This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Prime Once, then Reprogram Locally: An Efficient Alternative to Black-Box Service Model Adaptation

Yunbei Zhang[1] Chengyi Cai[2] Feng Liu[2] Jihun Hamm[1]
[1]Tulane University [2]University of Melbourne

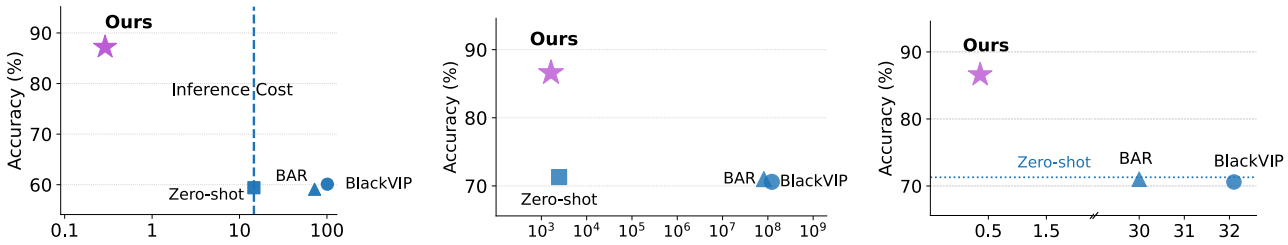
Abstract

Adapting closed-box service models (i.e., APIs) for target tasks typically relies on reprogramming via Zeroth-Order Optimization (ZOO). However, this standard strategy is known for extensive, costly API calls and often suffers from slow, unstable optimization. Furthermore, we observe that this paradigm faces new challenges with modern APIs (e.g., GPT-4o). These models can be less sensitive to the input perturbations ZOO relies on, thereby hindering performance gains. To address these limitations, we propose an Alternative efficient Reprogramming approach for Service models (AReS). Instead of direct, continuous closed-box optimization, AReS initiates a single-pass interaction with the service API to prime an amenable local pre-trained encoder. This priming stage trains only a lightweight layer on top of the local encoder, making it highly receptive to the subsequent glass-box (white-box) reprogramming stage performed directly on the local model. Consequently, all subsequent adaptation and inference rely solely on this local proxy, eliminating all further API costs. Experiments demonstrate AReS's effectiveness where prior ZOO-based methods struggle: on GPT-4o, AReS achieves a +27.8% gain over the zero-shot baseline, a task where ZOO-based methods provide little to no improvement. Broadly, across ten diverse datasets, AReS outperforms state-of-the-art methods (+2.5% for VLMs, +15.6% for standard VMs) while reducing API calls by over 99.99%. AReS thus provides a robust and practical solution for adapting modern closed-box models. Code: https://github.com/yunbeizhang/AReS.

1. Introduction

In recent years, Model-as-a-Service (MaaS) has emerged as a dominant paradigm for deploying and accessing state-of-the-art (SOTA) machine learning models [1, 5, 48, 56]. Companies and research institutions often release their large pre-trained models (which we refer to as *service models* in this work) as inference APIs, allowing users to query these powerful models without direct access to model parameters or architecture details. This closed-box nature creates a significant challenge for transfer learning [54, 55, 63]. Unlike some approaches that may assume access to intermediate features or token embeddings [42, 64, 79] (see Table 1 for a detailed comparison), this work focuses on the most restrictive closed-box setting, where only raw input-prediction access to the service model is available. In this challenging scenario, adapting models to downstream tasks, which typically requires gradient information, becomes particularly difficult. Model Reprogramming (MR) [7, 11, 25, 43, 57], known as Visual Reprogramming (VR) or Visual Prompting (VP) in vision domains, has been developed as one solution, enabling adaptation of MaaS models through input-level modifications optimized using zeroth-order optimization (ZOO) [31, 51, 52, 60] based solely on model outputs.

While closed-box visual reprogramming approaches such as BAR [57] and BlackVIP [41] have demonstrated promising results under this strict input-prediction only setting, they entail practical challenges. The optimization process, reliant on approximate gradients, can be slow and unstable, and the computational and practical costs could be prohibitive [32, 77]. Such methods require numerous API calls not only during the lengthy training phase but also for every inference instance, leading to significant financial burdens and requiring uninterrupted connectivity to the service model [41, 57, 64]. More critically, our experiments indicate that this ZOO-based paradigm faces new challenges with modern, robust APIs. We find that powerful models like GPT-4o can be less sensitive to the noisy input perturbations central to ZOO, yielding little to no performance gains (Fig. 1a and Table 4), while on models like LLaVA [30], these perturbations may affect delicate vision-language alignment. Furthermore, the substantial resources expended for these SOTA methods do not always translate into commensurate performance enhancements (Fig. 1b and 1c), raising concerns about their efficiency-to-effectiveness ratio [41, 73]. These substantial hurdles in efficiency, cost, and now, effectiveness on modern models, form the core motivation for our work. It compels us to ask: *Are there alternative approaches that can offer advantages beyond conventional closed-box model reprogramming, particularly in overcoming these practical and economic limitations?*



(a) **GPT-4o Cost (USD)** (b) **#API Calls** (c) **Training Time (hour)**

Figure 1. **(a)** On the real-world GPT-4o API, ZOO-based methods show limited effectiveness, providing little to no improvement over zero-shot performance while incurring high total (training and inference) costs. **(b, c)** On CLIP ViT-B/16, these methods require $\sim 10^8$ API calls and over 32 hours of training on Flowers102, yet still underperform ours, AReS, which uses only $\sim 10^3$ API calls and 0.4 hours.

We propose an **A**lternative efficient **Re**programming approach for **S**ervice models (**AReS**) that directly addresses these challenges (see Fig. 2). Instead of direct, continuous closed-box optimization, AReS initiates a single-pass interaction with the service API to prepare an amenable local pre-trained encoder (e.g., a publicly available one, making the same initial assumption as methods like BlackVIP [41] but utilizing it for a fundamentally different purpose). This priming stage trains only a lightweight layer on top of the local encoder. Crucially, this step does not require a shared label space; it successfully prepares the local model for reprogramming even when the service API's outputs are in a disjoint label space (e.g., ImageNet) from the target task (e.g., Flowers). This approach, supported by a theoretical analysis connecting priming faithfulness to downstream performance, enables effective glass-box visual reprogramming to occur entirely and efficiently on the local model. Consequently, all subsequent adaptation and inference rely solely on this local model, eliminating further API costs. For example, on the Flowers102 dataset (as detailed in Fig. 1b and 1c), our approach achieves 86.6% accuracy, outperforming BlackVIP (70.6%), while dramatically reducing API calls from $\sim 10^8$ down to $\sim 10^3$ (>**99.99**% reduction) and slashing computation time from over 30 hours to less than half an hour (>**98.88**% reduction). Crucially, on GPT-4o, AReS improves by $+27.8\%$, while BlackVIP gains only $+0.7\%$. This is because the service model's robustness to input perturbations hinders ZOO-based methods, a limitation AReS bypasses by avoiding direct reprogramming.

We evaluate AReS across a broad range of service models. Our primary evaluation spans standard Vision Models (VMs) (e.g., ViT-B/16 [13], ResNet101 [16]) and Vision-Language Models (VLMs) (e.g., CLIP [48]) over ten diverse datasets. We also conduct evaluations on modern models, including the Multimodal Large Language Models (MLLMs) LLaVA [30], the proprietary MLLM API GPT-4o [1], and the commercial VM API Clarifai. Through these comprehensive experiments, we demonstrate that our approach consistently outperforms SOTA ZOO-based methods while dramatically reducing computational requirements and associated costs.

These findings have important implications for the broader field of transfer learning in increasingly API-centric machine learning ecosystems, suggesting a more effective path forward that balances performance, efficiency, and practicality. Our main contributions are summarized as follows:

1. We identify a key challenge for ZOO-based reprogramming on modern, robust APIs and present an effective alternative that shifts adaptation to a local model after a minimal, single-pass interaction with the service API.
2. We introduce **AReS**, a simple, theoretically-supported framework that combines a priming step with efficient glass-box reprogramming, uniquely enabling *cost-free* inference without further API dependency.
3. Through extensive experiments, AReS achieves superior performance, including significant gains on real-world APIs (e.g., GPT-4o and Clarifai) where prior methods fail, alongside average accuracy improvements of $+2.5\%$ for VLM and $+15.6\%$ for VM over BlackVIP, while reducing API calls by $> 99.99\%$.

2. Related Work

Closed-box Tuning for Model-as-a-Service (MaaS). Closed-box tuning for MaaS has been explored across various modalities, with approaches for Large Language Models (LLMs) [54, 55] and methods for VLMs that modify inputs at the token level while requiring access to text embeddings, placing them outside the strictest closed-box definition [15, 42, 64, 73], as detailed in Table 1. In the vision domain, approaches like BlackVIP [41] and BAR [57] employ ZOO to reprogram models by modifying visual inputs. While applicable to both VLMs and VMs, these ZOO-based approaches are known for optimization instability and high API call costs during training and inference [32, 41, 43, 57, 77]. More critically, their reliance on input perturbations may face new challenges in the era of modern APIs, e.g., powerful service models like GPT-4o can be robust enough to simply ignore these perturbations, rendering the costly ZOO adaptation process ineffective (Fig. 1a). These limitations highlight the need for an alternative strategy that avoids direct closed-box optimiza-

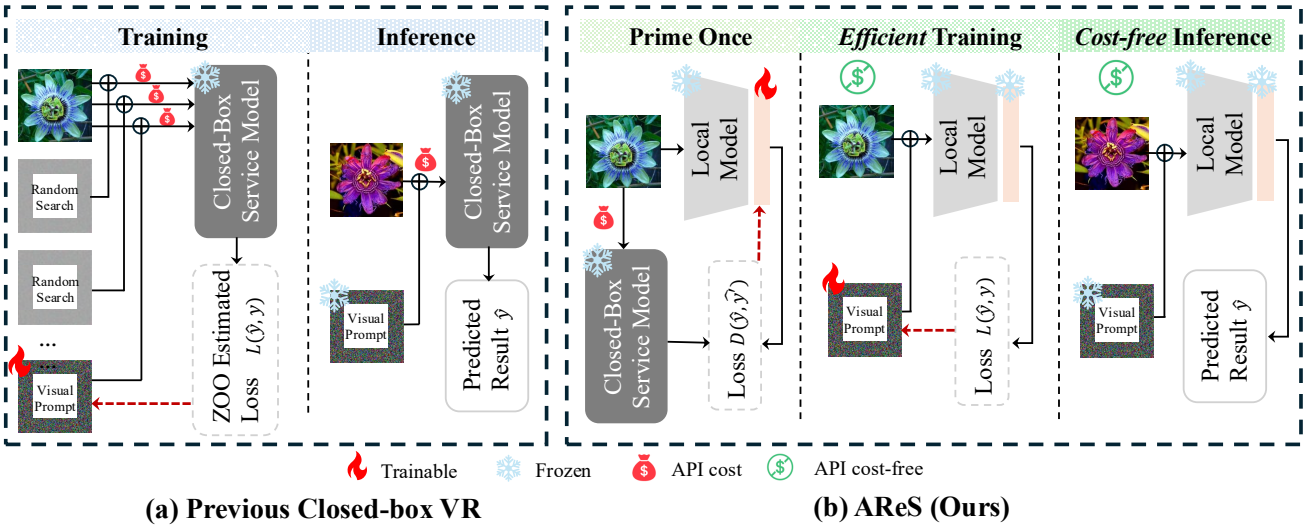


Figure 2. **(a)** Previous closed-box methods use Zeroth-Order Optimization (ZOO), which estimates gradients by querying the API multiple times for the same training image with random perturbations. This results in continuous API dependency: numerous calls during training and one API call per image during inference. **(b)** Our AReS approach performs a single-pass priming, requiring only one API call per training image to prepare a local model. This enables efficient, gradient-aware VR locally, eliminating all subsequent API costs during inference.

tion and instead enables efficient and effective adaptation for downstream tasks without repeated service-model queries.

Model Reprogramming (MR). MR [3, 11, 25, 76], termed Visual Reprogramming (VR) or Visual Prompting (VP) in vision domains, encompasses two main strategies. The first, token-based prompting, is tailored for transformer architectures like Vision Transformers (ViT) [3, 25, 71, 74, 84, 85]. These methods inject learnable tokens and require access to the model's embedding space or intermediate layers, rendering them unsuitable for strict closed-box settings or non-transformers [15, 42, 64, 73, 78]. The second category, input-level prompting [6, 8, 41, 57, 58, 80], directly modifies input images, offering model-agnostic flexibility with only input-output access. Both approaches can be optimized using gradient-based methods (glass-box) or ZOO techniques (closed-box) [41, 57, 64]. However, ZOO-based methods can be associated with efficiency and stability trade-offs. Their reliance on high query volumes and continuous API connectivity may be less suited for practical scenarios requiring rapid adaptation or offline use, such as in remote field operations. These practical considerations particularly motivate our work in exploring an alternative strategy.

3. Problem Setting

Closed-box Model Reprogramming. MR adapts pre-trained models to solve new tasks without modifying their parameters. Given a pre-trained model $\mathcal{F}_S : \mathcal{X}^S \mapsto \mathbb{R}^{K^S}$ where K^S is the number of source classes, the key concept is to introduce a learnable prompt $\mathbf{P}$ that transforms inputs from the downstream task domain $\mathcal{X}^T$ to the source domain $\mathcal{X}^S$ through an input transformation function $g_{\text{in}}(\cdot|\mathbf{P})$. Additionally, a label mapping function $g_{\text{out}} : \mathbb{R}^{K^S} \mapsto \mathbb{R}^{K^T}$ is employed to bridge the source (with K^S classes) and target (with K^T classes) label spaces. If the source and downstream tasks have the same label space, g_{out} becomes an identity mapping (e.g., in VLMs). Otherwise, for VMs, a gradient-free label mapping is established following existing works [6, 8, 57] (Detailed in Appendix 7.2). Importantly, only the input transformation g_{in} contains trainable parameters updated through gradient descent, while the label mapping g_{out} is computed and updated during training but does not introduce additional trainable parameters.

In MaaS settings, powerful pre-trained models are accessible only through APIs without access to their parameters or internal representations. We denote such a closed-box Service API model as $\mathcal{F}_S : \mathcal{X}^S \mapsto \mathbb{R}^{K^S}$, which accepts inputs from domain $\mathcal{X}^S$ and produces output predictions. Adhering to the strictest closed-box assumption established in prior work [57], we consider the scenario where the API returns only prediction probabilities rather than logits or embeddings, representing the most constrained access scenario (see Table 1). When adapting MaaS models to downstream tasks, the central challenge is to leverage these powerful models in a computationally and economically efficient manner. This requires developing methods that can optimize task-specific visual prompts using only the limited input-output responses from the API while minimizing computational costs and API usage during both training and inference.

Existing BMR Approaches. Existing ZOO-based BMR methods like BAR [57] employs random gradient-free (RGF) optimization [31, 60] to estimate the gradient of the loss function ℓ with respect to prompt parameters $\mathbf{P}$: $\nabla\ell(\mathbf{P}) \approx \frac{d}{q}\sum_{i=1}^{q}\frac{\ell(\mathbf{P}+\mu u_i)-\ell(\mathbf{P})}{\mu}u_i$, where d is the pa-

Table 1. Comparison of tuning methods for Model-as-a-Service (MaaS). The Access column indicates the strictness, distinguishing translucent-box (grey, requiring intermediate Features or Embedding access) from closed-box (black, Input/Predictions only) settings. We compare methods by: MR (Model Reprogramming-based); required Input Access and Output Access; support for VMs (Standard Vision Models) and VLMs (Vision-Language Models); Prompt type (L: Language, V: Visual); and Encoder Flexibility. Our method, AReS, uniquely offers Cost-free Inference for both VMs and VLMs while operating under the strictest closed-box constraints.

Access	Method	MR	Input Access	Output Access	VM	VLM	Prompt	Encoder Flexibility	Cost-free Inference
	LFA [42]	✗	Input	Features	✗	✓	–	✓	✗
	CBBT [15]	✗	Embedding	Features	✗	✓	L	✓	✗
	LaViP [29]	✓	Input	Features	✓	✓	V	✓	✗
	BPT-VLM [73]	✗	Embedding	Predictions	✗	✓	L&V	ViT Only	✗
	CraFT [64]	✗	Embedding	Predictions	✗	✓	L	✓	✗
	BAR [57]	✓	Input	Predictions	✓	✓	V	✓	✗
	BlackVIP [41]	✓	Input	Predictions	✓	✓	V	✓	✗
	LLM-Opt [33]	✗	Input	Predictions	✗	✓	L	✓	✗
	AReS (Ours)	✓	Input	Predictions	✓	✓	V	✓	✓

rameter dimension, q is the number of random directions, μ is a small constant, and u_i are random unit vectors. BlackVIP [41] introduces the Simultaneous Perturbation Stochastic Approximation with Gradient Correction (SPSA-GC) [51, 52] to improve gradient estimation: $\nabla\ell(\mathbf{P}) \approx \frac{\ell(\mathbf{P}+c\Delta)-\ell(\mathbf{P}-c\Delta)}{2c}\Delta^{-1}$, where c is a small constant and Δ is a random perturbation vector. BlackVIP also incorporates a Coordinator network for input-dependent prompt generation using a local pre-trained encoder (e.g., ViT-B/16). While these methods have shown promising results, they require multiple API calls per update and during inference, leading to increased computational and financial costs. Additionally, their zeroth-order gradient approximation can result in less stable optimization compared to exact gradients (i.e., first-order optimization) [32, 41, 77]. The challenge remains to develop more efficient approaches that can better leverage available local encoders while minimizing API usage.

4. Method

Prime Once. In the first stage, we utilize a locally available encoder $\mathcal{F}_L$ (the same assumption as BlackVIP [41]) and prime it using the closed-box service model $\mathcal{F}_S$. For each image x_i from the target domain dataset $\mathcal{D}^T = \{x_i\}_{i=1}^N$, we query the service model to obtain prediction probabilities $p_S(x_i) = \mathcal{F}_S(x_i)$. It is important to note that this is the *only time* we interact with the API, and after this single round of querying, the API is *no longer needed* for the rest of the process. We use the local encoder $\mathcal{F}_L$ (outputting d_{enc}-dim features) as a fixed feature extractor and add a trainable linear layer with parameters $\theta \in \mathbb{R}^{K^S \times (d_{\text{enc}}+1)}$ on top of it. Note that we abuse the notation $\mathcal{F}_L$ to refer to both the local encoder alone and the local encoder combined with the learned linear layer, depending on the context. Only the linear layer parameters are updated during priming:

$$\theta^* = \arg\min_\theta \sum_{i=1}^N \mathcal{L}_{\text{P}}(\mathcal{F}_L(x_i;\theta), p_S(x_i)) \,. \tag{1}$$

Our priming objective, inspired by the loss function from knowledge distillation [18, 53], is defined as: $\mathcal{L}_{\text{P}}(p_L, p_S) := -\sum_{j=1}^{K^S} p_{S,j} \log p_{L,j}$, where p_S/p_L are the output probabilities of the same image from the service/local models, respectively. This loss is equivalent to the Kullback-Leibler divergence $\text{KL}(p_S||p_L)$.

For both VMs and VLMs, we freeze the local encoder and only train the added linear layer. The key difference between VM and VLM adaptation is in how the outputs are handled: for VMs, the priming occurs in the *service model label probability space*, which presents a non-trivial challenge as this fixed vocabulary does not align with the downstream task's labels. For VLMs, the outputs are naturally aligned with the downstream task label space through the text encoder. This approach establishes a fair comparison with BlackVIP [41], as both methods only require a local encoder. Moreover, this opens up the potential to utilize extra **unlabeled** data from the downstream distribution, a significant advantage unavailable to other methods. (see Fig. 3d).

Remark (Priming vs. Knowledge Distillation)**.** Our priming stage is mechanistically similar to knowledge distillation [2, 9, 18, 22, 24, 46] but differs fundamentally in its objective and applicability. Whereas traditional distillation aims to create a final, high-performance student model—typically requiring a shared label space—our priming is solely a preparatory step to make the local model more amenable to subsequent reprogramming. This preparatory focus allows priming to operate even when the service and downstream label spaces are disjoint (e.g., using Flowers data in the ImageNet label space), providing crucial flexibility for closed-box model adaptation.

Efficient VR Training. After the priming, we proceed to the second stage of AReS: learning an effective visual prompt for the downstream task using gradient-based optimization. Unlike closed-box approaches that rely on approximate gradients, we can leverage exact gradient information since our local encoder with the learned linear layer is fully accessible. We define a learnable visual prompt $\mathbf{P}$ and an input transformation function g_{in} similar to closed-box methods, along with a label mapping function g_{out} (and becomes an

Table 2. Accuracy and Efficiency comparison on ten diverse visual recognition datasets using CLIP ViT-B/16 (Service) and ViT-B/16 (Local) in a 16-shot setting. AReS achieves superior average accuracy with a drastic reduction in resources (99.99% fewer API calls and 98% less training time). For this and all subsequent tables, Gray denotes glass-box Service VR results. #API (M): total API calls in millions (training and inference); Wall-clock Time (h): total training hours run on an RTX 6000 Ada GPU.

Method	Flowers	DTD	UCF	Food	GTSRB	EuroSAT	Pets	Cars	SUN	SVHN	Avg.	#API (M)	Time (h)
VR (glass-box)	86.8	62.0	74.0	81.6	65.1	90.9	90.7	66.2	66.7	60.2	74.4	16.82	9.8
Zero-shot	71.3	43.9	66.9	**85.9**	21.0	47.9	**89.1**	65.2	62.6	17.9	57.2	0.12	0
BAR	71.0	46.8	64.2	84.4	21.5	77.3	88.4	63.0	62.4	34.6	61.4	612.84	185.6
BlackVIP	70.6	45.3	**68.7**	**85.9**	21.3	73.3	**89.1**	**65.4**	**64.5**	44.4	62.9	754.20	197.5
LLM-Opt	67.6	45.0	59.9	78.5	21.2	48.0	87.7	56.2	60.3	20.2	54.5	5011.32	5.1
AReS	**86.6**	**48.2**	67.1	68.8	**39.4**	**85.7**	88.9	43.2	62.8	**63.2**	65.4	**0.02**	**3.7**
AReS-MS	**86.6**	**48.2**	67.1	**85.9**	**39.4**	**85.7**	88.9	65.2	62.8	**63.2**	**69.3**	**0.06**	**3.7**

identity mapping for VLMs). We optimize the prompt using gradient descent on our primed local encoder by minimizing the cross-entropy loss:

$$\mathbf{P}^* = \arg\min_{\mathbf{P}} \mathbb{E}_{(x,y)\sim\mathcal{D}^T}[\ell(g_{\text{out}}(\mathcal{F}_L(g_{\text{in}}(x,\mathbf{P});\theta^*)),y)]\,. \quad (2)$$

This gradient-based optimization on local model enables more stable, efficient, and rapid learning of the visual prompt compared to ZOO-based methods, resulting in better and faster adaptation to the downstream task while maintaining the benefits of input-level modifications.

Cost-free Inference. For inference on test examples from the downstream task, we apply the learned visual prompt to the input and process it through the local encoder $\hat{y} = g_{\text{out}}(\mathcal{F}_L(g_{\text{in}}(x,\mathbf{P}^*);\theta^*))$. This approach eliminates API dependency during inference, enabling truly cost-free deployment. By removing the need for constant connectivity to service models, AReS becomes particularly valuable in practical scenarios like remote field operations or edge devices with intermittent network access.

Model Selection Strategy. For applications where maximizing performance is the top priority, we also propose a simple variant named **AReS-MS** (Model Selection). This approach leverages the initial priming stage as a low-cost diagnostic tool, as it yields both the zero-shot performance of the service API and the potential performance of the primed local model. AReS-MS then automatically selects the optimal inference path: if the local model's potential meets the required performance threshold, defined as the zero-shot performance minus a tolerance τ, it proceeds with the cost-free local model; otherwise, it defaults to using the more powerful zero-shot API. This turns AReS into a more complete and robust framework that not only enables efficient adaptation but also guides practitioners to the optimal cost-performance trade-off for each individual task.

Theoretical Insight. To formally understand AReS's efficacy, we provide a theoretical analysis (detailed in Appendix 9) that bounds the performance of our primed local model, $\mathcal{F}_L$, relative to the service model, $\mathcal{F}_S$. This analysis hinges on two key assumptions: *Service Model Superiority* (Assumption 1), which reasonably posits that $\mathcal{F}_S$ is more powerful or better aligned with the downstream task data $\mathcal{D}^T$ than $\mathcal{F}_L$ initially, and ϵ-*Faithful Priming* (Assumption 2 and Definition 1). The latter states our one-time priming effectively aligns the output logit distributions of $\mathcal{F}_L$ and $\mathcal{F}_S$, with their expected L_1 difference bounded by a small ϵ, indicating a successful knowledge transfer. While our theoretical analysis assumes *logits* for tractability, our ablation studies (Fig. 3c) confirm that the performance of our practical, probability-only method is unaffected.

Leveraging these assumptions and the Lipschitz continuity of the cross-entropy loss (Lemma 1), Theorem 3 establishes a crucial performance bound. It shows that the risk of the optimally reprogrammed local model, $\mathcal{R}_L(\mathcal{D}^T,\mathbf{P}^*)$, can closely approach that of an optimally reprogrammed service model, $\mathcal{R}_S(\mathcal{D}^T,\mathbf{Q}^*)$, differing by at most ϵ:

$$\mathcal{R}_L(\mathcal{D}^T,\mathbf{P}^*) - \epsilon \le \mathcal{R}_S(\mathcal{D}^T,\mathbf{Q}^*) \le \mathcal{R}_L(\mathcal{D}^T,\mathbf{P}^*)\,. \quad (3)$$

This result offers a key insight: traditional ZOO-based methods expend considerable resources attempting to directly optimize for a low $\mathcal{R}_S(\mathcal{D}^T,\mathbf{Q}^*)$ via unstable, query-heavy processes on $\mathcal{F}_S$. AReS, however, transforms this challenge. By first achieving a faithful priming (small ϵ), the problem shifts to optimizing $\mathcal{R}_L(\mathcal{D}^T,\mathbf{P}^*)$ on the local, glass-box model $\mathcal{F}_L$. This local optimization using efficient First-Order Optimization (FOO) is more stable and drastically reduces API calls and computation, thus providing a more practical path to leveraging the service model's capabilities.

5. Experiments

Datasets and Models. We evaluate AReS on ten diverse visual recognition datasets widely used in previous works [6, 41], spanning various domains: fine-grained categorization (Flowers102 [40], StanfordCars [28]), texture recognition (DTD [12]), action recognition (UCF101 [50]), food classification (Food101 [4]), traffic sign recognition (GTSRB [20]), satellite imagery (EuroSAT [17]), animal recognition (OxfordPets [44]), scene classification (SUN397 [67]), and digit recognition (SVHN [38]). We compare our approach with SOTA closed-box VR methods: BAR [57] and BlackVIP [41]. For the VLM experiments, we additionally compare against LLM-Opt [33], a recent baseline applicable only to

Table 3. Accuracy and Efficiency comparison using ViT-B/16 (Service) in Full-shot setting.

Method	Flowers	DTD	UCF	Food	GTSRB	EuroSAT	Pets	Cars	SUN	SVHN	Avg.	#API (M)	Time (h)
VR (glass-box)	69.6	52.1	49.7	38.7	60.6	96.7	77.1	6.2	34.0	83.3	56.8	43.4	39.6
BAR	14.8	24.3	29.5	12.2	14.1	43.1	24.9	1.0	10.6	29.6	20.4	1,724.2	668.2
BlackVIP (RN50)	15.2	42.3	34.0	14.5	13.9	67.3	63.6	3.7	22.6	30.8	30.8	2,586.2	686.9
AReS (RN50)	**43.5**	**43.4**	**39.4**	**25.9**	**48.7**	**84.3**	**73.9**	**5.8**	**19.9**	**73.7**	**45.9**	**0.2**	**8.7**
BlackVIP (ViT-B/32)	27.4	44.1	36.6	18.5	38.4	59.6	65.4	3.9	23.7	30.3	34.8	2,586.2	757.6
AReS (ViT-B/32)	**53.6**	**46.9**	**41.4**	**27.3**	**58.8**	**95.6**	**65.0**	**4.7**	**26.9**	**83.5**	**50.4**	**0.2**	**22.4**

VLMs. For VLMs, we use CLIP ViT-B/16 [48] as the service model and ViT-B/16 as the local encoder in a 16-shot learning setting, following the same configuration as BlackVIP [41] due to VLMs' strong generalization capabilities. We also include zero-shot CLIP performance as an important baseline to measure adaptation gains. For VMs, we employ pre-trained ViT-B/16 [13] and ResNet101 (RN101) [16] as service models with ViT-B/32 and ResNet50 (RN50) as local encoders, adopting the Full-shot setting following BAR [57] since VMs are not as generalizable as VLMs. For all settings using VMs as a service, we employ Bayesian-guided Label Mapping (BLM) [6] as g_{out}. In addition, we further evaluate our methods on the MLLM LLaVA [30], the real-world API GPT-4o [23], and the commercial VM API service Clarifai. Additional details can be found in Appendix 8.

Implementation Details and Evaluation Metrics. For the priming stage in AReS, we use the Adam optimizer [26] with a learning rate of 0.001. For local encoder VR, we follow the implementation in [6], using a padding-based visual prompt approach and the Adam optimizer with a learning rate of 0.01. For baselines, we use the officially released implementations with their recommended hyperparameter settings. All experiments are conducted over *three* rounds on a single NVIDIA RTX 6000 Ada GPU. We evaluate each method using three key metrics to assess practical deployment value: (1) Average accuracy across three runs on downstream tasks (**%**), (2) Total training and inference API calls required in millions (**#API (M)**), and (3) Total wall-clock training time in hours (**Time (h)**) across all datasets. For experiments on real-world APIs, we also report the total financial cost in US dollars ($) for both training and inference.

5.1. Experimental Results

Results on Vision-Language Model (VLMs). Following BlackVIP [41], we evaluate our approach using a CLIP ViT-B/16 service model and a ViT-B/16 local encoder, with results presented in Table 2. Our method, **AReS**, achieves a 65.4% average accuracy across ten diverse datasets, outperforming both BAR (61.4%) and BlackVIP (62.9%). We also compare against LLM-Opt [33], a recent method that uses GPT-4 to optimize text prompts; despite its extremely high API costs for both CLIP (5011.3M API calls) and the LLM optimizer (additional ∼$110 USD), it underperforms the zero-shot baseline and is only applicable to VLMs. This superior performance by AReS is achieved with a drastic reduction in computational requirements: only 0.02M API calls for the initial priming, compared to 612.84M for BAR and 754.20M for BlackVIP (a reduction of $> 99.99\%$). Furthermore, inferring all datasets' test sets (e.g., zero-shot) requires 0.12M API calls; our method uses only approximately 1/6 of these calls for priming and then requires no API calls during inference, while still outperforming zero-shot by $+8.2\%$ on average. Training time is also significantly reduced to only 3.7 hours, compared to 185.6 and 197.5 hours for BAR and BlackVIP, respectively (a reduction of $> 98\%$). These efficiency gains are particularly notable on challenging datasets such as SVHN (63.2% vs. 44.4% for BlackVIP and 34.6% for BAR) and EuroSAT (85.7% vs. 73.3% for BlackVIP and 77.3% for BAR). Our performance on Flowers102 is also strong at 86.6%, substantially outperforming BlackVIP (70.6%) and BAR (71.0%), and closely approaching the glass-box VR performance (86.8%) without direct use of the service model.

On certain complex datasets, notably Food101 and Cars, we observe that all evaluated reprogramming methods, including our own (with a failure case analysis provided in Appendix 10.7), struggle to outperform the strong zero-shot CLIP baseline. This suggests a potential inherent limitation of input-level visual reprogramming for these specific domains. This hypothesis is strongly supported by the results in Table 2, where a fully glass-box VR approach is shown to be ineffective, even causing a 4.3% performance *decrease* on Food101 compared to the zero-shot baseline. This insight motivates our **AReS-MS** variant, which functions as an intelligent model selection strategy. By using the priming stage to identify these challenging datasets and defaulting to the zero-shot API, AReS-MS boosts the average accuracy to a new state-of-the-art 69.3% (Table 2), effectively navigating the cost-performance trade-off by applying reprogramming only where it is beneficial.

Results on Vision Models (VMs). We further validate our approach using ViT-B/16 as the closed-box service model, paired with ViT-B/32 and RN50 local encoders (details in Table 3; results using an RN101 service model are in Appendix Table 8). Our method consistently outperforms BAR and BlackVIP across all configurations. As shown in the main results, we achieve an average accuracy of 50.4% (with a ViT-B/32 local encoder), compared to significantly lower accuracy for BAR (20.4%) and BlackVIP (34.8%). This su-

Table 4. Performance and Cost comparison on the open-source MLLM LLaVA, the proprietary API GPT-4o, and the commercial VM API Clarifai, evaluated on the EuroSAT 16-shot benchmark. A zero-shot baseline is not applicable for Clarifai as it is a closed-vocabulary model. Costs are shown in total API calls (#) or US dollars ($). AReS demonstrates significant gains where ZOO-based methods struggle.

Method	LLaVA				GPT-4o				Clarifai			
	Acc. ↑	Train	Infer.	Total (#) ↓	Acc. ↑	Train	Infer.	Total ($) ↓	Acc. ↑	Train	Infer.	Total ($) ↓
glass-box	91.3	–	–	–	–	–	–	–	–	–	–	–
Zero-shot	40.1	–	8100	8100	59.4	–	14.6	14.6	–	–	–	–
BAR	34.1	$\sim 10^6$	8100	$\sim 10^6$	59.1	57.6	14.6	72.2	68.1	38.4	9.7	48.1
BlackVIP	39.4	$\sim 10^7$	8100	$\sim 10^7$	60.1	86.4	14.6	101.0	72.1	57.6	9.7	67.3
AReS	**73.1**	160	0	**160**	**87.2**	0.3	0	**0.3**	**83.2**	0.2	0	**0.2**

perior performance comes with drastically reduced resource requirements; in the full-shot setting, our method uses less than 0.02% of the API calls and only 3-7% of the computation time required by competitors like BlackVIP, which demands over 2500M API calls and 750 hours of training.

To further analyze the effectiveness of our priming stage, we compare AReS against a direct glass-box VR baseline performed on the local model. Surprisingly, simply reprogramming the local encoder directly already outperforms the complex ZOO-based BlackVIP (e.g., 45.3% vs. 34.8% for the ViT-B/32 local model in Appendix Table 11). Our priming process then further elevates this strong baseline by a significant margin (e.g., an additional $+5.1\%$ to 50.4%). Moreover, our approach often exhibits synergistic effects, surpassing both the local VR baseline and the estimated glass-box performance of the service model on several datasets. For instance, with an RN101 service and ViT-B/32 local model, on Flowers102 our method achieves 51.3% (vs. 42.3% service glass-box vs. 38.7% local VR). A detailed analysis is in Appendix Sec. 10.5. This demonstrates our method's potential to effectively combine knowledge from both models for enhanced reprogramming outcomes with high efficiency.

Results on MLLMs and Real-World APIs. To validate AReS's practical effectiveness and efficiency, we conducted a targeted evaluation on modern services using the EuroSAT 16-shot benchmark. We test AReS against baselines on three distinct models: the open-source MLLM LLaVA, the proprietary API GPT-4o, and the commercial VM API Clarifai. For the commercial APIs, we report the real-world cost in USD, for which glass-box baselines are unavailable. AReS demonstrates substantial gains in both accuracy and efficiency in Table 4, confirming its value in real-world settings.

AReS's strong performance stems from its conceptual strategy shift, which circumvents challenges faced by ZOO-based methods on modern models. On LLaVA, for example, AReS avoids using input perturbations that appear to disrupt the model's delicate vision-language alignment, a problem that causes BAR and BlackVIP's performance to drop below the Zero-shot baseline. Similarly, on the robust GPT-4o, where ZOO-based methods struggle to find effective perturbations, AReS's local reprogramming remains highly effective, achieving a remarkable +27.8% gain over Zero-shot.

Table 5. Ablation study across Service/Local model backbones.

Method	**Service Backbone**				
	RN50	RN101	ViT-B/32	ViT-B/16	Avg.
ZS	37.5	32.6	45.2	47.9	40.8
BAR	26.9	33.5	70.3	77.2	52.0
BlackVIP (RN50)	51.3	50.8	62.9	68.5	58.4
AReS (RN50)	**62.2**	**68.5**	**63.2**	**78.6**	**68.1**
BlackVIP (ViT-B/16)	48.4	51.3	67.9	73.3	60.2
AReS (ViT-B/16)	**81.8**	**81.9**	**82.0**	**85.7**	**82.9**

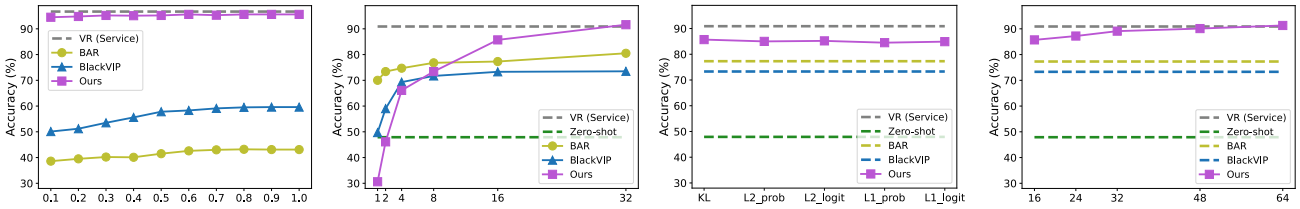
On the commercial Clarifai API, a *closed-vocabulary* model where zero-shot is not an option, AReS again achieves the highest accuracy (83.2%) at a fraction of the cost ($0.2), making it over *300*x *cheaper* than BlackVIP ($67.3). These results confirm that AReS is a robust, effective, and economically viable solution for adapting modern closed-box models. More details in the Appendix Sec. 10.8.

5.2. Ablation Study

We present ablation studies in this section. All experiments use EuroSAT with 16 shots per class, CLIP ViT-B/16 (Service) and ViT-B/16 (Local), unless specified otherwise.

Backbone Versatility. We examine the effect of varying service model backbones and local encoder choices to demonstrate our approach's versatility when the service model is VLM. The results in Table 5 indicate that our method, whether using a ViT-B/16 or an RN50 local encoder, consistently outperforms baselines like BAR and BlackVIP across diverse CLIP backbones (RN50, RN101, ViT-B/32, and ViT-B/16) used in VLM contexts. For instance, averaged across these service backbones, our method with a ViT-B/16 local encoder achieves an average accuracy of 82.9%, substantially higher than BlackVIP with the same local encoder (60.2%). This underscores the robustness of our method across different architectural choices.

Component Analysis. We dissect the contributions of our core components in Table 6: Priming, Visual Reprogramming (VR). We observe that priming the local encoder alone ("Primed") yields 45.6%, which is slightly below the 47.9% zero-shot baseline. It is important to note that this Primed result is not always directly comparable; the priming occurs in the service model's label space (e.g., ImageNet), which



(a) Data Fraction (b) Few-Shot (c) Priming Loss (d) Extra Unlabeled Data

Figure 3. Ablation studies for AReS on the EuroSAT dataset: impact of (a) training data fraction (for VMs), (b) few-shot sample size (for VLMs), (c) priming loss function, and (d) amount of extra unlabeled data for priming on accuracy (%). Default setting involves 16-shot learning, CLIP ViT-B/16 Service, and ViT-B/16 Local encoder unless otherwise indicated by the subplot analysis.

Table 6. Component analysis of Priming, Visual Reprogramming (VR), and Linear Probing (LP).

Method	Priming	VR	LP	Accuracy
Zero-shot	–	–	–	47.9
Primed	✓	–	–	45.6
Local VR	–	✓	–	70.6
Local LP	–	–	✓	73.8
Local VR+LP	–	✓	✓	80.1
AReS (Ours)	✓	✓	–	**85.7**

can be disjoint from the target task's labels (e.g., EuroSAT). This preparatory focus, which does not require a shared label space, fundamentally distinguishes our approach from conventional knowledge distillation. While the Primed model's accuracy is lower than the zero-shot baseline, an expected outcome, as its performance is bounded by the service model, the true goal is to prepare the local model for reprogramming. Our final AReS result (85.7%) demonstrates that the local model's reprogrammability is significantly enhanced by this step. Local adaptation methods without priming show varied success, with Local VR (70.6%), Local LP (73.8%), and their combination "Local VR+LP" (80.1%) all improving on the baseline. However, our full AReS approach, which combines Priming with Reprogramming, achieves the highest accuracy of 85.7%. This result demonstrates that both components are crucial, outperforming all local-only tuning baselines, and confirms that the priming stage synergistically prepares the local model for a more effective reprogramming.

Impact of Training Data Size. The impact of training data size is investigated for both VMs and VLMs. For VMs, Fig. 3a shows our method consistently outperforms BAR and BlackVIP, maintaining significant performance even when data is scarce. More VM evaluation in the 16-shot setting is provided in Appendix Sec. 10.1. For VLMs, varying the number of samples per class in a few-shot setting is presented in Fig. 3b. While at very low shot counts (e.g., < 8 shots), the performance of our method can be comparable or slightly underperform some baselines due to suboptimal priming and local VR training from extreme data scarcity, it stabilizes quickly. From 8 shots onwards, our method consistently and significantly outperforms baselines, and with more data (e.g., 32-shots), its performance can be further improved to levels comparable to glass-box service model VR.

Priming Strategies. We explore different priming loss functions and the utility of extra unlabeled data, with results in Fig. 3c and Fig. 3d respectively. Among various priming losses (KL divergence, L1/L2 losses on probabilities or logits), KL divergence, used in our main experiments, provides strong and stable performance. While we operate under the strict closed-box assumption (probabilities only), logits-based losses are included for broader comparative analysis. Furthermore, a key benefit of priming is its lack of requirement for extra labeled data, which allows our method to leverage unlabeled downstream task data. As shown in Fig. 3d, increasing the amount of unlabeled data for the priming stage (while keeping VR training fixed at 16-shot) steadily improves our method's performance. This ability to benefit from extra unlabeled data is a unique advantage of our approach compared to BMR baselines (BAR and BlackVIP), which cannot utilize such data in this manner.

6. Conclusion

In this work, we addressed key challenges in adapting closed-box Model-as-a-Service APIs, including the heavy query costs, slow optimization, and diminishing effectiveness of ZOO-based methods on modern service models. We introduced AReS, an alternative reprogramming strategy that uses a single-pass interaction with the service API to prepare a local encoder, enabling efficient glass-box reprogramming without further API dependence. This simple, theoretically-supported design yields substantial gains in both performance and practicality. Across diverse benchmarks, AReS consistently outperforms state-of-the-art ZOO approaches on VMs and VLMs, and delivers large improvements on proprietary APIs such as GPT-4o while ZOO-based approaches falter. By dramatically reducing API calls by over 99.99%, AReS enables cost-free inference and offline deployment. These comprehensive findings underscore AReS as a robust, economical, and highly effective solution for closed-box model adaptation. We show that reprogrammability can be efficiently unlocked in a local model through a minimal, one-time priming interaction, facilitating a more practical path for transfer learning in API-centric ecosystems.

Acknowledgment

We thank the anonymous reviewers for their insightful comments and suggestions. This work was supported in part by the NSF EPSCoR-Louisiana Materials Design Alliance (LAMDA) program #OIA-1946231.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 2

[2] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27, 2014. 4

[3] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 3, 13

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 5, 14

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[6] Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Bayesian-guided label mapping for visual reprogramming. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3, 5, 6, 13

[7] Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Sample-specific masks for visual reprogramming-based prompting. In *International Conference on Machine Learning (ICML)*, 2024. 1

[8] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and improving visual prompting: A label-mapping perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19133–19143, 2023. 3, 13

[9] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017. 4

[10] Hongzhan Chen, Siyue Wu, Xiaojun Quan, Rui Wang, Ming Yan, and Ji Zhang. MCC-KD: Multi-CoT consistent knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6805–6820, Singapore, 2023. Association for Computational Linguistics. 25

[11] Pin-Yu Chen. Model reprogramming: Resource-efficient cross-domain machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 22584–22591, 2024. 1, 3, 13

[12] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 5, 14

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 2, 6

[14] Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. In *ICLR*, 2018. 13

[15] Zixian Guo, Yuxiang Wei, Ming Liu, Zhilong Ji, Jinfeng Bai, Yiwen Guo, and Wangmeng Zuo. Black-box tuning of vision-language models with effective gradient approximation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5356–5368, Singapore, 2023. Association for Computational Linguistics. 2, 3, 4, 15

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 6

[17] 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. 5, 14

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 4

[19] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, 2023. 25

[20] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *IJCNN*, 2013. 5, 14

[21] Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada, 2023. Association for Computational Linguistics. 25

[22] Daniel Hsu, Ziwei Ji, Matus Telgarsky, and Lan Wang. Generalization bounds via distillation. *arXiv preprint arXiv:2104.05641*, 2021. 4

[23] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 6

[24] Guangda Ji and Zhanxing Zhu. Knowledge distillation in wide neural networks: Risk bound, data efficiency and imperfect teacher. *Advances in Neural Information Processing Systems*, 33:20823–20833, 2020. 4

[25] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022. 1, 3, 13, 26

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[27] Eliska Kloberdanz, Jin Tian, and Wei Le. An improved (adversarial) reprogramming technique for neural networks. In *ICANN*, 2021. 13

[28] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshops*, 2013. 5, 14

[29] Nilakshan Kuna., Jing Zhang, and Mehrtash Harandi. Lavip: Language-grounded visual prompting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2840–2848, 2024. 4, 15

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 1, 2, 6, 15

[31] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in neural information processing systems*, 31, 2018. 1, 3, 14

[32] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O. Hero III, and Pramod K. Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020. 1, 2, 4

[33] Shihong Liu, Samuel Yu, Zhiqiu Lin, Deepak Pathak, and Deva Ramanan. Language models as black-box optimizers for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12687–12697, 2024. 4, 5, 6, 15, 16

[34] Jueqing Lu, Wray Buntine, Yuanyuan Qi, Joanna Dipnall, Belinda Gabbe, and Lan Du. Navigating conflicting views: Harnessing trust for learning. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 40411–40435. PMLR, 2025. 21

[35] Sarthak Kumar Maharana, Shambhavi Mishra, Yunbei Zhang, Shuaicheng Niu, Taki Hasan Rafi, Jihun Hamm, Marco Pedersoli, Jose Dolz, and Yunhui Guo. Continual test-time adaptation: A comprehensive survey. 2026. 27

[36] Zheda Mai, Ping Zhang, Cheng-Hao Tu, Hong-You Chen, Quang-Huy Nguyen, Li Zhang, and Wei-Lun Chao. Lessons and insights from a unifying study of parameter-efficient fine-tuning (peft) in visual recognition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14845–14857, 2025. 26

[37] Akshay Mehra, Yunbei Zhang, and Jihun Hamm. Understanding the transferability of representations via task-relatedness. In *Advances in Neural Information Processing Systems*, pages 116513–116546. Curran Associates, Inc., 2024. 17

[38] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NeurIPS workshop*, 2011. 5, 14

[39] Dang Nguyen, Sunil Gupta, Kien Do, and Svetha Venkatesh. Black-box few-shot knowledge distillation. In *European conference on computer vision*, pages 196–211. Springer, 2022. 25

[40] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*. IEEE, 2008. 5, 14

[41] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24224–24235, 2023. 1, 2, 3, 4, 5, 6, 13, 14, 15, 16, 21

[42] Yassine Ouali, Adrian Bulat, Brais Martinez, and Georgios Tzimiropoulos. Black box few-shot adaptation for vision-language models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3, 4, 15

[43] Seonghwan Park, Jaehyeon Jeong, Yongjun Kim, Jaeho Lee, and Namhoon Lee. ZIP: An efficient zeroth-order prompt tuning for black-box vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2, 27

[44] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012. 5, 14

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilimuntala, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chaudhuri. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 16

[46] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *International conference on machine learning*, pages 5142–5151. PMLR, 2019. 4

[47] Yuanyuan Qi, Xiaohao Yang, Jueqing Lu, Guoxiang Guo, Joanne Enticott, Liu Gang, and Lan Du. Next generation active learning: Mixture of llms in the loop. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. 26

[48] 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. 1, 2, 6, 16

[49] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 22

[50] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5, 14

[51] James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992. 1, 4, 14

[52] James C Spall. A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, 33(1): 109–112, 1997. 1, 4, 14

[53] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? *Advances in neural information processing systems*, 34:6906–6919, 2021. 4

[54] Haotian Sun, Yuchen Zhuang, Wei Wei, Chao Zhang, and Bo Dai. BBox-adapter: Lightweight adapting for black-box large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 47280–47304. PMLR, 2024. 1, 2

[55] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *Proceedings of ICML*, 2022. 1, 2

[56] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1

[57] 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 *International Conference on Machine Learning*, pages 9614–9624. PMLR, 2020. 1, 2, 3, 4, 5, 6, 13, 14, 15

[58] Hsi-Ai Tsao, Lei Hsiung, Pin-Yu Chen, Sijia Liu, and Tsung-Yi Ho. AutoVP: An Automated Visual Prompting Framework and Benchmark. In *The Twelfth International Conference on Learning Representations*, 2024. 3, 13

[59] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):180161, 2018. 24

[60] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 742–749, 2019. 1, 3, 14

[61] Cheng-Hao Tu, Zheda Mai, and Wei-Lun Chao. Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7725–7735, 2023. 26

[62] Janet Wang, Xin Hu, Yunbei Zhang, Diabate Almamy, Vagamon Bamba, Konan Amos Sébastien Koffi, Yao Koffi Aubin, Zhengming Ding, Jihun Hamm, and Rie R. Yotsu. eskinhealth: A multimodal dataset for neglected tropical skin diseases. In *Proceedings of the 33rd ACM International Conference on Multimedia*, page 12949–12956, New York, NY, USA, 2025. Association for Computing Machinery. 24

[63] Janet Wang, Yunbei Zhang, Zhengming Ding, and Jihun Hamm. Doctor approved: Generating medically accurate skin disease images through ai-expert feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 1, 24

[64] Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Connecting the dots: Collaborative fine-tuning for black-box vision-language models. In *Proceedings of International Conference on Machine Learning (ICML)*, 2024. 1, 2, 3, 4, 15

[65] Zhuxuanzi Wang, Mingqiao Mo, Xi Xiao, Chen Liu, Chenrui Ma, Yunbei Zhang, Xiao Wang, Smita Krishnaswamy, and Tianyang Wang. Ctr-lora: Curvature-aware and trust-region guided low-rank adaptation for large language models. *arXiv preprint arXiv:2510.15962*, 2025. 26

[66] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 16

[67] 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. 5, 14

[68] Xi Xiao, Yunbei Zhang, Xingjian Li, Tianyang Wang, Xiao Wang, Yuxiang Wei, Jihun Hamm, and Min Xu. Visual instance-aware prompt tuning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, page 2880–2889, New York, NY, USA, 2025. Association for Computing Machinery. 26

[69] Xi Xiao, Yunbei Zhang, Yanshuh Li, Xingjian Li, Tianyang Wang, Jihun Hamm, Xiao Wang, and Min Xu. Visual variational autoencoder prompt tuning. *arXiv preprint arXiv:2503.17650*, 2025.

[70] Xi Xiao, Chenrui Ma, Yunbei Zhang, Chen Liu, Zhuxuanzi Wang, Yanshu Li, Lin Zhao, Guosheng Hu, Tianyang Wang, and Hao Xu. Not all directions matter: Toward structured and task-aware low-rank adaptation. *arXiv preprint arXiv:2603.14228*, 2026. 26

[71] Xi Xiao, Yunbei Zhang, Lin Zhao, Yiyang Liu, Xiaoying Liao, Zheda Mai, Xingjian Li, Xiao Wang, Hao Xu, Jihun Hamm, Xue Lin, Min Xu, Qifan Wang, Tianyang Wang, and Cheng Han. Prompt-based adaptation in large-scale vision models: A survey. *Transactions on Machine Learning Research*, 2026. 3, 13

[72] Siyuan Yan, Ming Hu, Yiwen Jiang, Xieji Li, Hao Fei, Philipp Tschandl, Harald Kittler, and Zongyuan Ge. Derm1m: A million-scale vision-language dataset aligned with clinical ontology knowledge for dermatology. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12681–12690, 2025. 24

[73] Lang Yu, Qin Chen, Jiaju Lin, and Liang He. Black-box prompt tuning for vision-language model as a service. In *IJCAI*, pages 1686–1694, 2023. 1, 2, 3, 4, 15

[74] Runjia Zeng, Cheng Han, Qifan Wang, Chunshu Wu, Tong Geng, Lifu Huang, Ying Nian Wu, and Dongfang Liu. Visual fourier prompt tuning. In *Advances in Neural Information Processing Systems*, pages 5552–5585. Curran Associates, Inc., 2024. 3

[75] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*. 15

[76] Guanhua Zhang, Yihua Zhang, Yang Zhang, Wenqi Fan, Qing Li, Sijia Liu, and Shiyu Chang. Fairness reprogramming. *Advances in neural information processing systems*, 35:34347–34362, 2022. 3, 13

[77] Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D Lee, Wotao

Yin, Mingyi Hong, et al. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: a benchmark. In *Proceedings of the 41st International Conference on Machine Learning*, pages 59173–59190, 2024. 1, 2, 4

[78] Yunbei Zhang, Akshay Mehra, and Jihun Hamm. Ot-vp: Optimal transport-guided visual prompting for test-time adaptation. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1122–1132. IEEE, 2025. Oral Presentation. 3, 27

[79] Yunbei Zhang, Akshay Mehra, Shuaicheng Niu, and Jihun Hamm. Dpcore: Dynamic prompt coreset for continual test-time adaptation. In *Forty-second International Conference on Machine Learning (ICML)*, 2025. 1, 27

[80] Yunbei Zhang, Shuaicheng Niu, Feng Liu, and Jihun Hamm. Adapting in the Dark: Towards Stable and Efficient Black-Box Test-Time Adaptation. *OpenReview*, 2026. 3, 27

[81] Lin Zhao, Yushu Wu, Xinru Jiang, Jianyang Gu, Yanzhi Wang, Xiaolin Xu, Pu Zhao, and Xue Lin. Taming diffusion for dataset distillation with high representativeness. In *International Conference on Machine Learning*, pages 77760–77780. PMLR, 2025. 26

[82] Lin Zhao, Xinru Jiang, Xi Xiao, Qihui Fan, Lei Lu, Yanzhi Wang, Xue Lin, Octavia Camps, Pu Zhao, and Jianyang Gu. Hieramp: Coarse-to-fine autoregressive amplification for generative dataset distillation. *arXiv preprint arXiv:2603.06932*, 2026. 26

[83] Lin Zhao, Yushu Wu, Aleksei Lebedev, Dishani Lahiri, Meng Dong, Arpit Sahni, Michael Vasilkovsky, Hao Chen, Ju Hu, Aliaksandr Siarohin, et al. S2dit: Sandwich diffusion transformer for mobile streaming video generation. *arXiv preprint arXiv:2601.12719*, 2026. 26

[84] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022. 3

[85] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3