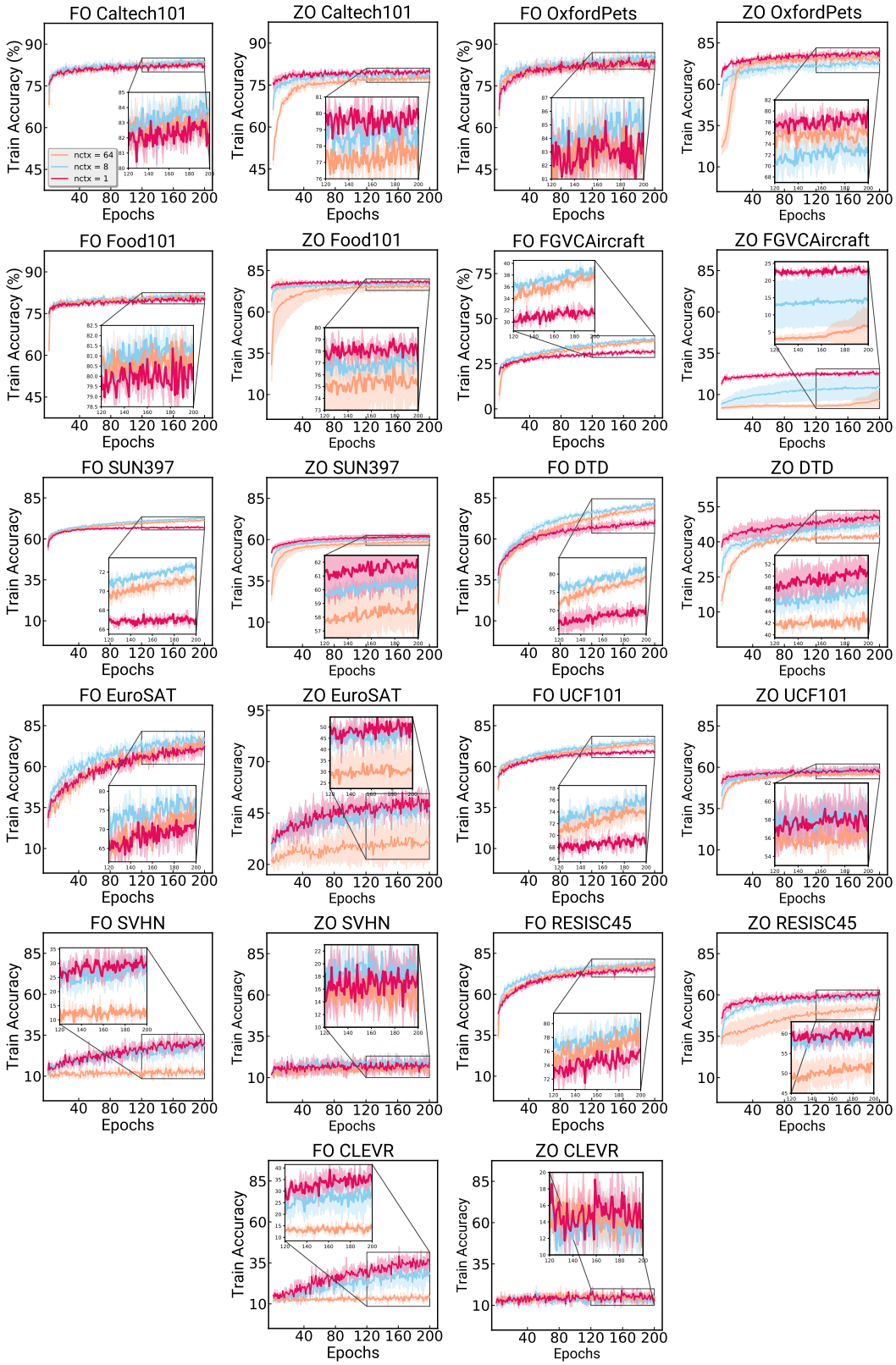


Figure 13: Effect of optimization methods across various vision-language tasks.

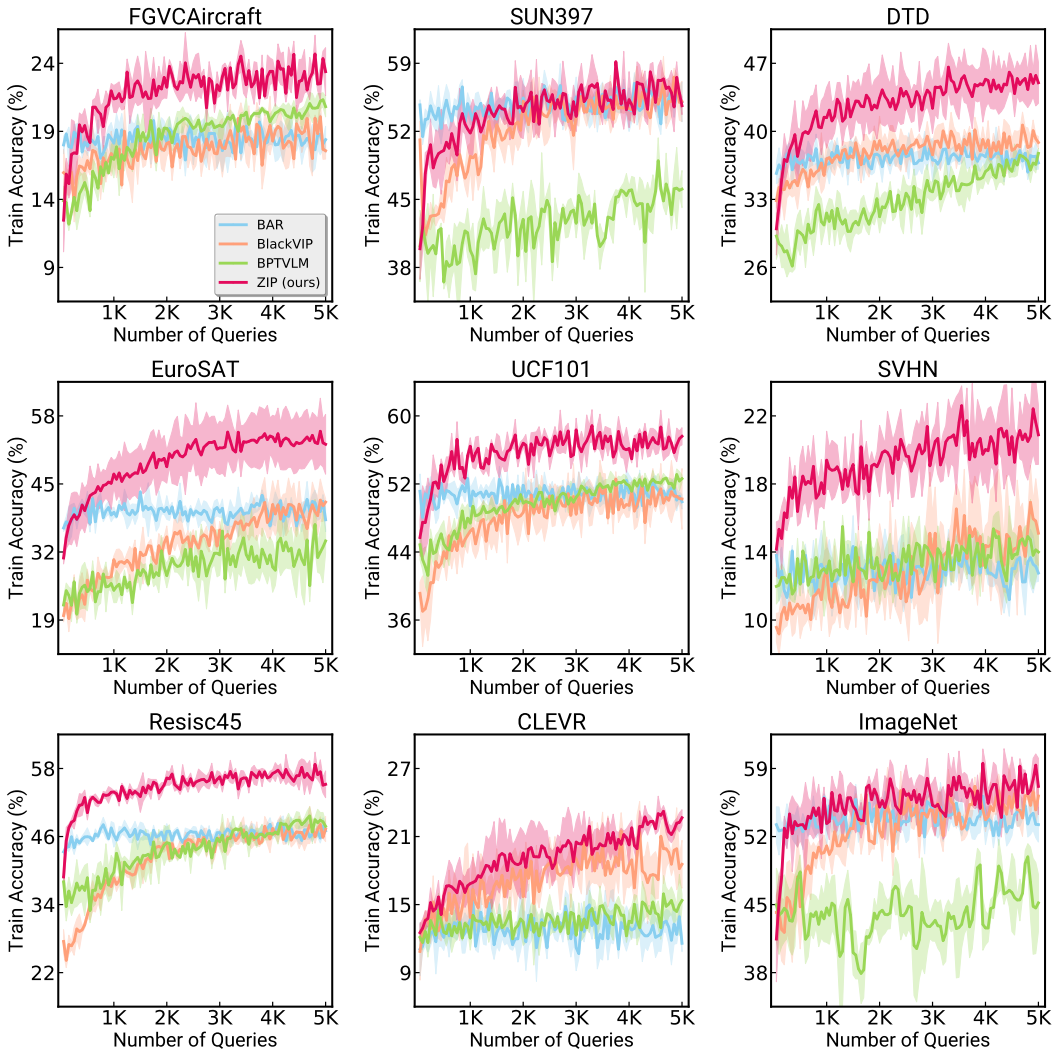


Figure 14: Training curves with 5,000 query budgets across various vision-language tasks.

Our findings reveal that zeroth-order optimization generally performs better with fewer context tokens (*e.g.*, 1 token). However, certain datasets such as UCF101, SVHN, and CLEVR deviate from this trend. In contrast, first-order optimization typically aligns with the trends shown in Section 3, displaying improved accuracy with a moderate number of context tokens across most datasets, except for SVHN and CLEVR, which demonstrate variations in optimal token counts.

These results suggest that the ideal number of context tokens can vary depending on the dataset, reinforcing our claim in Section 3 that first-order optimization generally benefits from a larger context token counts, whereas zeroth-order optimization tends to be more effective with fewer tokens.

C.2 Query Efficiency

Figure 4 and 14 display the training accuracy curves of ZIP under a 5,000 query budget across various tasks. Throughout the training process, ZIP consistently demonstrates faster training speeds and achieves higher accuracy compared to other BBPT methods across most datasets, highlighting its capability to utilize the available query budget more efficiently.

In Figure 5 and 15, we further analyze the number of API calls required to reach specific accuracy targets across various datasets. The target accuracy is determined as the minimum of the maximum accuracy achieved by all methods. The results indicate that ZIP consistently reaches these accuracy

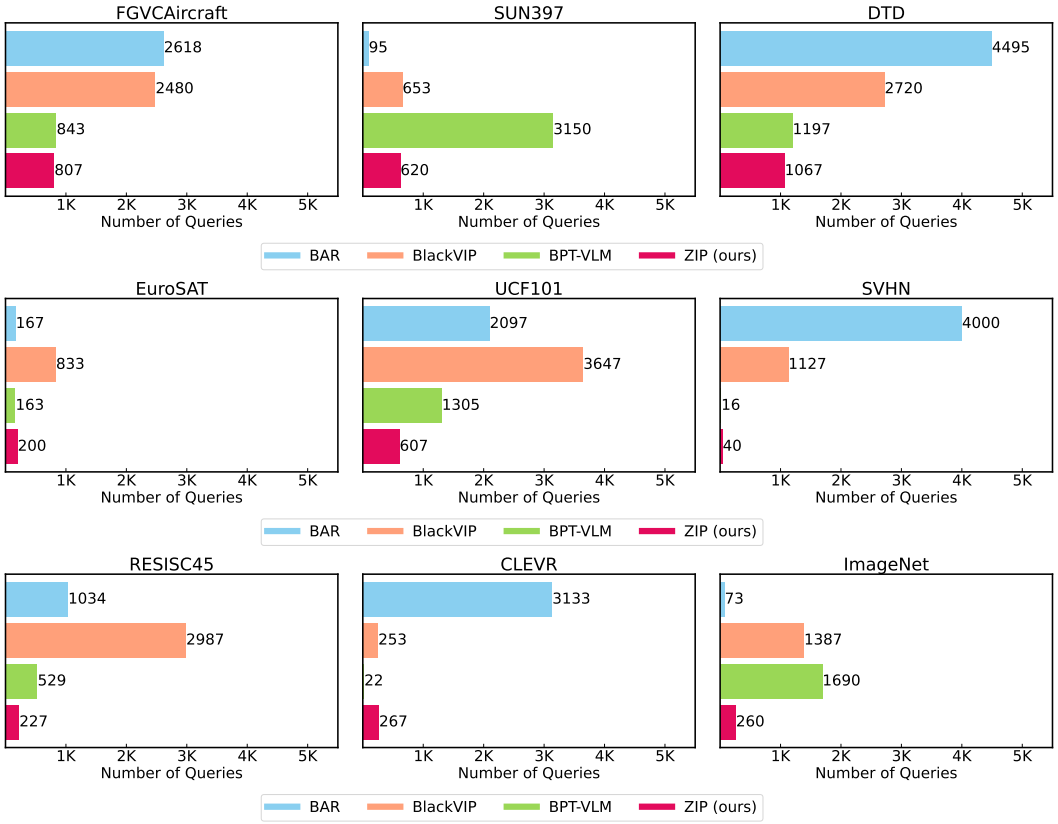


Figure 15: Queries to reach target accuracy across various vision-language tasks.

milestones with fewer queries than other methods, underscoring its query-efficient design and adaptability across a diverse range of tasks.

Additionally, in Figure 6 and 16, we compare the performance of first-order, zeroth-order optimization, and ZIP across multiple datasets. These results further validate our claim in Section 5.3 that ZIP effectively bridges the gap between first-order and zeroth-order optimization. ZIP not only consistently outperforms standard zeroth-order methods in test accuracy across all evaluated datasets but also frequently surpasses first-order optimization, demonstrating its outstanding training efficiency.

Moreover, we include results for context token $m = 1$ as a reference (See Figure 17), demonstrating that naive zeroth-order optimization with one token often struggles to match the performance of ZIP with 8 tokens, particularly in maintaining stable training accuracy. ZIP significantly outperforms the naive method on OxfordPets, FGVCAircraft, EuroSAT, and CLEVR. While the naive method shows comparable results on some other datasets, it is worth noting that even the first-order method with 8 tokens does not yield substantial improvements over the first-order method with 1 token on Caltech101, OxfordPets, and Food101 (See Figure 13). Additionally, using 1 token performs better on CLEVR and SVHN, highlighting that the optimal number of prompt tokens remains an important factor for performance.

These supplementary findings reinforce our assertions in Section 5.3, confirming that ZIP not only accelerates training but also makes highly efficient use of query budgets, making it exceptionally suited for resource-constrained scenarios.

C.3 Low-rank approximation with diagonal matrix

To further validate the effectiveness of our low-rank approximation with a diagonal matrix, introduced in Section 4.1, we conducted a comprehensive ablation study. This study compares the standard

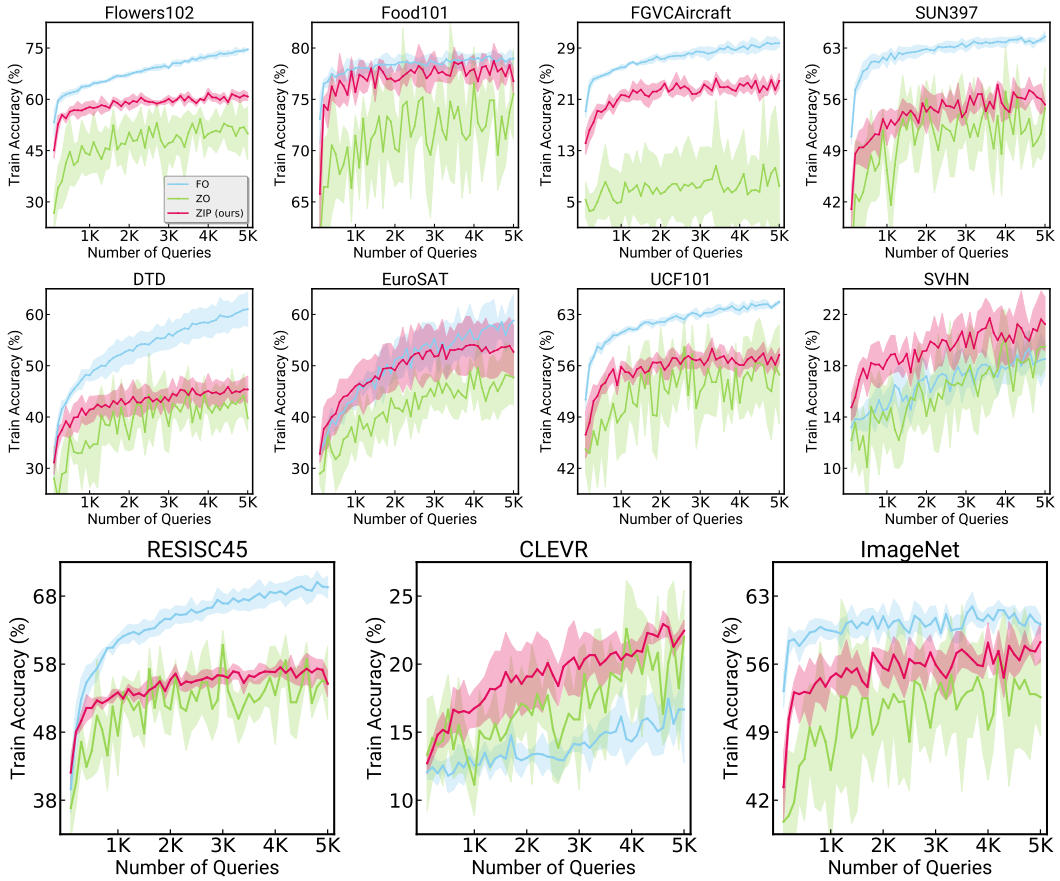


Figure 16: Training curves of first-order, zeroth-order and ZIP across various vision-language tasks.

Table 7: Benefits of low-rank approximation with diagonal matrix. Our method outperforms both standard dimensionality reduction and LoRA, showing significant improvements in test accuracy.

Method	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	SUN397	DTD	SVHN	EuroSAT	Resisc45	CLEVR	UCF101	ImageNet	**Average**
Standard	90.9	88.1	67.5	84.6	23.8	57.9	43.2	31.5	56.5	58.3	18.3	65.3	62.3	57.6
LoRA	90.7	89.3	**68.1**	85.0	23.7	57.4	43.9	36.0	59.2	57.0	**21.2**	65.2	62.6	58.4
Ours	**93.1**	**90.8**	67.1	**86.0**	**25.2**	**59.0**	**44.4**	**40.9**	**60.6**	**63.3**	20.2	**67.4**	**64.8**	**60.2**

dimensionality reduction technique with our proposed low-rank approximation, evaluated in two settings.

First, we fixed the intrinsic dimensionality at 500 for both the standard method and our approach. However, our method applies an additional low-rank approximation with a diagonal matrix, reducing the parameter size to 417. As shown in Figure 18, this results in improved training speed.

Next, to isolate the effects of the low-rank approximation, we set the parameter size to 417 for both methods, demonstrating that hyper-parameter size alone is not the key factor driving the efficiency gains. As illustrated in Figure 19, our low-rank approximation method retains core information while reducing parameters, significantly enhancing both training speed and performance.

Additionally, we compared our technique to the LoRA-style approximation (Hu et al., 2022). Our method, which introduces only r parameters in the diagonal matrix, effectively captures essential information from the parameter space, boosting the model's expressive power without significant parameter overhead. Table 7 presents the test accuracy comparison between our approach, the standard

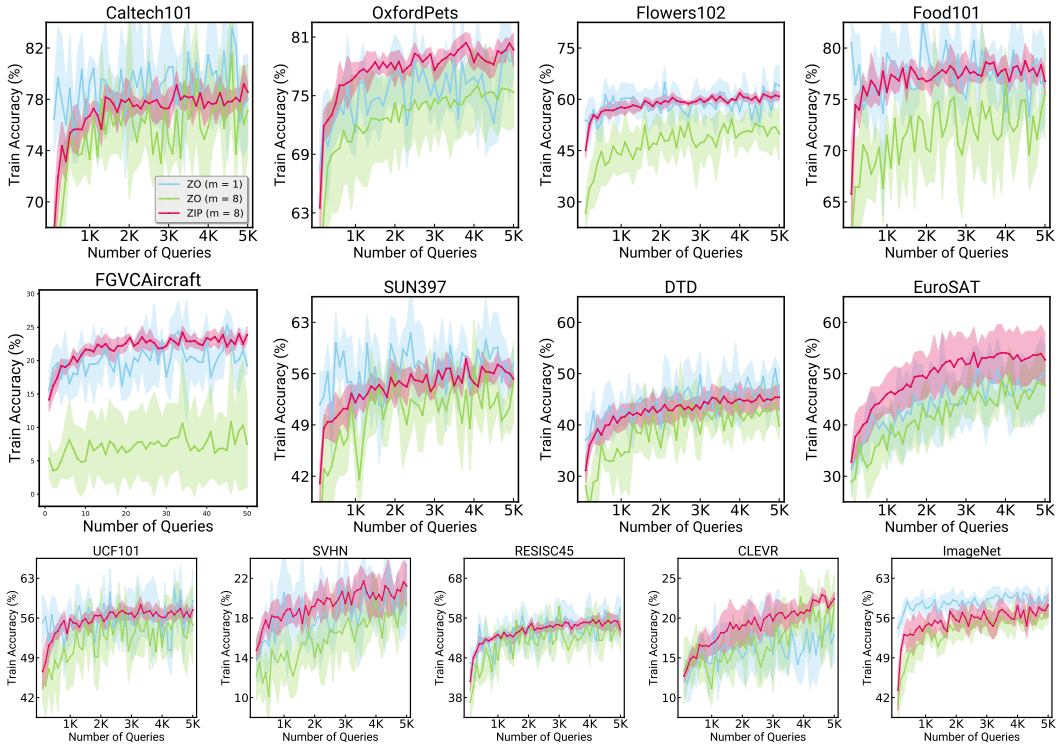


Figure 17: Training curves of zeroth-order ($m = 1$), zeroth-order ($m = 8$) and ZIP ($m = 8$) across various vision-language tasks.

dimensionality reduction method, and LoRA. Our method consistently outperforms both alternatives, demonstrating the clear advantage of integrating a diagonal matrix with low-rank approximation.

These findings highlight the effectiveness of our approach in preserving model expressiveness while optimizing parameter efficiency, making it a compelling solution for efficient model training.

C.4 Gradient clipping and optimal threshold

In Figure 15 and 21, we further investigate the impact of gradient clipping and the effect of varying the optimal clipping threshold across multiple datasets. The results indicate that applying gradient clipping consistently enhances training accuracy and reduces loss across most datasets, demonstrating its effectiveness in stabilizing the training process.

When evaluating test accuracy with varying gradient clipping thresholds, ZIP achieves near-optimal performance across the majority of datasets, consistently outperforming cases where no gradient clipping is applied. Although there are some exceptions, such as SVHN, DTD, and CLEVR, where gradient clipping does not yield significant improvements in test accuracy, the results remain comparable to ZIP without clipping, indicating that the technique does not hinder performance in these cases.

These findings substantiate that our gradient clipping approach significantly improves the overall performance of zeroth-order optimization, and the selected $\sqrt{\delta}$ threshold effectively serves as a reliable and practical choice for enhancing training efficiency.

D Experiment details

D.1 Algorithm

During the training process, our method, ZIP, initiates by calculating the low-rank approximation and integrating shared feature representations. These approximations are subsequently utilized to

Algorithm 1: The training process of ZIP.

Input: The training data $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}$, pre-trained CLIP model g, projection matrix $\{\mathbf{M}_i\}_{i=1}^m$, learnable parameters of each context token θ_i, gradient clipping threshold $\sqrt{\delta}$, context token counts m, number of gradient estimates N for N-SPSA, smoothing parameter c, batch size $\mathcal{B}$, and API call budget $\mathcal{T}$.

Function f (Ξ_t; X) **:**
- Calculate the original token parameters θ_t
- **for** i to m **do**
  - $\theta_{t,i} = \theta_{0,i} + \mathbf{M}_i \boldsymbol{w}_{t,i}$
- **end**
- Forward propagate through CLIP model with reconstructed tokens $\widetilde{g} = g(\theta_t; X)$
- **return** $\widetilde{g}$

Function N−SPSA (Ξ_t, c, N, X) **:**
- **for** n to N **do**
  - Sample $a \sim Uniform(0, 1)$, with ensuring a is not 0
  - Sample $z_n \sim Bernoulli(a : 0.5, -a : 0.5)$
  - Calculate the first loss $f(\Xi_t + cz_n; X)$
  - Calculate the second loss $f(\Xi_t - cz_n; X)$
  - Calculate the n-th gradient estimation $\widehat{\nabla} f_n(\Xi_t; X) = \frac{f(\Xi_t + cz_n; X) - f(\Xi_t - cz_n; X)}{2c}(z_n)^{-1}$.
- **end**
- Calculate N-SPSA gradient estimation $\widehat{\nabla} f(\Xi_t; X) = \frac{1}{N}\sum_{n=1}^{N} \widehat{\nabla} f_n(\Xi_t; X)$
- **return** $\widehat{\nabla} f(\Xi_t; X)$

Initialize $\Xi_0, \mathbf{U}_0, \boldsymbol{s}_0, \mathbf{V}_0$

for t to $\mathcal{T}/2N$ **do**
- **for** each training mini-batch X, Y **do**
  - Calculate the weight matrix $\Xi_t = [\boldsymbol{w}_{t,1}|\boldsymbol{w}_{t,2}|\cdots|\boldsymbol{w}_{t,q}] = \mathbf{U}_t \text{diag}(\boldsymbol{s}_t)\mathbf{V}_t^T + \boldsymbol{u}_t \otimes \mathbb{1}$
  - Calculate the gradient estimation $\widehat{\nabla} f(\Xi_t; X)$ using N−SPSA (Ξ_t, c, N, X)
  - Calculate the clipping coefficient $\alpha_t = \mathbf{min}(\frac{\sqrt{\delta}}{\sqrt{\sum_{i=1}^{\delta} \widehat{\nabla} f(\theta_t)_i^2}}, 1)$
  - Gradient descent using clipping $\Xi_{t+1} = \Xi_t - \eta_t \alpha_t \widehat{\nabla} f(\Xi_t)$
- **end**

end

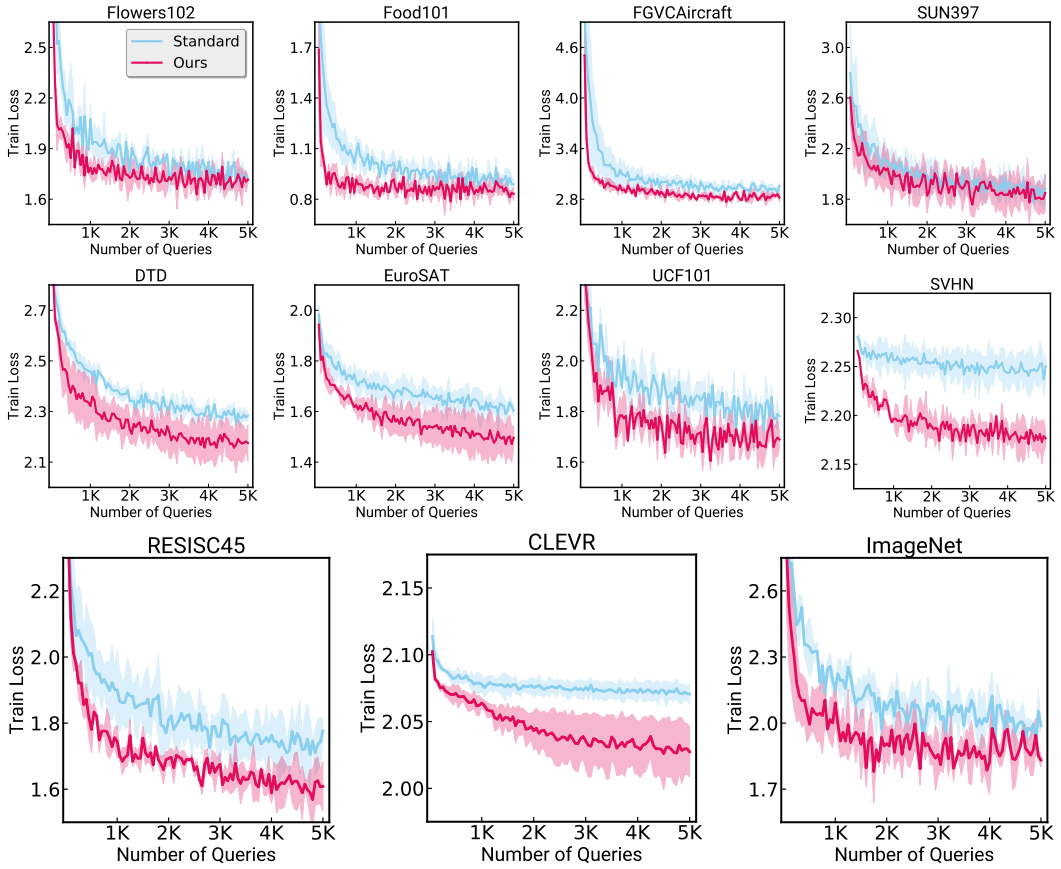


Figure 18: Effects of low-rank approximation with diagonal matrix across various vision-language tasks.

reconstruct the original parameter space through random projection, allowing ZIP to generate the prompt representations necessary for loss computation efficiently. To ensure clarity and provide a comprehensive understanding of the training procedure, the summarized training algorithm can be found in Algorithm 1, which outlines each stage of the process for easy reference.

D.2 DATASET DETAILS

In this study, we leverage a total of 13 general classification datasets and 4 out-of-distribution (OOD) datasets, widely used in prior research. These 13 classification tasks are employed to comprehensively evaluate ZIP's performance in general few-shot learning, base-to-new generalization, and cross-dataset transfer scenarios. Additionally, the 4 OOD datasets are used to rigorously assess ZIP's ability to handle out-of-distribution generalization. A detailed overview of each dataset, including task descriptions and evaluation metrics, is provided in Table 8.

D.3 HYPER-PARAMETERS

To achieve stable and accurate gradient approximations, zeroth-order optimization algorithms typically perform multiple gradient estimations, with the results being averaged to obtain a more reliable gradient estimate. Following the methodology outlined in Oh et al. (2023), we repeat this gradient estimation process five times for all zeroth-order-based baselines to ensure consistency and robustness. For SPSA methods, we tune key hyper-parameters, including the perturbation magnitude and decay factor. For evolutionary strategies, we adjust the population size, intrinsic dimensionality, and the number of visual and text tokens. The search ranges for these hyper-parameters are based on the recommendations provided by the authors of BAR (Tsai et al., 2020), BLACKVIP (Oh et al., 2023),

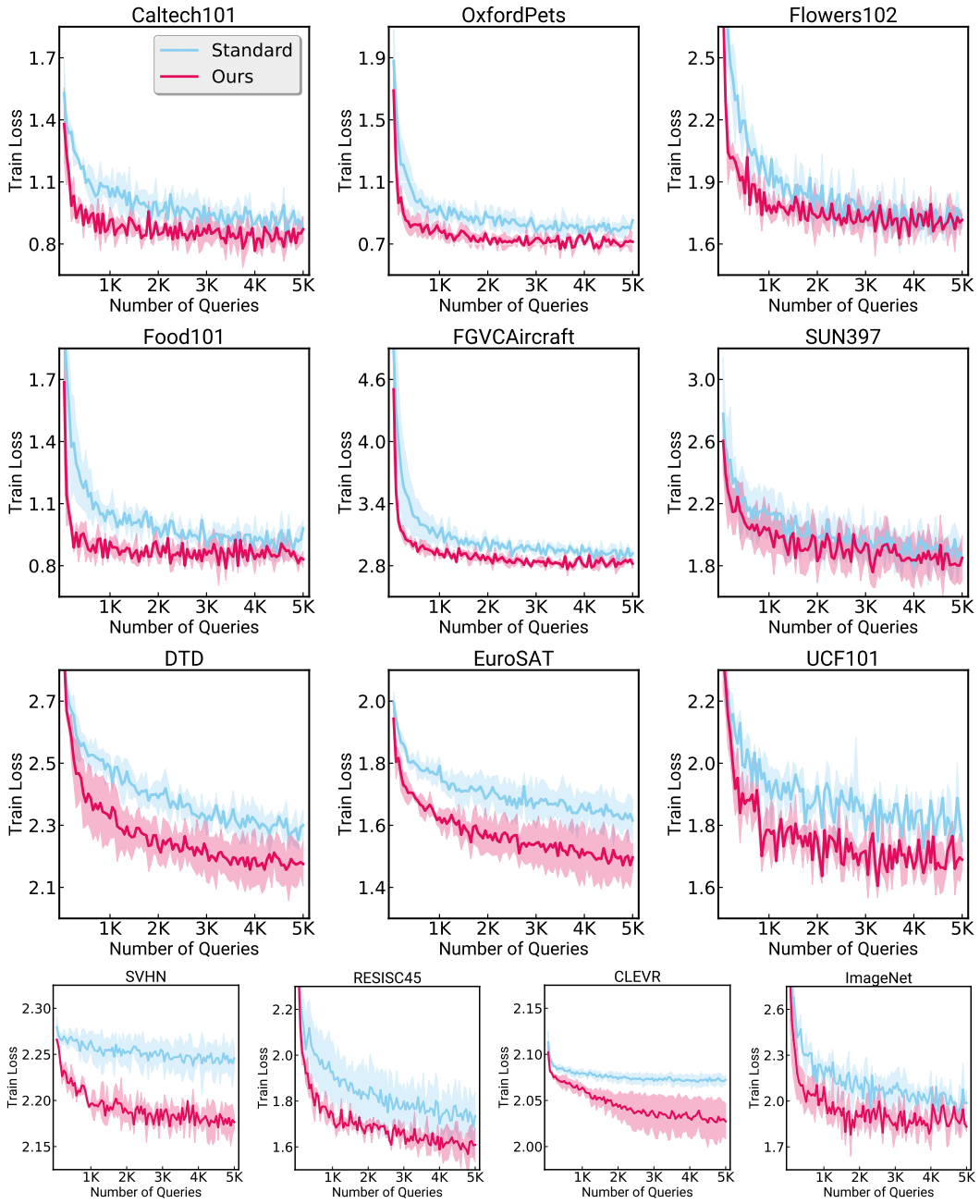


Figure 19: Effects of low-rank approximation with diagonal matrix at fixed parameter size (*i.e.*, 417) across various vision-language tasks.

and BPTVLM (Yu et al., 2023), and are summarized in Table 9. Regarding the learning objectives, cross-entropy loss is employed for BLACKVIP and BPTVLM, while focal loss is used for BAR. All BBPT experiments utilize a batch size of 128 across all datasets, ensuring consistent and comparable evaluation.

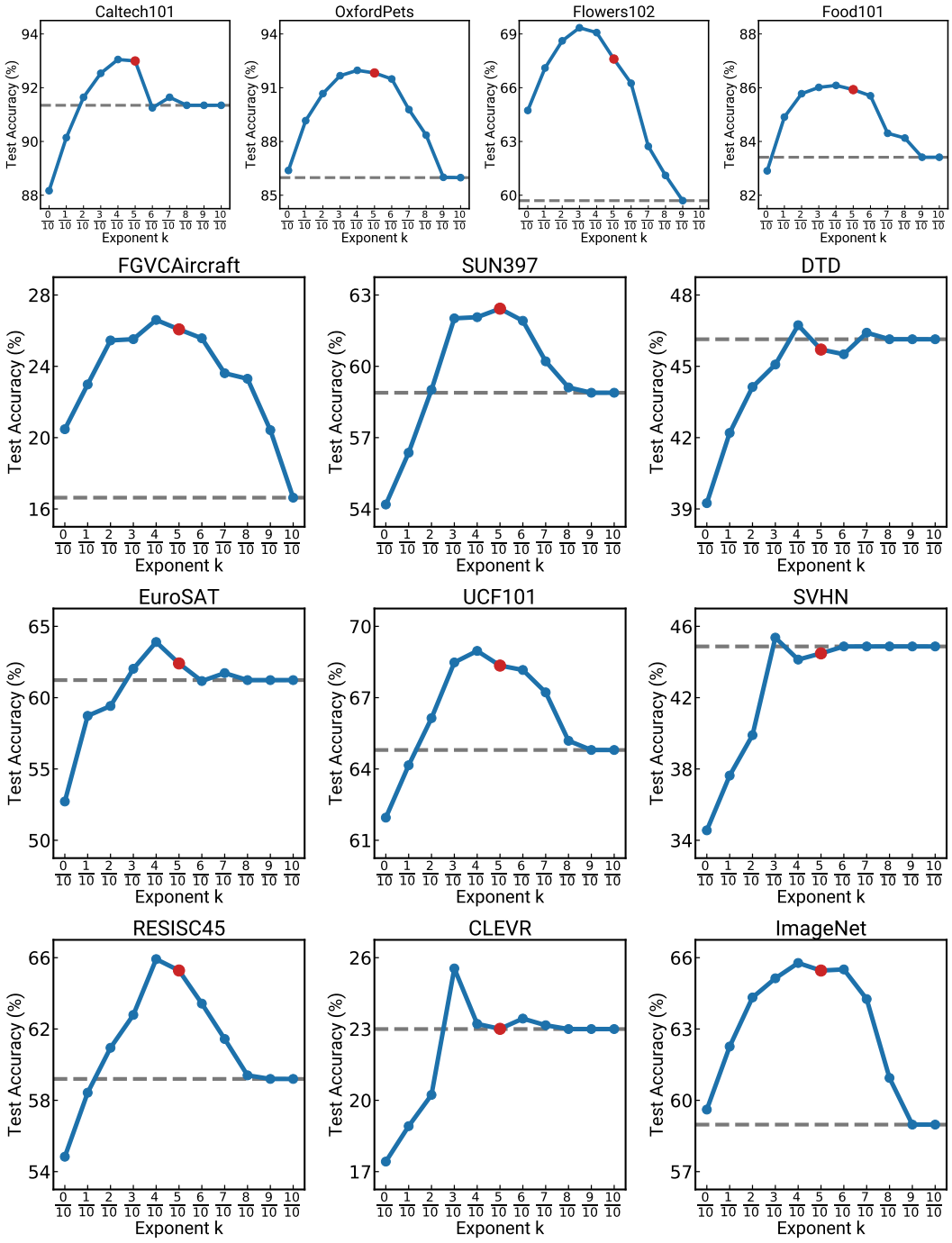


Figure 20: Effects of optimal threshold across various vision-language tasks.

D.4 Baseline Details

D.4.1 Zero-shot CLIP

CLIP (Radford et al., 2021) is a prominent vision-language foundation model widely employed across various tasks, such as classification, segmentation, and other vision-language applications. Trained on large-scale image-text datasets, CLIP has demonstrated exceptional effectiveness in numerous downstream tasks, thanks to its ability to leverage visual concepts learned from natural language supervision. It performs zero-shot classification using manually crafted prompt templates

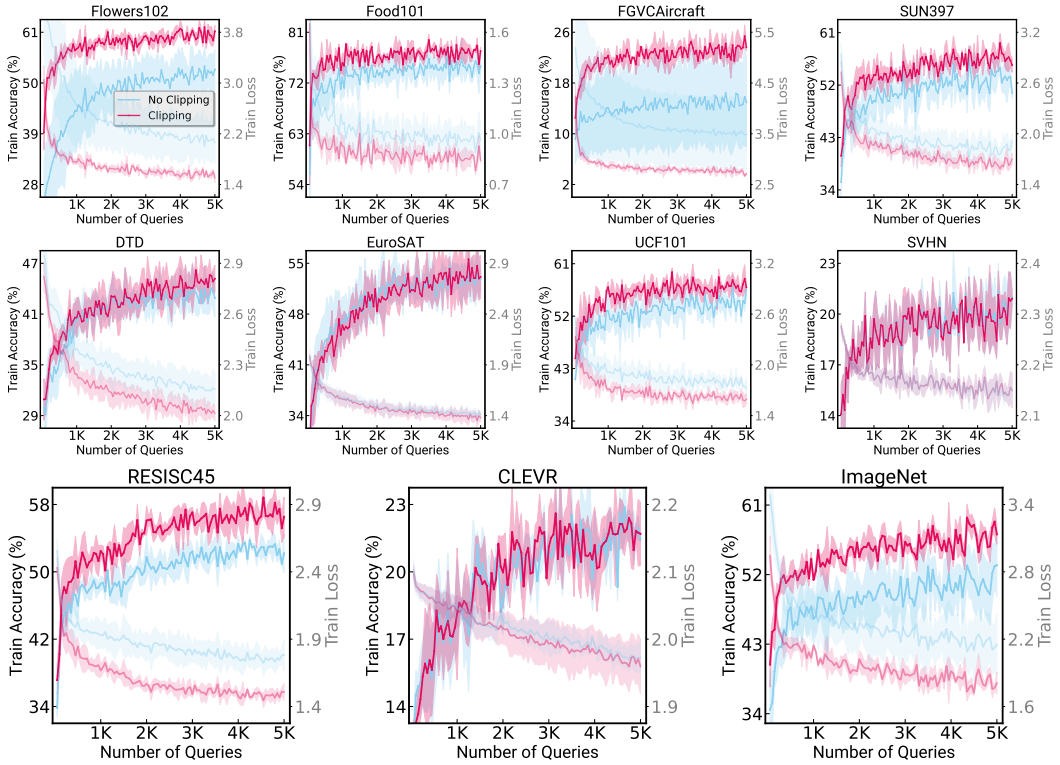


Figure 21: Effects of zeroth-order gradient clipping across various vision-language tasks.

Classification Tasks					
Dataset	**#Train**	**#Valid**	**#Test**	**Classification Type**	**Manual Prompt**
ImageNet	1.28M	N/A	50,000	Generic object	"a photo of a [CLASS]."
Caltech101	4,128	1,649	2,465	Generic object	"a photo of a [CLASS]."
OxfordPets	2,944	736	3,669	Fine-grained objects	"a photo of a [CLASS], a type of pet."
Flowers102	4,093	1,633	2,463	Fine-grained objects	"a photo of a [CLASS], a type of flower."
Food101	50,500	20,200	30,300	Fine-grained objects	"a photo of [CLASS], a type of food."
FGVCAircraft	3,334	3,333	3,333	Fine-grained objects	"a photo of a [CLASS], a type of aircraft."
SUN397	15,880	3,970	19,850	Scene	"a photo of a [CLASS]."
DTD	2,820	1,128	1,692	Text	"[CLASS] texture."
SVHN	73,257	26,032	26,032	Digit	"This is a photo of a [CLASS]."
EuroSAT	13,500	5,400	8,100	Satellite	"a centered satellite photo of a [CLASS]."
Resisc45	6,300	2,520	7,560	Scene	"This is a photo of a [CLASS]."
CLEVR	70,000	15,000	15,000	Diagnosis	"This is a photo of [CLASS] objects."
UCF101	7,639	1,898	3,783	Action	"a photo of a person doing [CLASS]."
ImageNetV2	N/A	N/A	10,000	Generic object	"a photo of a [CLASS]."
ImageNet-Sketch	N/A	N/A	50,889	Sketch image	"a photo of a [CLASS]."
ImageNet-A	N/A	N/A	7,500	Adversarially filtered image	"a photo of a [CLASS]."
ImageNet-R	N/A	N/A	30,000	Cartoon, Sculptures, Paintings	"a photo of a [CLASS]."

Table 8: The datasets used in this study, along with the corresponding manual prompts. Samples are drawn exclusively from the original training set to ensure consistency with baseline data.

(*e.g.*, "a photo of a CLASS"). Due to its versatility and strong performance, CLIP serves as the backbone for many black-box prompt tuning models, including our proposed method, ZIP.

D.4.2 BAR

Originally developed for transferring knowledge from an ImageNet pre-trained model to the medical domain, BAR (Tsai et al., 2020) reprograms pre-trained models using a frame-shaped, learnable program that embeds the target task image within this frame and optimizes it via zeroth-order

Hyper-parameter	Assignment	Algorithm
initial LR	{40.0, 20.0, 10.0, 5.0, 1.0}	BAR
initial LR (a_1)	{1.0, 0.1, 0.01, 0.005}	BLACKVIP, ZIP
min LR	{0.1, 0.01, 0.001}	BAR
decaying step	{0.9, 0.5, 0.1}	BAR
LR decaying factor	{0.6, 0.5, 0.4, 0.3}	BLACKVIP, ZIP
initial PM (c_1)	{0.01, 0.005, 0.001}	BLACKVIP, ZIP
PM decaying factor	{0.2, 0.1}	BLACKVIP, ZIP
std. of perturbation	{1.0, 0.5}	BAR
smoothing	{0.1, 0.01, 0.001}	BAR
gradient smoothing	{0.9, 0.7, 0.5, 0.3}	BLACKVIP
population size	{5, 10, 15, 20}	BPTVLM
intrinsic dimensionality	{500, 1000, 2000}	BPTVLM, ZIP
rank	{1, 3, 5}	ZIP
visual tokens	{5, 10}	BPTVLM
text tokens	{5, 10}	BPTVLM

Table 9: Hyper-parameter search range for BBPT approaches.

algorithms. The size of this learnable program is adjusted based on the input image resolution. For example, in the original study, when the resolution of the downstream image was larger than that of the pre-trained model, an embedded target image size of 64×64 was used within a 299×299 learnable program. In contrast, BLACKVIP (Oh et al., 2023) modified this approach by designing an embedded image resolution of 194×194 to avoid performance degradation caused by the heavy-padding of thin images within the prompt. In this paper, we adopt the settings established by BLACKVIP (Oh et al., 2023) when optimizing BAR, ensuring consistency and addressing the limitations of the original design.

D.4.3 BLACKVIP

BLACKVIP (Oh et al., 2023) generates input-conditional visual prompts for each image via a projection network, allowing prompts to adapt dynamically to the specific features of each input. For the optimization process, BLACKVIP employs Simultaneous Perturbation Stochastic Approximation with Gradient Correction (SPSA-GC), which integrates Nesterov Accelerated Gradients (NAG)(Nesterov, 1983), enhancing the efficiency of zeroth-order training. Unlike other methods such as CoCoOp (Zhou et al., 2022a), which optimize additional input-attached parameters, BLACKVIP focuses exclusively on the projection network, effectively creating adaptive, input-conditioned visual prompts for BBPT tasks. While this design choice makes BLACKVIP highly adaptable and well-suited for black-box settings, the large number of parameters introduced by the projection networks can negatively impact training efficiency, posing a challenge in resource-constrained environments.

D.4.4 BPTVLM

BPTVLM (Yu et al., 2023) utilizes evolutionary strategies for BBPT, distinguishing itself from previous approaches. In this method, BPTVLM introduces learnable parameters into both text and image prompts, enabling a more comprehensive adaptation to various tasks. To enhance efficiency, BPTVLM incorporates the concept of intrinsic dimensionality, reducing the overall number of learnable parameters by applying a random projection matrix to both text and image prompts. This approach effectively balances adaptability and parameter efficiency, making BPTVLM a more versatile option for BBPT scenarios.