Adapting in the Dark: Efficient and Stable Test-Time Adaptation for Black-Box Models

Yunbei Zhang[1,†] **Shuaicheng Niu**[2] **Chengyi Cai**[3] **Feng Liu**[3] **Jihun Hamm**[1,†]

[1]Tulane University [2]Nanyang Technological University [3]The University of Melbourne

Abstract

Test-Time Adaptation (TTA) for black-box models accessible only via APIs remains a largely unexplored challenge. Existing approaches such as post-hoc output refinement offer limited adaptive capacity, while Zeroth-Order Optimization (ZOO) enables input-space adaptation but faces high query costs and optimization challenges in the unsupervised TTA setting. We introduce **BETA** (Black-box Efficient Test-time Adaptation), a framework that addresses these limitations by employing a lightweight, local white-box steering model to create a tractable gradient pathway. Through a prediction harmonization technique combined with consistency regularization and prompt learning-oriented filtering, BETA enables stable adaptation with no additional API calls and negligible latency beyond standard inference. BETA achieves a 7.1-percentage-point accuracy gain on ViT-B/16 (ImageNet-C) and 3.4 points on CLIP (ImageNet-S/R), surpassing several strong white-box and gray-box methods including TENT and TPT. On a commercial API, BETA achieves comparable performance to ZOO at 250× lower cost while maintaining real-time inference speed, establishing it as a practical and efficient solution for real-world black-box TTA. **Project website: https://yunbeizhang.github.io/BETA/.**

1 Introduction

Modern deep learning models often suffer performance degradation when deployed in the wild due to distribution shifts between training and test data [1, 2]. Test-Time Adaptation (TTA) [3–7] addresses this by adapting a pre-trained model on-the-fly using unlabeled target data. However, feasible adaptation strategies depend critically on the level of model access. While white-box methods assume full access to parameters and gradients [4, 5], many state-of-the-art (SOTA) models are now deployed as opaque APIs [8, 9], where users can only submit inputs and receive output predictions.

We study TTA in this *strict black-box setting*. Concretely, from the user's perspective, one can only send a raw image to the API and receive output prediction probabilities; the model's architecture, parameters, and pre-training data remain entirely unknown. Beyond the adaptation challenge itself, each API call incurs monetary cost and network latency, making query efficiency a first-class concern. An effective black-box TTA method must therefore be not only accurate but also economical, ideally requiring only a single API call per test sample. This setting is fundamentally more challenging than both white-box TTA and supervised black-box adaptation. White-box methods [4–6] rely on backpropagation through model parameters, which is impossible without internal access. Supervised approaches such as BlackVIP [10] use labeled support sets to guide prompt learning, but TTA must operate on unlabeled test streams without ground-truth. The combination of no gradient access and no labels makes black-box TTA particularly difficult.

Recent backpropagation-free methods [11–14] improve efficiency but still require access to internal tokens or features, placing them in a "gray-box" category (illustrated in Table 1). Feature-based output refinement methods such as LAME [15] also fall into the gray-box category: their affinity

[†]Corresponding authors: Yunbei Zhang (yzhang111@tulane.edu) and Jihun Hamm (jhamm3@tulane.edu).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Comparison of TTA methods across key capabilities. We evaluate each method's requirements for accessing model parameters, internal tokens, intermediate features, and gradients, alongside its visual encoder architectural flexibility, support for different model types (VMs/VLMs), query efficiency (One API call per test sample), and inference latency.

Access	Method	w/o Params.	w/o Tokens	w/o Feats.	w/o Grad.	Arch-Agnostic	VMs	VLMs	1 API/Sample	Low Latency
□	TENT [4]	✗	✗	✗	✗	✓	✓	✓	✓	✓
	TPT [7]	✗	✗	✗	✗	✓	✗	✓	✓	✓
▒	T3A [27]	✗	✓	✗	✓	✓	✓	✓	✗	✓
	FOA [11]	✓	✗	✗	✓	ViT-only	✓	✓	✗	✗
	B^2TPT [28]	✓	✗	✓	✓	ViT-only	✗	✓	✗	✗
	BCA [14]	✓	✓	✗	✗	✓	✓	✓	✓	✓
	LAME [15]	✓	✓	✗	✓	✓	✓	✓	✓	✓
■	Augmentation [16]	✓	✓	✓	✓	✓	✓	✓	✗	✗
	Purification [17]	✓	✓	✓	✓	✓	✓	✓	✗	✗
	ZOO	✓	✓	✓	✓	✓	✓	✓	✗	✗
	BETA (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓

graphs require target features. They refine predictions post-hoc but offer limited adaptive capacity. Truly black-box input modification methods face practical constraints. Augmentation-based approaches [16] aggregate predictions over multiple augmented views, but this linearly increases API costs. Purification-based methods [17] use diffusion models to denoise inputs, but they require training on the source domain data and introduce substantial inference latency due to iterative denoising. Visual prompting via Zeroth-Order Optimization (ZOO) [18–22] can learn input perturbations without gradients, but suffers from prohibitive query costs and catastrophic instability when guided by noisy unsupervised signals [23, 24]. For instance, accuracy on the Contrast corruption collapses from 32.6% to 4.1% with ZOO (shown in Table 2).

We propose Black-box Efficient Test-time Adaptation (**BETA**), a framework that achieves stable and efficient adaptation using a local, lightweight steering model. This steering model is initialized from public checkpoints and operates entirely on the client side, requiring no access to the target model's internals or pre-training data. Since naive gradient transfer between different architectures is ineffective (Fig. 2), BETA instead employs *prediction harmonization* to fuse outputs from both models, creating a shared objective optimized through the steering model's gradient pathway. To address the instability of learning prompts from random initialization (Fig. 3), we introduce *consistency regularization* and *prompt learning-oriented filtering*, which together ensure robust adaptation without any ground-truth labels.

BETA achieves strong results across diverse settings. On ImageNet-C with ViT-B/16, it achieves 62.6% accuracy (+7.1 percentage points), outperforming white-box methods like TENT [4] and CoTTA [6] with only a single API call per sample. On CLIP, BETA reaches 63.4% accuracy, surpassing specialized VLM methods including TPT [7], DynaPrompt [25], and TCA [26]. On the real-world commercial Clarifai API, BETA achieves a **5.2-point** gain for just \$0.4, while ZOO requires over \$100 for comparable performance, demonstrating a **250**× cost advantage.

Contributions. (1) We provide the first systematic study of TTA in the strict black-box setting, revealing the limitations of existing approaches including post-hoc refinement, augmentation, purification, and ZOO-based prompting. (2) We introduce BETA, which bypasses expensive queries by using a steering model with prediction harmonization, stabilized by consistency regularization and prompt-oriented filtering for fully unsupervised adaptation. We give an analytical grounding for the harmonized objective through an entropy decomposition that exposes a Jensen–Shannon alignment term between the steering model and the black-box target. (3) We establish new SOTA results for black-box TTA. BETA surpasses several strong white-box methods with no additional API calls and negligible latency beyond standard inference, making it practical for real-world deployment.

2 Method

2.1 Problem Formulation

TTA aims to adapt a model f, pre-trained on a source domain, to an unlabeled target domain $\mathcal{D}_T = \{x_j^T\}_{j=1}^{|\mathcal{D}_T|}$ encountered during inference. In the online setting, target data arrives as a stream of batches $\{B_t\}_{t=1}^{T}$, and the model is updated on-the-fly without ground-truth labels. The feasible adaptation strategies depend on the level of model access, which falls into three categories (Table 1):

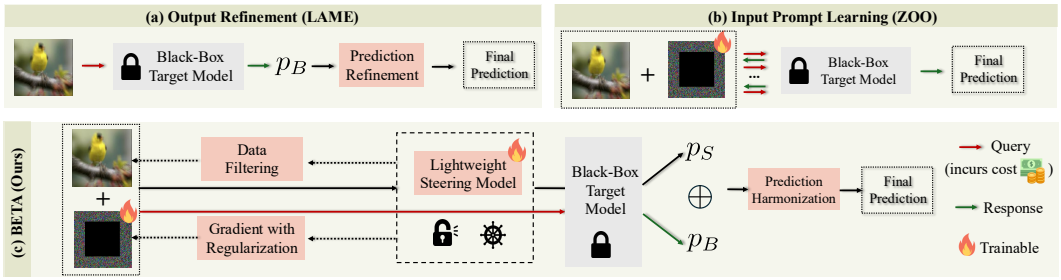


Figure 1: Comparison of test-time adaptation strategies. **(a)** Output Refinement (LAME, a gray-box reference) is limited to post-processing predictions, while **(b)** ZOO-based Input Prompt Learning requires multiple expensive API calls for prompt optimization. In contrast, **(c)** BETA achieves efficient single-query adaptation by leveraging a lightweight steering model with prediction harmonization to create a tractable gradient pathway, stabilized through data filtering and regularization. The prompted input feeds both models in parallel; gradients flow only through the local steering pathway.

- **White-Box Access** (▭): The full model architecture and all its weights are accessible. This allows for the computation of gradients via backpropagation.
- **Gray-Box Access** (▬): Intermediate representations, e.g., internal tokens or features, are accessible, while the full computational graph and partial model parameters remain hidden.
- **Black-Box Access** (■): The model is an opaque API. Users can only send raw input x and receive output probability vector $p(y|x) = f(x)$. The model's architecture, parameters, pre-training data, training algorithms, and intermediate states are entirely unknown.

API requirements. BETA assumes an exposed, closed label vocabulary that can be matched to the steering model's outputs, and a public steering model with task-relevant representations. The pixel prompt is applied before API submission; unknown API-side preprocessing remains part of the target mapping (Appx. A.1). Full probabilities are the default, but top-10 access achieves 62.2% versus 62.6% with the full vector (Appx. B.4).

Operational constraints of the black-box setting. Black-box TTA imposes three concrete constraints on feasible methods. (i) *API cost*: every query is monetarily charged, so query-heavy methods are economically infeasible. (ii) *End-to-end latency*: each call incurs network round-trip time that dominates local compute. (iii) *Local compute*: the user typically runs on a personal machine with bounded GPU memory, ruling out heavy on-device tools. The target model's internals are hidden, but the user can use any auxiliary public resource that fits within these budgets. An effective method must deliver *stable* accuracy gains on the user's stream while staying *efficient* along all three axes.

2.2 Related Work and Limitations of Existing Approaches

Test-time adaptation. TTA adapts pre-trained models on unlabeled target streams to handle distribution shifts [29, 3]. The dominant line of work assumes *white-box* access and updates internal parameters via entropy or consistency objectives [4–7, 30–36]. Backpropagation-free methods [11, 14, 26, 13, 28] avoid gradients but still consume internal tokens or features, and VLM-specific methods conditioning on text or visual embeddings [37, 24] fall in the same *gray-box* regime. Truly *black-box* TTA, where only inputs and softmax outputs are exposed, remains underexplored.

Existing black-box methods miss the goal. A separate thread studies black-box adaptation in the *offline supervised* transfer regime with labeled support sets [10, 38–44]. Transferring these techniques to online TTA is non-trivial: ground-truth labels disappear, supervised losses give way to noisy proxies (entropy, consistency), and updates must run on a streaming budget rather than a curated support set. Re-purposing these and adjacent input/output strategies for our setting yields four candidates, each violating at least one of the stability or efficiency requirements established above. *Output refinement* (LAME [15]) is query-efficient but requires target features, making it a gray-box reference with limited adaptive capacity. *Input augmentation* [45, 16] averages predictions over N views at $N\times$ API cost; some variants further require labeled source data [45] or raw logits [16] that strict APIs do not expose. *Diffusion purification* [17, 46] demands source-domain training and adds seconds-scale latency from iterative denoising, failing both the local-compute and latency axes. *ZOO-based prompting* [10, 39, 11, 24] estimates gradients via random perturbations and is the

Table 2: Classification accuracy (%) on ImageNet-C (severity 5) using **ViT-B/16** (87M) as the black-box model. BETA achieves the highest performance among black-box methods and outperforms several strong white-box approaches. *White-box and gray-box methods are shown for reference*. Within black-box methods, **bold** indicates best and underline indicates second best.

Access	Method	Noise			Blur				Weather				Digital				Avg.	Gain
		Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Bright.	Contr.	Elastic	Pixel.	JPEG		
	Source	56.8	56.8	57.5	46.9	35.6	53.1	44.8	62.2	62.5	65.7	77.7	32.6	46.0	67.0	67.6	55.5	0.0
□	TENT	60.3	61.6	61.8	59.2	56.5	63.5	59.1	54.2	64.5	2.2	79.1	67.4	61.5	72.5	70.6	59.6	+4.1
□	SAR	59.1	60.5	60.6	57.1	55.6	61.5	57.4	65.8	63.4	67.4	78.7	62.6	62.2	72.0	70.2	63.6	+8.1
□	CoTTA	63.3	63.9	64.5	55.0	51.0	63.5	56.1	68.8	69.2	71.2	78.3	9.6	64.3	73.4	71.2	61.6	+6.1
□	ETA	60.9	62.2	62.2	59.5	57.4	63.6	60.1	68.3	65.8	71.5	79.3	66.9	64.9	72.9	71.1	65.8	+10.3
▒	T3A	56.4	56.9	57.3	47.9	37.8	54.3	46.9	63.6	60.8	68.5	78.1	38.3	50.0	67.6	69.1	56.9	+1.4
▒	FOA*	57.0	58.5	57.8	51.7	35.0	37.1	27.2	20.2	11.9	72.2	76.8	0.6	39.1	66.7	67.0	45.3	-10.2
▒	LAME	56.5	56.5	57.2	46.4	34.7	52.7	44.2	58.4	61.5	63.1	77.4	24.7	44.6	66.6	67.2	54.1	-1.4
■	ZOO-CMA	58.2	59.6	60.3	50.8	38.6	55.2	45.7	58.5	59.6	59.7	76.7	4.1	49.8	71.2	70.0	54.5	-1.0
■	ZOO-RGF	59.6	58.7	60.4	47.7	37.8	53.5	44.6	58.2	61.7	63.4	76.7	26.8	49.4	70.7	70.2	56.0	+0.5
■	ZOO-SPSA-GC†	59.6	58.7	60.2	47.9	38.0	53.7	44.7	58.2	61.7	63.6	76.7	12.7	49.4	70.7	70.2	55.1	-0.4
■	TT-Aug‡	55.4	54.2	55.2	43.7	48.6	48.9	45.5	57.8	63.1	60.0	76.9	49.6	41.7	65.7	67.8	55.6	+0.1
■	DDA§	**64.7**	**65.0**	**64.6**	46.3	41.3	51.7	43.7	59.1	61.3	45.0	74.9	40.6	54.4	72.2	68.4	56.9	+1.4
■	**BETA (Ours)**	60.5	60.7	61.1	**54.5**	**52.2**	**59.9**	**56.3**	**63.6**	**64.7**	**66.1**	**78.1**	**53.4**	**62.1**	**73.3**	**72.0**	**62.6**	+7.1

*FOA [11] uses entropy minimization as the original activation discrepancy requires source statistics unavailable in the black-box setting.
†ZOO-SPSA-GC adapted from [10]; uses entropy instead of cross-entropy as no ground-truth labels are available.
‡TT-Aug adapted from Shanmugam et al. [45] which requires labeled source data; we only aggregate predictions over augmented views.
§DDA requires training a diffusion model on source data [17]; we use the released model as source data is typically unavailable.

most expressive option in principle, but is query-intensive and unstable under unsupervised entropy, occasionally collapsing accuracy by tens of points (Tables 2, 3). No existing method delivers stable adaptation while staying within the joint API-cost, latency, and local-compute budget. Per-method protocols appear in Appx. A.2.

2.3 BETA: Black-box Efficient and Stable Test-time Adaptation

The limitations of existing approaches motivate **BETA**, which seeks to combine the adaptive capacity of input prompting with the query efficiency of output-based methods. To address the inaccessibility of the target model's gradients while avoiding the high cost of ZOO, BETA operates with two models: a *target model* f_B, the powerful but inaccessible black-box API (e.g., Clarifai API) from which we can only obtain predictions $p_B(x) = f_B(y|x)$, and a *steering model* f_S, a lightweight local model (e.g., ViT-Small) with full access to parameters and gradients.

To adapt the black-box model without altering its weights, we learn an additive visual prompt $\delta \in \mathbb{R}^{H\times W\times C}$. This prompt is added to the input image x to produce a prompted version $x' = x + \delta$. The goal is to optimize δ using gradients derived locally from f_S to improve the predictions. A straightforward approach to optimize δ in the black-box setting is to employ ZOO to minimize the Shannon entropy of the model's predictions [4]: $\mathcal{H}(p_B(x')) = -\sum_{k=1}^{K} p_B^k(x') \log p_B^k(x')$, where $p_B^k(x')$ is the predicted probability for class k. However, this baseline fails: zeroth-order estimation is query-intensive, and the unsupervised entropy objective is unstable, often driving δ toward degenerate solutions that leads to catastrophic collapse on challenging domains: on the Contrast corruption, accuracy drops from 32.6% to 4.1%, 26.8%, and 12.7% across three ZOO methods (Table 2).

2.4 Prediction Harmonization

From Naive Transfer to Harmonized Relaxation. Our approach is motivated by the failure of direct gradient estimation. To formalize the analysis, we use $\nabla\mathcal{H}(p;\cdot)$ to denote the gradient of the entropy of prediction p, computed by backpropagating through the model indicated by the second argument. Our ultimate goal is to minimize the entropy of the black-box model, which requires following the *target gradient* $g_{\text{Black}} = \nabla\mathcal{H}(p_B; f_B)$. However, since f_B is inaccessible, g_{Black} is intractable. Existing alternatives fail to provide reliable substitutes: ZOO suffers from costs and instability, while naively transferring the *local gradient* $g_{\text{Local}} = \nabla\mathcal{H}(p_S; f_S)$ from the steering model is ineffective, as our analysis shows the gradient similarity between different architectures is consistently near zero (≈ 0.0006).

To overcome this, we relax the problem to finding a prompt that improves both models simultaneously. We define a *harmonized prediction* p_H that fuses the outputs of the steering and target models with a weighting parameter $\alpha \in [0, 1]$:

$$p_H(x') = \alpha \cdot p_S(x') + (1 - \alpha) \cdot p_B(x'). \tag{1}$$

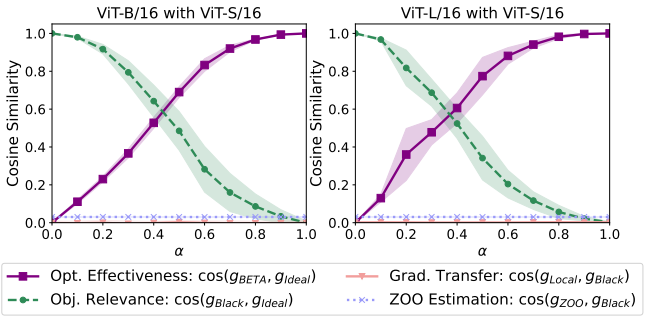


Figure 2: We analyze the trade-off between Objective Relevance (alignment with the true target gradient) and Optimization Effectiveness (alignment with the practical steering gradient) as a function of α. The intersection of these opposing curves identifies the optimal range (e.g., $\alpha \in [0.3, 0.5]$) where the objective is simultaneously relevant to the target and tractable for optimization. The curves are plotted based on the validation sets of ImageNet-C.

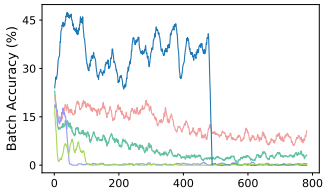


Figure 3: Five independent runs of using solely Eqn. (1), showing either performance collapse or failure to improve.

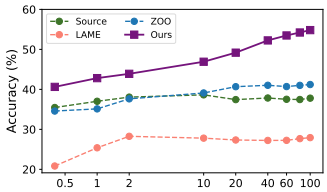


Figure 4: Performance vs. API budget on the Real-world Clarifai API.

The ideal update direction $g_{\text{Ideal}} = \nabla_\delta \mathcal{H}(p_H; f_S, f_B)$ requires backpropagating through both models. Since f_B is inaccessible, g_{Ideal} remains intractable. We employ an asymmetric optimization strategy: we compute the gradient of the same harmonized objective but restrict gradient flow exclusively to the steering model's pathway. This yields our tractable proxy $g_{\text{BETA}} = \nabla_\delta \mathcal{H}(p_H; f_S)$, which targets the joint harmonized distribution without requiring internal access to the black-box model.

Analytical insight. Since f_B admits no gradient, the entropy of the mixture p_H decomposes into $\mathcal{H}(p_S)$, a constant target-entropy term that drops out, and an α-weighted Jensen–Shannon divergence $\text{JS}_\alpha(p_S \| p_B)$. The prompt update therefore minimizes the steering entropy together with the JS divergence between the two models. The JS piece is critical: although p_B has no gradient, the divergence is a function of δ through p_S, so minimizing it pulls the prompt toward classes the target already favors. Without it, the optimization collapses to one-sided entropy and drives p_S confident on classes the target rejects (failure mode confirmed in Fig. 2). Full derivation in Appx. C.1.

Empirical Justification. To validate g_{BETA} as a proxy for g_{Ideal}, we conduct a gradient analysis across four validation corruption domains using ViT-B/16 and ViT-L/16 as black-box target models and ViT-S/16 as the local steering model. For this analysis only, we temporarily assume white-box access to the target models to compute the otherwise inaccessible vectors g_{Black} and g_{Ideal}. Simpler strategies fail (in Fig. 2): the cosine similarity between g_{Local} and g_{Black} is consistently near zero for both target models, confirming that naive gradient transfer across different architectures is ineffective. ZOO gradient estimates are equally noisy in the one-step setting despite their high query cost.

BETA's effectiveness stems from how α navigates a trade-off between two competing factors. *Objective Relevance* measures how well our tractable objective aligns with the true goal: $\text{Relevance}(\alpha) = \cos(g_{\text{Ideal}}, g_{\text{Black}})$. *Optimization Effectiveness* measures how well our practical proxy can optimize this objective: $\text{Effectiveness}(\alpha) = \cos(g_{\text{BETA}}, g_{\text{Ideal}})$. These factors oppose each other: low α yields high Relevance but negligible Effectiveness since gradients cannot flow through f_B, while high α yields perfect Effectiveness for an irrelevant objective. BETA identifies an optimal range (e.g., $\alpha \in [0.3, 0.5]$) where both are balanced. This confirms that BETA succeeds not by approximating the target gradient, but by constructing a shared optimization problem where g_{BETA} aligns with g_{Ideal}.

Zero-shot steering recipe. For specialized target domains where no pre-trained steering model is available, particularly with VLMs, we instantiate the steering model f_S as a CLIP zero-shot classifier following [47]: a small public CLIP's text encoder maps prompt-templated label names (e.g., "a photo of a [class].") into class embeddings, which serve as the weights of a linear classifier on top of CLIP's frozen visual encoder. This f_S requires no task-specific training, and its standalone accuracy on the target task is often low. Even so, BETA only needs f_S as a gradient pathway rather than a strong classifier. We validate this recipe on Derm7pt (Table 6) and EuroSAT (Table 12).

Table 3: Classification accuracy (%) on ImageNet-C (severity 5) using **ViT-L/16** (304M) as the black-box model. BETA achieves the best performance among black-box methods and outperforms several strong white-box approaches. *White-box and gray-box methods are shown for reference*. Within black-box methods, **bold** indicates best and underline indicates second best.

Access	Method	Noise			Blur				Weather				Digital				Avg.	Gain
		Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Bright.	Contr.	Elastic	Pixel.	JPEG		
	Source	62.5	62.0	63.3	52.9	45.3	60.7	55.2	66.0	62.3	62.6	79.9	40.1	56.2	74.3	72.8	61.1	0.0
□	TENT	67.2	67.3	65.4	59.2	0.9	66.7	63.8	69.7	67.0	61.9	81.0	60.3	65.4	77.3	74.1	63.1	+2.0
□	SAR	65.6	66.7	66.9	58.6	57.8	60.5	61.0	69.3	67.0	68.1	81.0	60.2	61.8	76.8	74.3	66.4	+5.3
□	CoTTA	68.3	69.7	69.9	57.1	54.2	53.5	63.2	72.5	70.4	26.2	80.9	53.5	65.6	77.1	74.9	63.8	+2.7
□	ETA	67.4	58.3	67.9	63.4	61.3	67.7	62.9	70.7	68.4	66.3	81.3	54.0	66.0	77.7	74.1	67.2	+6.1
▒	T3A	62.6	62.2	63.5	54.0	46.1	61.3	56.4	66.6	63.2	57.3	79.9	39.1	58.9	74.6	73.3	61.3	+0.2
▒	FOA*	48.1	56.1	59.1	50.2	50.6	59.6	42.4	57.5	58.8	56.1	72.2	29.1	59.5	72.0	70.4	56.1	-5.0
▒	LAME	62.2	61.6	63.0	52.4	44.9	60.3	54.8	65.5	61.7	61.7	79.8	39.9	55.4	74.1	72.4	60.6	-0.5
■	ZOO-CMA	61.7	62.5	63.1	57.1	50.4	61.6	55.4	63.9	62.5	59.5	78.4	22.5	56.5	75.8	74.2	60.3	-0.8
■	ZOO-RGF	61.3	62.9	62.2	56.9	50.9	59.5	52.5	59.0	58.9	56.9	75.7	31.2	57.1	74.7	72.4	59.5	-1.6
■	ZOO-SPSA-GC†	62.8	63.5	63.4	57.0	52.2	59.8	55.9	59.0	59.7	61.7	75.5	43.0	59.9	75.1	72.4	61.4	+0.3
■	TT-Aug‡	62.9	63.1	63.3	54.1	48.4	60.8	56.0	60.8	63.8	62.8	79.2	49.5	57.4	74.2	72.9	61.9	+0.8
■	DDA§	**68.0**	**68.3**	**68.0**	52.8	49.8	59.3	53.8	64.3	63.4	55.8	78.0	46.9	61.1	**76.4**	73.1	62.6	+1.5
■	**BETA (Ours)**	63.1	64.0	63.5	**59.7**	**55.1**	**63.6**	**59.4**	**66.1**	**65.0**	**66.2**	**80.0**	**55.1**	**65.0**	76.2	**74.5**	**65.1**	+4.0

2.5 Stabilization and Joint Optimization

Instability of Unconstrained Optimization. While the harmonized objective provides a tractable gradient pathway, our investigation reveals that this process is inherently unstable when applied in isolation. We evaluated a baseline using only the harmonized objective (Eqn. 1) on ImageNet-C Contrast with ViT-B/16 as the target model across five independent runs. As shown in Fig. 3, naively optimizing the randomly initialized prompt leads to either gradual performance decay or catastrophic collapse. This instability arises because noisy unsupervised signals can cause the optimization to learn degenerate solutions that corrupt the input's semantic features. To ensure robust adaptation, BETA incorporates two stabilization mechanisms.

Prompt Learning-oriented Data Filtering. The first mechanism filters the training signal. Updating the prompt using all incoming data degrades performance because high-entropy samples provide noisy gradients. We therefore update the prompt using only samples with prediction entropy $\mathcal{H}(p_S(x))$ below a threshold ϵ. This filtering is integrated into the harmonization objective via a weight term:

$$\mathcal{L}_{\text{Harmon}}(x') = w_H(x')\mathcal{H}(p_H(x')), \tag{2}$$

where $w_H(x) = \frac{1}{\exp[\mathcal{H}(p_S(x))-\epsilon]} \cdot \mathbb{I}_{\{\mathcal{H}(p_S(x))<\epsilon\}}(x)$ filters out high-entropy samples and assigns confidence-based weights to reliable ones. Unlike methods that filter for pre-trained normalization parameters, we retain all reliable samples including redundant ones for the prompt update, as learning a visual prompt from random initialization is more challenging and benefits from more data.

Consistency Regularization. While filtering removes noisy samples, the optimization itself requires regularization to prevent collapse. Since prompts are randomly initialized, an unconstrained entropy objective can be minimized by learning degenerate solutions that destroy the model's representations (Fig 3). We introduce a consistency regularization that anchors updates to the model's pre-trained knowledge by minimizing the KL-divergence between predictions on clean (x) and prompted (x') images:

$$\mathcal{L}_{\text{consist}}(x, x') := D_{\text{KL}}(p_S(x) \| p_S(x')) = \textstyle\sum_{k=1}^{K} p_S^k(x) \log \frac{p_S^k(x)}{p_S^k(x')}. \tag{3}$$

Final Objective and Joint Optimization. BETA operates in a strictly online, one-pass manner, performing a single gradient step per batch B_t to ensure minimal latency. The optimization targets are distinct: the harmonization loss and consistency regularization update the visual prompt δ, while the steering loss updates only the normalization parameters θ of the steering model. Following Niu et al. [30], we adapt the normalization layers using only samples that are both reliable and non-redundant, where non-redundancy is determined by comparing a sample's prediction against an exponential moving average of past predictions $\bar{p}_{t-1}$ using a diversity margin d. The steering loss is:

$$\mathcal{L}_{\text{Steer}}(x') = w_S(x')\mathcal{H}(p_S(x')), \tag{4}$$

where $w_S(x) = \frac{1}{\exp[\mathcal{H}(p_S(x))-\epsilon]} \cdot \mathbb{I}_{\{\mathcal{H}(p_S(x))<\epsilon\}} \cdot \mathbb{I}_{\{|\cos(p_S(x),\bar{p}_{t-1})|<d\}}$ selects samples that are both reliable and non-redundant. The final objective combines all components:

$$\mathcal{L}_{\text{BETA}} = \mathbb{E}_{x \in B_t}\left[\mathcal{L}_{\text{Harmon}}(x') + \mathcal{L}_{\text{Steer}}(x') + \lambda \mathcal{L}_{\text{consist}}(x, x')\right]. \tag{5}$$

Table 4: Accuracy (%) on ImageNet-S/R.

Access	Method	ViT-B/16			CLIP (ViT-B/16)		
		Sketch	Rendition	Avg.	Sketch	Rendition	Avg.
	Source	44.9	59.5	52.2	46.1	74.0	60.0
□	TENT	49.1	63.9	56.5	49.5	75.3	62.4
□	SAR	48.7	63.3	56.0	49.2	76.1	62.7
□	CoTTA	50.0	63.5	56.8	50.4	75.6	63.0
□	TPT	–	–	–	48.0	77.1	62.5
□	DynaPrompt	–	–	–	48.2	78.2	63.2
□	DPE	–	–	–	52.3	80.4	66.3
▒	T3A	48.5	58.0	53.3	49.1	75.6	62.4
▒	FOA*	44.7	59.2	52.0	45.8	73.2	59.5
▒	TDA	–	–	–	50.5	80.2	65.4
▒	B^{2}TPT	–	–	–	49.5	78.6	64.1
▒	RA-TTA	–	–	–	50.8	79.7	65.3
▒	TCA	–	–	–	49.0	77.1	63.0
▒	BCA	–	–	–	50.9	80.7	65.8
▒	LAME	44.4	59.0	51.7	45.4	72.8	59.1
■	ZOO-CMA	44.7	58.8	51.8	45.6	72.5	59.1
■	ZOO-RGF	44.4	58.1	51.3	45.3	72.1	58.7
■	ZOO-SPSA-GC†	45.1	59.3	52.2	46.0	72.8	59.4
■	ZERO¶	-	-	-	48.4	**77.2**	62.8
■	**Ours**	**49.3**	**63.3**	**56.3**	**50.9**	76.0	**63.4**

¶ZERO originally requires logits for re-scaling for CLIP [16]; we adapt it to use output prediction probabilities.

Table 5: Efficiency analysis on ImageNet-C (ViT-B/16). BETA matches standard inference in API cost and latency while achieving the best accuracy.

Method	#API /Img	Local Compute	Mem (MB)	Time (ms)	Acc (%)	Gain (%)
Source	1	✗	-	45	55.5	-
LAME	1	✓	2	46	54.1	-1.4
ZOO-SPSA-GC†	16	✓	52	450	55.1	-0.4
TT-Aug‡	64	✓	-	1,800	55.6	+0.1
DDA§	2	✓	23,427	12,722	56.9	+1.4
BETA (ViT-Tiny)	1	✓	1,292	47	58.2	+2.7
BETA (ViT-Small)	**1**	✓	2,616	48	**62.6**	+7.1

Table 6: Accuracy (%) on Derm7pt (7-class skin lesion) with an off-the-shelf zero-shot CLIP steering model.

Target (Black-box)	Source	LAME	TT-Aug	Ours
CLIP ViT-B/16	55.9	56.0	56.6	**58.6**
BiomedCLIP	60.9	60.4	61.0	**62.1**

3 Experiments

Datasets and Models. We evaluate our method across several challenging benchmarks: ImageNet-C at severity level 5 [48], ImageNet-S (Sketch) [49], and ImageNet-R (Rendition) [50]. In our experiments, we treat powerful, large-scale models as the inaccessible black-box targets: standard Vision Transformers ViT-B/16 (87M parameters) and ViT-L/16 (304M), and the Vision-Language Model CLIP with a ViT-B/16 backbone (CLIP-B/16, 150M) [51, 52]. Adaptation is guided by a much smaller, fully accessible ViT-S/16 (22M) steering model. To validate BETA in a practical, real-world scenario, we also test it using a commercial Clarifai[1] API, which charges $0.0032 per request.

Compared Methods. We conduct our comparison in the *source-free Fully TTA* setting, benchmarking against methods with varying levels of model access. For White-box methods, we include those applicable to both VMs and VLMs (Tent [4], SAR [5], ETA [30], and CoTTA [6]), along with specialized approaches for VLMs (TPT [7], DynaPrompt [25], and DPE [53]). For Gray-box methods, we compare against T3A [27], LAME [15], FOA (for both VMs and VLMs) [11] and others specific to VLMs (TDA [12], B^{2}TPT [28], TCA [26], BCA [14], RA-TTA [13]). Our primary comparison is against truly *Black-box* methods: three ZOO baselines (CMA-ES, RGF, and SPSA-GC) that we implemented to learn a visual prompt, augmentation-based TT-Aug [45] and VLM-specific ZERO [16], and purification-based DDA [17].

Implementation Details. For BETA, we set the weighting parameter α to 0.4. The shared visual prompt δ is trained with the AdamW optimizer using a learning rate of 0.01. We update only the normalization layers of the local steering model using SGD with a learning rate of 2×10^{-5}. The weight for the KL consistency regularization λ is set to 50, and we set the entropy threshold $\epsilon = 0.9 \log K$ for sample filtering, where K is the number of classes. The visual prompt is structured as a padded frame with a width of 16 pixels, amounting to 39,936 learnable parameters, and is initialized from a Gaussian distribution. Hyperparameters are selected on four disjoint validation corruptions and then fixed across test corruptions, datasets, architectures, and seeds (Appx. A.3). More details are in Appx. A.

3.1 Experimental Results

Results on ImageNet-C with Vision Models. We evaluate BETA against white-box, gray-box, and black-box methods on ImageNet-C using ViT-B/16 as the black-box model (Table 2). The gray-box LAME reference fails to improve over the source. ZOO methods are inconsistent and collapse on hard domains such as Contrast while requiring far more API calls. Augmentation-based methods (TT-Aug) incur high query costs with marginal gains, while purification-based methods (DDA) introduce substantial latency and memory overhead due to iterative diffusion. In contrast, BETA consistently improves across all domains with a single API call, surpassing all black-box baselines

[1] https://www.clarifai.com/

and outperforming white-box methods like TENT and CoTTA despite operating under much stricter access constraints. This trend continues on ViT-L/16 (Table 3), where BETA again achieves the best black-box performance. Notably, the steering model (ViT-S/16) is substantially weaker than the target model, and even when fully adapted in a white-box setting, its performance remains well below the target's baseline (Appx. Table 22). Yet BETA successfully leverages this weaker model to improve the powerful black-box model, demonstrating that our method discovers and transfers beneficial adaptation signals rather than simply relying on the steering model's predictions.

Five-run stability. Full BETA obtains $62.5 \pm 0.06\%$ over 15 corruptions and $53.1 \pm 0.16\%$ on Contrast at severity 5 (mean $\pm$ SD, five independent seeds; Appx. B.1). The main tables retain the original single-run results, including 62.6% for ViT-B/16.

Results on ImageNet-S and ImageNet-R. We further evaluate BETA's generalization on ImageNet-S and ImageNet-R (Table 4). On both datasets with ViT-B/16, BETA significantly improves upon the source model, surpassing black-box baselines and outperforming white-box methods like T3A and SAR. We then extend our evaluation to VLMs, applying BETA to CLIP with a ViT-B/16 backbone. To our knowledge, this is the first work to explore TTA for VLMs in the strict black-box setting. BETA is the only black-box method that effectively improves the pre-trained CLIP model, even surpassing several white-box (TPT, DynaPrompt) and gray-box (TCA) VLM baselines. This consistent success across datasets and model types demonstrates that BETA is a general framework for black-box adaptation. More results on VLMs are provided in Appx. B.2.

Results on a Real-world API. To validate BETA's practicality, we evaluate it on a commercial Clarifai API, comparing performance against API cost on the challenging ImageNet-C Contrast domain (Fig. 4). The results demonstrate BETA's superior cost-effectiveness. With a budget of just \$0.4, sufficient to process approximately 120 test samples, BETA improves upon the source model by 5.2 percentage points. In contrast, achieving comparable performance with a query-intensive ZOO method requires over \$100, representing a *250× cost advantage* for our approach. Moreover, when given the same \$100 budget, BETA's advantage becomes even more pronounced, delivering a substantial 17.1-point gain. These results highlight BETA's significant real-world utility, establishing it as a practical and cost-effective solution for adapting commercial API-based models. Five independent runs on Contrast at \$5 per run obtain $46.7 \pm 0.57\%$ accuracy, versus 38.6% for Source. Separately, evaluation across all 15 corruptions at \$5 per domain improves mean accuracy from 44.3% to 54.7% (Appx. B.3).

Generalization beyond ImageNet. We test BETA where neither the target nor any natural pre-trained steering model is a close semantic match for the task: satellite remote sensing (EuroSAT [54]) and medical dermatology (Derm7pt [55]). For both, f_S is built zero-shot from CLIP ViT-B/32's text embeddings of the target labels, with no task-specific training. On EuroSAT (Table 12), BETA with a CLIP ViT-B/16 target reaches 53.3%, an 11.3-point gain over the source (42.0%) and above gray-box B^{2}TPT (46.8%) at $120\times$ lower query cost. Domain shifts across skin-lesion datasets motivate evaluating adaptation in medical imaging [56]. On Derm7pt (Table 6) we further stress the mismatch by switching the black-box target to BiomedCLIP [57] while keeping the steering model on natural-image CLIP; BETA improves both targets. These results confirm that BETA does not need a domain-strong steering model, only one that supplies a usable gradient signal. Sensitivity to the choice of zero-shot CLIP (RN50 vs. ViT-B/32) is reported in Appx. B.6.

Computational Efficiency. We analyze BETA's efficiency in Table 5, measuring wall-clock latency and resource usage on a single NVIDIA RTX 3090 GPU. All black-box adaptation methods require local computation, whether CPU-based (e.g., augmentation) or GPU-based (e.g., diffusion or steering models); BETA's overhead remains within reasonable bounds for consumer-grade hardware. In this GPU-based setup, local computation is negligible compared to API latency, which dominates total inference time. Methods requiring multiple API calls per image thus suffer severe slowdowns, while BETA matches standard inference in both API cost and latency. On eight AMD Ryzen Threadripper PRO 5955WX CPU threads, ViT-T/16 takes 24.0 ms/image at batch size 8 for the complete local update, excluding API and network latency; full CPU profiles appear in Appx. B.5.

3.2 Ablation Studies

Analysis of BETA's Components. We conduct an ablation study to dissect each component's contribution, with results in Table 8. Our analysis reveals that strategies focusing solely on output

Table 7: Effect of steering model choice, with local Source/TENT references. Averages use the three displayed rows; shared ImageNet-Sketch steering results count twice.

Dataset	Black-Box Model	Source	LAME	ZOO	ViT-Tiny (6M)			ResNet50 (26M)			ViT-Small (22M)		
					Source	TENT	**BETA**	Source	TENT	**BETA**	Source	TENT	**BETA**
ImageNet-C	ViT-B/16 (87M)	55.5	54.1	56.0	21.4	22.0	58.2	24.2	31.4	60.8	39.5	51.9	**62.6**
ImageNet-Sketch	ViT-B/16 (87M)	44.9	44.4	45.1	20.9	21.3	45.2	27.9	29.7	47.5	32.8	35.6	**49.3**
	CLIP-B/16 (150M)	46.1	45.4	46.0			47.0			48.7			**50.9**
Average	-	48.8	48.0	49.0	21.1	21.5	50.1	26.7	30.3	52.3	35.0	41.0	**54.3**

Table 8: Component analysis. **In-Adapt**: input adaptation via visual prompt learning; **Out-Adapt**: output adaptation via Prediction Refinement (**PR**, LAME) or our Prediction Harmonization (**PH**); **KL Reg.**: consistency KL-regularization; **Filt.**: sample filtering.

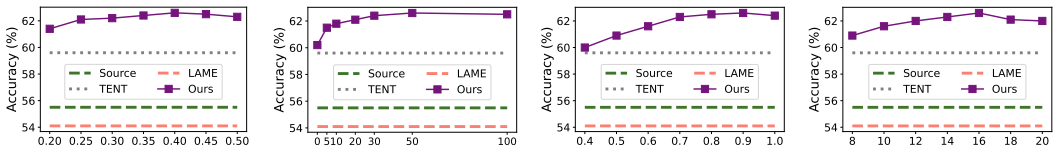
Method	In-Adapt	KL Reg.	Filt.	Out-Adapt	Acc.	Gain
Source	-	-	-	-	55.5	0.0
LAME	-	-	-	PR	54.1	-1.4
ZOO	✓	-	-	-	56.0	+0.5
Exp-1	-	-	-	PH	54.2	-1.3
Exp-2	✓	-	-	PH	51.6	-3.9
Exp-3	✓	✓	-	PH	59.7	+4.3
Exp-4	✓	-	✓	PH	60.2	+4.7
BETA	✓	✓	✓	PH	**62.6**	+7.1

adaptation are insufficient: both LAME's Prediction Refinement and our Prediction Harmonization used in isolation (Exp-1) fail to improve upon the source model, demonstrating that effective black-box TTA requires input adaptation. However, naively adding an input prompt (Exp-2) leads to performance collapse, highlighting the inherent instability of learning a randomly initialized prompt without supervision. Our stabilization techniques address this challenge. Introducing either KL regularization (Exp-3) or sample filtering (Exp-4) provides substantial gains of 4.3 to 4.7 percentage points, and the full BETA framework integrating both techniques achieves the best performance at 62.6%. This confirms that both stabilization mechanisms are essential for robust prompt learning.

Prompt and steering updates. Learning the prompt with a fully frozen steering model improves accuracy from 55.5% to 60.7%; steering adaptation without prompt learning reaches 57.1%. Their combination achieves 62.6%, separating the roles of the two update mechanisms from the stabilizers above (Appx. B.7).

Hyperparameter Sensitivity. We analyze BETA's sensitivity to its key hyperparameters in Fig. 5. Our analysis of the fusion weight α in Eq. 1 shows that the framework's performance is empirically robust, exhibiting stable and high performance across a wide range of values from 0.3 to 0.5 (Fig. 5a). The KL regularization weight λ in Eq. 5 is shown to be a critical component; without it ($\lambda = 0$), performance is suboptimal as the prompt can learn degenerate solutions. As shown in Fig. 5b, performance improves significantly with the introduction of regularization and stabilizes across a broad range of λ values from 20 to 100. For the entropy margin ϵ in Eq. 2, our results show that BETA performs robustly with a more lenient margin (tested from $0.4 \cdot \ln(1000)$ to $1.0 \cdot \ln(1000)$). Unlike methods adapting pre-trained parameters, learning a prompt from random initialization requires more data, making a less restrictive filter beneficial (Fig. 5c). Finally, for the prompt size (Fig. 5d), which corresponds to the frame width, we observe a clear trade-off: smaller prompts may lack the capacity to capture the domain shift, while larger prompts are harder to optimize. The performance peaks around a width of 16 pixels and remains stable across the tested range of 8 to 20.

Effect of Steering Model Choice. We investigate how the local steering model affects BETA's performance (Table 7). Our analysis confirms that BETA consistently improves upon the source model across steering models of different sizes and architectures. Even with a 6M-parameter ViT-Tiny, our method successfully boosts large black-box models. The framework also demonstrates cross-architecture generalization, as a CNN-based ResNet-50 can effectively steer Transformer-based ViT and CLIP models. Notably, BETA's improvement far exceeds the capabilities of the steering models themselves. Even when fully adapted via TENT, our strongest steering model (ViT-Small) attains only 41.0% accuracy, which is significantly lower than the 48.8% black-box baseline. Despite



(a) Fusion Weight α (b) KL Reg. Weight λ (c) Entropy Margin ϵ (d) Prompt Size

Figure 5: Sensitivity analysis of BETA's hyperparameters, showing stable performance across fusion weight α in Eq. 1, regularization weight λ in Eq. 5, entropy margin ϵ in Eq. 2, and prompt size.

this performance gap, BETA leverages these weaker models to steer the target to 54.3% average accuracy. This demonstrates that BETA effectively discovers and transfers beneficial adaptation signals rather than simply relying on the steering model's predictions.

Robustness Analysis. We test challenging streams and restricted API outputs (Table 9). Under label imbalance [5] and continual domain shifts [6], BETA maintains 61.8% and 61.5% accuracy respectively, with minimal degradation from the standard i.i.d. setting (62.6%) (Appx. B.15). BETA also exhibits high robustness to batch size variations, remaining effective even with a batch size of 4 (Appx. B.17). Furthermore, under strict real-time streaming constraints [58], BETA maintains 62.5% accuracy while ZOO drops to 54.3% due to adaptation lag (Appx. B.5). The black-box model remains frozen; adaptation updates the visual prompt and local steering normalization parameters, with consistency regularization stabilizing prompt learning.

Table 9: **BETA retains most gains with top-k API outputs.** ImageNet-C, ViT-B/16.

Setting	Source	Top 1	Top 10	Top 20	Full
Accuracy (%)	55.5	58.1	62.2	62.5	**62.6**

Operating limits. Gains depend on useful steering semantics and capacity. Confidently incorrect target predictions can misguide harmonization. Our evidence covers classification with CNN and Transformer steering models; additional architectures and tasks remain future work (Appx. D).

4 Conclusion

In this work, we addressed the critical challenge of adapting powerful models in the strict black-box setting where only API access is available. We introduced **BETA**, a novel framework that enables efficient and stable Test-Time Adaptation by leveraging a lightweight white-box steering model. The core of our method is a prediction harmonization technique that creates a tractable, shared objective, which is made robust through consistency regularization and a prompt-oriented data filtering strategy. Our extensive experiments show that BETA significantly outperforms existing black-box methods, achieves performance competitive with strong white-box approaches on both Vision and Vision-Language models, and demonstrates immense practical value on a commercial API with a 250$\times$ cost advantage over ZOO-based techniques. By demonstrating that a smaller, local model can effectively steer a powerful, inaccessible one, our work makes robust black-box TTA a practical reality and opens up new possibilities for adapting models in the dark at test time.

Future directions. A key next step is to improve steering reliability when the target is confidently wrong or the local model has weak task-relevant representations. Adaptive harmonization and update selection based on model disagreement are promising directions for these cases. Extending BETA to open-vocabulary recognition, detection, and segmentation would also require explicit class mappings and object- or pixel-level output correspondence.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their insightful comments and suggestions. Yunbei Zhang gratefully acknowledges support from the Lambda Research Grant Program in the form of cloud computing credits.

References

[1] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

[2] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. WILDS: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021.

[3] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020.

[4] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

[5] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023.

[6] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

[7] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *Advances in Neural Information Processing Systems*, 2022.

[8] 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.

[9] 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.

[10] 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, June 2023.

[11] Shuaicheng Niu, Chunyan Miao, Guohao Chen, Pengcheng Wu, and Peilin Zhao. Test-time model adaptation with only forward passes. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=qz1Vx1v9iK.

[12] Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[13] Youngjun Lee, Doyoung Kim, Junhyeok Kang, Jihwan Bang, Hwanjun Song, and Jae-Gil Lee. RA-TTA: Retrieval-augmented test-time adaptation for vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=V3zobHnS61.

[14] Lihua Zhou, Mao Ye, Shuaifeng Li, Nianxin Li, Xiatian Zhu, Lei Deng, Hongbin Liu, and Zhen Lei. Bayesian test-time adaptation for vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29999–30009, 2025.

[15] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8334–8343, 2022. URL https://api.semanticscholar.org/CorpusID:246015836.

[16] Matteo Farina, Gianni Franchi, Giovanni Iacca, Massimiliano Mancini, and Elisa Ricci. Frustratingly easy test-time adaptation of vision-language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=eQ6VjBhevn.

[17] Jin Gao, Jialing Zhang, Xihui Liu, Trevor Darrell, Evan Shelhamer, and Dequan Wang. Back to the source: Diffusion-driven adaptation to test-time corruption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11786–11796, June 2023.

[18] 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.

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

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

[21] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[22] 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*, 11(1):1–18, 2003.

[23] 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.

[24] 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.

[25] Zehao Xiao, Shilin Yan, Jack Hong, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiayi Shen, Qi Wang, and Cees GM Snoek. DynaPrompt: Dynamic test-time prompt tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[26] Zixin Wang, Dong Gong, Sen Wang, Zi Huang, and Yadan Luo. Is less more? exploring token condensation as training-free test-time adaptation. *arXiv preprint arXiv:2410.14729*, 2024.

[27] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2427–2440. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/1415fe9fea0fa1e45dddcff5682239a0-Paper.pdf.

[28] Fan'an Meng, Chaoran Cui, Hongjun Dai, and Shuai Gong. Black-box test-time prompt tuning for vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 6099–6107, 2025.

[29] 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. *Preprints.org*, 2026. doi: 10.20944/preprints202603.1578.v1. Preprint.

[30] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning*, 2022.

[31] Shuaicheng Niu, Guohao Chen, Peilin Zhao, Tianyi Wang, Pengcheng Wu, and Zhiqi Shen. Self-bootstrapping for versatile test-time adaptation. In *Forty-second International Conference on Machine Learning*, 2025.

[32] Guohao Chen, Shuaicheng Niu, Deyu Chen, Jiahao Yang, Zitian Zhang, Mingkui Tan, Pengcheng Wu, and Zhiqi Shen. ZeroSiam: An efficient asymmetry for test-time entropy optimization without collapse. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=x6jHZYhnhL.

[33] Yunbei Zhang, Akshay Mehra, and Jihun Hamm. OT-VP: Optimal transport-guided visual prompting for test-time adaptation. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 1122–1132, February 2025.

[34] 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*, 2025. URL https://openreview.net/forum?id=A6zDim0rQf.

[35] Chandler Timm C Doloriel, Yunbei Zhang, Muhammad Salman Siddiqui, Tor Kristian Stevik, Fadi Al Machot, Kristian Hovde Liland, and Habib Ullah. Continual test-time adaptation via entropy sensitivity-guidance in strict online setting. *arXiv preprint arXiv:2608.29920*, 2026.

[36] Chandler Timm Cagmat Doloriel, Yunbei Zhang, Yeonguk Yu, Taki Hasan Rafi, Tor Kristian Stevik, Fadi Al Machot, Kristian Hovde Liland, Habib Ullah, et al. Family matters: A systematic study of spatial vs. frequency masking for continual test-time adaptation. *Transactions on Machine Learning Research*, 2026.

[37] 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.

[38] Haotian Sun, Yuchen Zhuang, Wei Wei, Chao Zhang, and Bo Dai. BBox-Adapter: Lightweight adapting for black-box large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 47280–47304. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/sun24p.html.

[39] 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.

[40] 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.

[41] Yunbei Zhang, Chengyi Cai, Feng Liu, and Jihun Hamm. Prime once, then reprogram locally: An efficient alternative to black-box service model adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[42] 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.

[43] 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. doi: 10.1109/MSP.2020.3003837.

[44] 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. ISSN 2835-8856. URL https://openreview.net/forum?id=UwtXDttgsE.

[45] Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1214–1223, 2021.

[46] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning*, pages 16805–16827. PMLR, 2022.

[47] 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.

[48] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[49] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019.

[50] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021.

[51] 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 *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[52] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[53] Ce Zhang, Simon Stepputtis, Katia P. Sycara, and Yaqi Xie. Dual prototype evolving for test-time generalization of vision-language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=jsgYYXaSiS.

[54] 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*, 12(7):2217–2226, 2019.

[55] Jeremy Kawahara, Sara Daneshvar, Giuseppe Argenziano, and Ghassan Hamarneh. Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE journal of biomedical and health informatics*, 23(2):538–546, 2018.

[56] Janet Wang, Yunbei Zhang, Zhengming Ding, and Jihun Hamm. Achieving reliable and fair skin lesion diagnosis via unsupervised domain adaptation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5157–5166. IEEE, 2024.

[57] Sheng Zhang, Yanbo Xu, Naoto Usuyama, Hanwen Xu, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, et al. BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. *arXiv preprint arXiv:2303.00915*, 2023.

[58] Motasem Alfarra, Hani Itani, Alejandro Pardo, Shyma Yaser Alhuwaider, Merey Ramazanova, Juan Camilo Perez, Zhipeng Cai, Matthias Müller, and Bernard Ghanem. Evaluation of test-time adaptation under computational time constraints. In *Forty-first International Conference on Machine Learning*, 2024.

[59] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

[60] Marvin Mengxin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=XrGEkCOREX2.

[61] Shuai Zhao, Xiaohan Wang, Linchao Zhu, and Yi Yang. Test-time adaptation with CLIP reward for zero-shot generalization in vision-language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=kIP0duasBb.

[62] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. NOTE: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

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

Appendix

This appendix follows the main paper's progression from access assumptions and experimental protocols to results, efficiency, and ablations. Appendix A details API access, class correspondence, baseline implementations, and hyperparameter selection. Appendix B reports extended results and five-run stability, restricted API outputs, CPU/GPU efficiency, and component analyses. Appendix C gives the analytical derivation and BETA algorithm. Appendix D discusses failure modes and limitations.

A Extended Related Work & Black-Box Setting Analysis

A.1 Detailed Analysis of Model Accessibility and Security Constraints

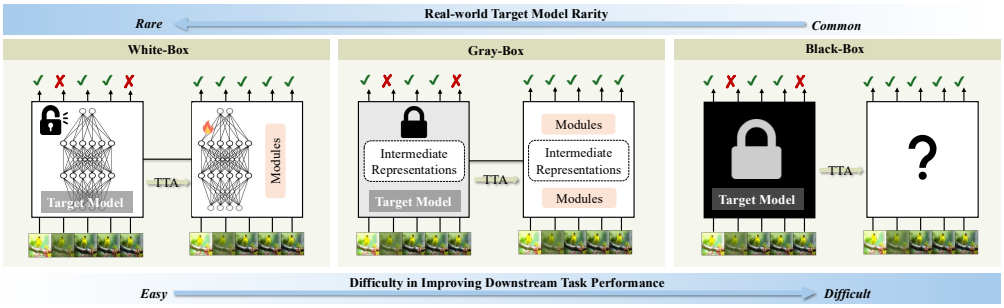


Figure 6: The black-box Test-time Adaptation setting studied in this work. From the client's perspective, the goal is to adapt a powerful server-side API model to a target distribution (e.g., corrupted or domain-shifted images) without any internal access. The client can only send raw input images and receive softmax probability vectors in return. Unlike white-box TTA, no gradients, parameters, intermediate features, or architectural details are available, making this a practical yet challenging scenario for real-world API-based deployment.

In this section, we provide a rigorous definition of the black-box setting adopted in this work. While prior literature often conflates different levels of restricted access, we draw sharp distinctions between access to *raw logits*, *softmax probabilities*, and *hard predictions*. This distinction is critical for evaluating the practical applicability of Test-Time Adaptation (TTA) methods on real-world commercial APIs.

Mathematical Definitions of Output Levels. Let $f_\theta(x)$ denote the pre-trained model. We distinguish between three specific levels of output granularity:

1. *Raw Logits (z):* The pre-activation output vector $z \in \mathbb{R}^C$, where values are unbounded ($-\infty < z_i < \infty$) and unnormalized.
2. *Softmax Probability Vector (p):* The normalized output distribution obtained via the softmax function $\sigma(\cdot)$, such that $p = \sigma(z) \in [0, 1]^C$ with $\sum_i p_i = 1$.
3. *Top-1 Hard Prediction ($\hat{y}$):* A single scalar value representing the class index with the highest confidence, $\hat{y} = \arg\max_i p_i$, often accompanied by a single confidence score.

Real-World API Protocols. To determine the most realistic setting for black-box adaptation, we analyze standard commercial Machine Learning APIs (e.g., OpenAI [8], Clarifai, Google Cloud Vision).

- *Why not Raw Logits?* Access to z is frequently restricted as a security measure. Raw logits contain rich information regarding inter-class relationships ("dark knowledge") that significantly facilitates Model Extraction attacks and Knowledge Distillation [59]. By hiding z, API providers mitigate the risk of model theft.
- *Why Softmax Probabilities?* Most commercial APIs return the probability distribution p rather than a single hard label $\hat{y}$. This is because downstream users typically require confidence estimates to make informed decisions (e.g., thresholding low-confidence predictions).

Justification for BETA's Setting. Based on these protocols, we define the strict *Black-Box* setting as one where the *Softmax Probability Vector* p is available, but *Raw Logits* z are hidden. This setting strikes the balance found in real-world deployments: it provides more information than the restrictive *Label-Only* setting (which only provides $\hat{y}$), enabling unsupervised objectives like entropy minimization ($\mathcal{H}(p) = -\sum p_i \log p_i$). In contrast, we classify methods that require access to raw logits z (e.g., for temperature scaling z/τ or re-normalization [16]) as *Gray-Box*. While these methods do not require gradients, they rely on information often hidden in secure deployment environments.

Operational API Requirements and Class Correspondence. BETA uses an exposed label vocabulary whose class identifiers or names can be matched to the steering model's output coordinates. The steering model need not share the target architecture or have a task-specific trained classifier. Given the target label names, a small public vision-language model such as CLIP-RN50 can construct a zero-shot classifier: its text encoder embeds the names to form classifier weights, without domain-specific training or source data. Appendix B.6 evaluates this construction on EuroSAT and Derm7pt, including a BiomedCLIP target.

For Clarifai, each returned prediction contains a public concept identifier, a class name, and a probability value, for example:

```
{"id": "<concept_id>", "name": "goldfish", "value": 0.72}
```

We match `name` to the ImageNet label dictionary used by the local model and place `value` at the corresponding class index. This fixed correspondence uses neither test labels nor target internals; it involves no label shuffling, score-based reordering, or learned remapping. Partially overlapping or open-vocabulary label spaces would require an additional semantic mapping mechanism.

The pixel-space prompt is applied before submitting the image to the API. Any API-side resizing, normalization, or compression remains unknown and is treated as part of the target mapping. The CLIP and Clarifai evaluations test BETA with distinct preprocessing pipelines. Appendix B.4 further evaluates restricted outputs by retaining and renormalizing only the visible top-k probabilities.

A.2 Extended Baselines Description

We compare BETA against a comprehensive suite of baselines with varying levels of model access, including white-box, gray-box, and black-box methods.

Tent [4] is a **white-box** method for fully test-time adaptation, which adapts a pre-trained model to a new test distribution without requiring any source data. The core idea is to encourage model confidence on the unlabeled test data by minimizing the Shannon entropy of its predictions for each incoming batch. To achieve this efficiently, Tent does not update the entire model; instead, it exclusively adapts the parameters within the model's normalization layers. For each test batch, it first updates the normalization statistics during the forward pass and then optimizes the learnable channel-wise affine transformation parameters via backpropagation on the entropy loss.

SAR [5] is a **white-box** method designed to stabilize online Test-Time Adaptation in challenging "wild" scenarios, such as with mixed domain shifts or small batch sizes, where standard entropy minimization can fail. The method identifies that model collapse during adaptation is often caused by noisy test samples producing large, disruptive gradients. To mitigate this, SAR employs a two-part strategy: it first filters out unreliable, high-entropy samples to reduce noise. For the remaining data, it then uses a sharpness-aware optimizer to guide the model parameters into a flat region of the loss landscape, enhancing robustness against any remaining noisy updates.

Continual Test-Time Adaptation (CoTTA) [6] is a **white-box** method designed to adapt models to continually changing target domains, addressing the challenges of error accumulation and catastrophic forgetting. To generate more reliable pseudo-labels, it employs a teacher-student framework where the student model is updated based on the weight-averaged and augmentation-averaged predictions of the teacher. To prevent catastrophic forgetting over long-term adaptation, CoTTA stochastically restores a small fraction of the student model's weights to their original source-trained values during the update process. The method is designed to adapt all parameters of the network.

Test-Time Template Adjuster (T3A) [27] is a **gray-box** method for domain generalization that adapts a model's final linear classifier at test time. The method is backpropagation-free and works by

first computing class-specific “pseudo-prototype" representations from the features of unlabeled test data. Once these prototypes are established, it classifies each new test sample based on its distance to these dynamically adjusted prototypes. This allows the model to leverage information from the target domain without requiring extensive optimization or altering the core feature extractor.

Forward-Optimization Adaptation (FOA) [11] is a **gray-box** method designed for test-time adaptation in scenarios where backpropagation is infeasible, such as on quantized models or edge devices. The approach is entirely training-free and avoids modifying model weights by learning an additive input prompt using a derivative-free optimizer (CMA-ES). To guide this optimization, FOA introduces a novel fitness function that combines prediction entropy with a term measuring the statistical discrepancy between the test sample's activations and pre-computed source data activations. The framework also includes a “back-to-source" activation shifting scheme that directly modifies the final layer's features during the forward pass to better align them with the source domain.

LAME [15] is a **gray-box** method for online test-time adaptation that operates without requiring access to model parameters or gradients. Instead of adapting the network's weights, it adapts the model's output probabilities directly for a given batch of test data. The method proposes a Laplacian Adjusted Maximum-likelihood Estimation (LAME) objective, which finds the optimal latent class assignments by maximizing the data likelihood while being regularized by a Laplacian term that encourages label consistency among neighboring samples in the feature space. This objective is optimized efficiently using a concave-convex procedure and does not require backpropagation.

LAME and TENT Implementation Protocols. We use the LAME and TENT implementations and evaluation procedures released in the official FOA codebase [11]. LAME constructs its affinity graph from target features, so it is a stronger-access reference rather than a method satisfying our strict probability-only API constraints. With this protocol, LAME obtains 54.1% on ImageNet-C with ViT-B/16, compared with 55.5% for Source. Under severe corruption, unreliable feature neighborhoods may propagate locally consistent but incorrect predictions during Laplacian refinement. Our TENT results also match those reported by FOA, including low accuracies on individual severity-5 corruptions associated with unstable online entropy minimization.

In contrast to the methods above, the following baselines are designed specifically for the adaptation of Vision-Language Models:

Test-Time Prompt Tuning (TPT) [7] is a **white-box** method that adapts Vision-Language Models like CLIP using only a single unlabeled test sample. For each test image, TPT creates multiple augmented views and optimizes a learnable text prompt via backpropagation to enforce prediction consistency across them. The optimization is guided by minimizing the entropy of the averaged predictions, and a confidence selection module filters out noisy augmentations that yield low-confidence outputs. TPT performs a one-step update on the prompt for each test sample.

Dual Prototype Evolving (DPE) [53] is a **white-box** method that performs test-time adaptation for VLMs by accumulating task-specific knowledge from both visual and textual modalities. The method maintains and evolves two sets of class prototypes—one textual and one visual—which are updated online as more test samples are processed. For each individual test sample, DPE learns temporary residual parameters to adjust both sets of prototypes. This sample-specific optimization is guided by a dual objective that encourages prediction consistency across augmented views and enforces alignment between the textual and visual prototypes for each class.

DynaPrompt [25] is a **white-box** method that improves online test-time prompt tuning by leveraging information from previous test samples while mitigating the problem of prompt collapse. The core of the method is an online prompt buffer containing a set of learnable prompts that evolve over time. For each new test sample, DynaPrompt employs a dynamic selection strategy based on prediction entropy and probability difference to choose a relevant subset of prompts from the buffer for optimization. To adapt to new data, the framework also dynamically appends new prompts to the buffer and removes inactive ones.

B^{2}TPT [28] is a **gray-box** method that addresses test-time prompt tuning for black-box Vision-Language Models (VLMs) where gradients are inaccessible. To overcome this, it employs a derivative-free algorithm (CMA-ES) to optimize low-dimensional "intrinsic prompts," which are then projected into the full prompt space to make the high-dimensional optimization tractable. For supervision, the framework uses a “Consistent or Confident" (CoC) pseudo-labeling strategy to generate labels

from the model's outputs. The method jointly optimizes text and vision prompts using a frozen CLIP ViT-B/16 backbone.

Training-free Dynamic Adapter (TDA) [12] is a **gray-box** method designed for efficient test-time adaptation of Vision-Language Models without requiring backpropagation. The method constructs a lightweight key-value cache during inference, which is progressively updated with incoming test samples. This cache consists of two components: a positive cache that stores image features and their corresponding high-confidence pseudo-labels, and a novel negative cache that stores negative pseudo-labels to improve robustness against label noise. The final prediction is a combination of the original CLIP output and the predictions derived from both the positive and negative caches.

Retrieval-Augmented TTA (RA-TTA) [13] is a **gray-box** method that adapts Vision-Language Models by incorporating external knowledge from a large image database at test time. Instead of a direct image-to-image search, RA-TTA uses a novel description-based retrieval process to find more relevant external images. For a given test image, it first identifies its most prominent visual features by selecting matching fine-grained text descriptions from a pre-compiled library. These selected text descriptions are then used as queries to retrieve semantically similar images from the database, and the VLM's initial prediction is refined using a relevance score derived from this external knowledge.

Bayesian Class Adaptation (BCA) [14] is a **gray-box** method that adapts Vision-Language Models by updating both the class likelihood and prior at test time. It frames the adaptation problem using Bayes' theorem, identifying that existing methods only adapt the likelihood (class embeddings) while overlooking the class prior, which can shift in new domains. BCA employs a dual-update mechanism: it adapts the likelihood by updating the most relevant class embedding with an incoming visual feature via a running average. Concurrently, it adapts the prior by using the model's posterior prediction for the current sample to update the prior distribution of the predicted class, allowing the model to learn the new class frequencies on the fly.

Token Condensation as Adaptation (TCA) [26] is a **gray-box** method that provides an efficient, training-free solution for test-time adaptation in Vision-Language Models. The method uniquely repurposes token condensation, a technique originally for improving ViT efficiency, as an adaptation mechanism. It introduces a domain-aware token reservoir that stores reliable class tokens from past test samples to serve as domain anchors. These anchors guide both a cross-head token condensation process, which prunes irrelevant visual tokens, and a logits self-correction mechanism that refines the model's final prediction.

A.3 Hyperparameter Selection Protocol

All hyperparameters are selected using four validation corruptions disjoint from the 15 test corruptions: Speckle Noise, Gaussian Blur, Spatter, and Saturate, following MEMO [60]. After selection, the settings are fixed across test corruptions, datasets, target architectures, and random seeds, without tuning on incoming test data. The sensitivity analyses use the same validation protocol.

The default harmonization weight is $\alpha = 0.4$, and the consistency weight is $\lambda = 50$. The prompt uses AdamW with learning rate 0.01, while steering normalization parameters use SGD with learning rate 2×10^{-5}. For a K-class task, the entropy threshold is $\epsilon = 0.9 \log K$: its class-count scaling follows EATA [30], and the coefficient 0.9 is selected once on the same validation corruptions. Thus, changing K changes only the prescribed threshold scale, without test-dependent retuning. Appendix B.9 compares the fixed default with a jointly learned α.

B Comprehensive Quantitative Analysis

B.1 Five-Run Stability on ImageNet-C

We retain the run reported in Table 2 and add four independent random-seed runs with the same ViT-B/16 target, ViT-S/16 steering model, and fixed hyperparameters. Table 10 reports accuracy at severity 5 over all 15 corruptions and on the challenging Contrast domain. The means and standard deviations are computed from unrounded accuracies; individual runs are displayed to one decimal place. The small variation across runs supports the stability of the complete method. The main table retains the original single-run result of 62.6%, while the five-run mean is 62.5%.

Table 10: BETA is stable across five independent runs on ImageNet-C at severity 5. All entries are accuracy (%); $\pm$ denotes standard deviation across runs.

Evaluation	Run 1	Run 2	Run 3	Run 4	Run 5	Mean $\pm$ SD
15-corruption average	62.6	62.5	62.5	62.4	62.5	62.5 ± 0.06
Contrast	53.4	53.2	53.0	53.0	53.1	53.1 ± 0.16

B.2 Additional Results on ImageNet Variants and EuroSAT

To provide a comprehensive evaluation, we extend our comparisons to include augmentation-based strategies and recent methods tailored for Vision-Language Models (VLMs). Specifically, we compare BETA against **ZERO** [16], a test-time augmentation method that optimizes temperature using input augmentations. We note that while ZERO requires access to raw logits—violating strict black-box API constraints that typically only provide probabilities—we grant it this access for a rigorous upper-bound comparison. We evaluate both the standard ZERO (64 calls/image) and **ZERO_ensemble** (448 calls/image, using 7 text templates). We also include **B^{2}TPT** [28], a recent gray-box prompt tuning method for VLMs.

Table 11: Performance comparison on ImageNet Variants with CLIP-B/16. BETA outperforms strong augmentation-based and gray-box baselines while requiring only a single API call per image.

Method	IN-S	IN-R	IN-A	IN-v2	IN	Avg.	Gain	#API
Source	46.1	74.0	47.9	60.9	66.7	59.1	–	1
LAME	45.4	72.8	48.1	61.6	66.7	58.9	-0.2	1
ZOO-SPSA-GC	46.0	72.8	50.2	61.5	65.8	59.3	+0.1	16
B^{2}TPT (w/ tokens)	49.5	78.6	55.3	65.4	69.6	63.7	+4.6	120
ZERO (w/ logits)	48.4	77.2	59.6	64.2	69.3	63.7	+4.6	64
ZERO_ens (w/ logits)	50.6	80.8	62.8	65.2	71.2	66.1	+7.0	448
BETA (Ours)	**50.9**	**76.0**	**62.8**	**65.1**	**77.5**	**66.5**	**+7.4**	**1**

Table 12: Performance on the fine-grained EuroSAT dataset with CLIP-B/16. BETA achieves significant gains (+11.3%) with high efficiency.

Method	Acc (%)	Gain	#API
Source	42.0	–	1
B^{2}TPT (w/ tokens)	46.8	+4.8	120
ZERO (w/ logits)	39.6	-2.4	64
ZERO_ensemble (w/ logits)	43.8	+1.8	448
BETA (Ours)	**53.3**	**+11.3**	**1**

Access-level framing. We do not claim blanket accuracy superiority over gray-box methods such as B^{2}TPT, BCA, TDA, and RA-TTA. These methods outperform BETA on several individual splits in Table 4 because they have access to CLIP-internal tokens, features, or gradients that a black-box API does not expose. Our contribution is that BETA is competitive with, and on average above, these methods under a strictly tighter access level, namely input images and softmax probabilities only, and that BETA applies uniformly to visual models and vision-language models, while every gray-box VLM baseline here is CLIP-specific.

Five-dataset average on CLIP ViT-B/16. Averaged over ImageNet-V2, ImageNet-A, ImageNet-S, ImageNet-R, and ImageNet, BETA reaches 66.5%, above gray-box B^{2}TPT (63.7%), TDA (65.0%), RA-TTA (64.9%), and BCA (65.8%).

Classification of B^{2}TPT as Gray-Box. We categorize B^{2}TPT as a gray-box method because it operates by modifying inputs in the embedding space. Specifically, it prepends learnable vectors directly to the text and image embeddings (e_t and e_v), requiring internal access to the model's intermediate feature representations. This contrasts with the strict black-box setting of commercial APIs, which accept only raw image or text inputs. Furthermore, its underlying optimization (CMA-ES) is query-intensive, requiring approximately 120 API calls per input.

Results on ImageNet Variants and EuroSAT. We evaluate these baselines on the full suite of ImageNet variants (ImageNet-S, R, A, v2, and standard ImageNet) and the challenging fine-grained EuroSAT dataset. The results are summarized in Table 11 and Table 12.

BETA consistently outperforms these query-intensive baselines while maintaining strict API efficiency. On the ImageNet variants (Table 11), BETA achieves the highest average accuracy of 66.5%, surpassing the ensemble version of ZERO (66.1%) which requires 448 API calls per image. The efficiency gap is even more pronounced on EuroSAT (Table 12), where BETA achieves a substantial gain of +11.3% over the source model with a single API call, whereas augmentation baselines struggle or yield marginal gains despite their high computational cost. This demonstrates that BETA's effectiveness stems from learned adaptation rather than simple data augmentation, making it a far more practical solution for real-world deployment where API costs and rate limits are critical constraints.

B.3 Full Clarifai Evaluation Across All 15 Corruptions

The main body reports Clarifai results on the ImageNet-C Contrast domain, where BETA improves upon Source by 5.2 percentage points after roughly 120 samples at a cost of \$0.4. ZOO requires over \$100 to reach comparable accuracy, giving BETA a $250\times$ cost advantage. To show that this is not a single-domain artifact, we extend the evaluation to all 15 ImageNet-C corruptions under a fixed \$5 per-domain budget. Table 13 reports the average.

Table 13: Real commercial Clarifai API, all 15 ImageNet-C corruptions at severity 5, \$5 budget per domain. ZOO is infeasible at this budget and is omitted.

Method	Source	LAME	**BETA**
Avg. Acc. (%)	44.3	43.2	**54.7**

The $+10.4\%$ average gain on a truly opaque commercial API is consistent with our ViT-B/16 and CLIP experiments, validating BETA in deployment rather than in simulation alone.

Five-Run Stability on the Commercial API. We additionally evaluate five independent runs on Contrast, using the same \$5 budget per run and otherwise identical settings. Table 14 reports a mean accuracy of $46.7 \pm 0.57\%$, compared with 38.6% for Source. This experiment measures run-to-run variation on one corruption; Table 13 instead averages over all 15 corruptions at \$5 per domain. The class correspondence used for the API is described in Appendix A.1.

Table 14: BETA consistently improves Clarifai on Contrast under a \$5 budget per run. Accuracy (%) is reported for five independent runs; $\pm$ denotes their standard deviation.

Source	Run 1	Run 2	Run 3	Run 4	Run 5	Mean $\pm$ SD
38.6	46.9	47.2	46.1	46.0	47.1	46.7 ± 0.57

B.4 Adaptation with Top-k API Outputs

We restrict the ViT-B/16 target on ImageNet-C to return only its top-k classes and probabilities. We renormalize the visible probabilities and leave the target and steering models, optimization procedure, and hyperparameters unchanged. Table 15 shows that top-10 access recovers 62.2% accuracy, close to 62.6% with the full vector. Top-1 access retains a 2.6-point gain over Source, although it supplies less uncertainty information. These results support adaptation under truncated API outputs without changing the default configuration.

The visible class identifiers retain the same correspondence to the steering vocabulary described in Appendix A.1. In the reported ImageNet evaluation, the top-10 and top-20 predictions frequently contain more than 0.9 cumulative probability mass; top-20 mass exceeds 0.99 for 99.99% of samples. This concentration helps explain why retaining a small set of outputs preserves most of the adaptation gain.

Table 15: Truncated API outputs retain most of BETA's gain on ImageNet-C. The target is ViT-B/16 and the steering model is ViT-S/16; gains are percentage points over Source.

Available API output	Accuracy (%)	Gain
Source (no adaptation)	55.5	–
Top 1	58.1	+2.6
Top 10	62.2	+6.7
Top 20	62.5	+7.0
Full probability vector	62.6	+7.1

B.5 Computational Efficiency and Real-Time Adaptation

To comprehensively assess the practicality of BETA, we analyze efficiency across two dimensions: API costs (query complexity) and local computational overhead. We further validate performance under a strict real-time streaming protocol, following [58].

Detailed Efficiency Breakdown. We conducted a granular breakdown of wall-clock latency and resource usage using a single NVIDIA RTX 3090 GPU. As summarized in Table 5, we compare BETA against baselines including ZOO-SPSA-GC† and Test-Time Augmentation (TT-Aug) [45].

In this GPU-based setup, the analysis yields two critical insights. First, **local computation is negligible** compared to API latency. While BETA introduces a local steering model (ViT-Small), it requires only 2.6GB of GPU memory—feasible for consumer-grade hardware—and adds a trivial 0.003s overhead per image for the backward pass. The primary bottleneck in black-box adaptation is the API forward pass ($T_{API} \approx 0.045s$), which is dominated by network latency. Second, **API calls dominate total latency**. Methods relying on multiple queries per image suffer from severe slowdowns. ZOO (16 calls) and TT-Aug (64 calls) are approximately $9.4\times$ ($0.450s$) and $37.5\times$ ($1.800s$) slower than BETA per image, respectively. This clarifies the context for "backpropagation-free" approaches in this setting: eliminating the local backward pass ($0.003s$) provides no practical speed benefit when the total time is dictated by the mandatory API call ($0.045s$).

Computationally Constrained Evaluation. To further rigorously test feasibility in streaming scenarios, we adopt the *Realistic Evaluation Protocol* from [58]. This protocol penalizes methods that cannot keep pace with a data stream arriving at the API's maximum throughput speed ($r = 1 \text{ img}/T_{API}$).

We define the relative adaptation cost based on the total processing time per step: $T_{Step} = \max(T_{API}, T_{Local_Fwd}) + T_{Local_Bwd}$. Crucially, BETA allows for the parallelization of the local steering model's forward pass with the API query latency. In the measured GPU setup, $T_{API} \gg T_{Local_Fwd}$, so the local forward cost is effectively hidden, leaving only the negligible backward pass. Consequently, BETA maintains a relative cost $\mathcal{C} \approx 1$, allowing it to adapt to virtually 100% of the data stream. In contrast, query-intensive methods like ZOO incur massive adaptation lag ($\mathcal{C} \gg 1$), forcing them to skip adaptation for the majority of samples to maintain throughput.

The results in Table 16 demonstrate the impact of this constraint. Under strict real-time conditions, ZOO's performance drops to 54.3% (worse than the Source), as it updates too infrequently. BETA, however, maintains an accuracy of 62.5%, confirming it is a viable solution for real-time black-box adaptation.

Table 16: Evaluation under computational time constraints. "Offline" assumes unlimited time; "Online" simulates realistic streaming where slow methods skip samples.

Method	**Offline Acc (%)**	**Online Acc (%)**
Source	55.5	55.5
LAME	54.1	54.1
ZOO	56.0	54.3
BETA (Ours)	**62.6**	**62.5**

CPU-Only Client-Side Profiling. We profile the complete local BETA update on an AMD Ryzen Threadripper PRO 5955WX using eight CPU threads (Table 17). Measurements include prompt construction, steering forward passes, backpropagation, and optimizer updates; target API execution and network latency are excluded. Peak memory is process memory, rather than GPU memory. BETA can therefore run without a local GPU, with ViT-T/16 taking 24.0 ms/image at batch size 8. For

the accuracy trade-off under the standard evaluation protocol, Table 7 reports 58.2% with ViT-T/16 and 62.6% with ViT-S/16. The CPU timings characterize local cost; total deployment latency also depends on the API and available overlap between local computation and remote inference.

Table 17: CPU execution enables BETA without a local GPU. The complete client-side update is profiled on eight CPU threads; API and network latency are excluded.

Steering model	Parameters	Batch size	ms/image	Peak memory (GB)
ViT-T/16	5.7M	1	48.0	0.70
ViT-T/16	5.7M	8	24.0	1.09
ViT-S/16	22.1M	1	89.0	0.82
ViT-S/16	22.1M	8	61.6	1.53

B.6 Sensitivity to the Zero-shot Steering Model

The zero-shot steering recipe in Sec. 2.4 requires picking a public CLIP whose text encoder supplies the label-name embeddings. We evaluate two natural choices: CLIP-RN50 (the smallest standard variant) and CLIP ViT-B/32 (slightly larger, still small relative to typical target VLMs). Table 18 reports both across three settings, including the medical Derm7pt benchmark with two distinct black-box targets (a generic CLIP and a domain-specialized BiomedCLIP). Both CLIP variants deliver gains over Source, LAME, and TT-Aug in all three rows; CLIP ViT-B/32 produces marginally larger gains on average, so the main paper reports that variant. Importantly, even CLIP-RN50, the weaker zero-shot classifier on its own, still improves the black-box target, which supports the design claim that f_S only needs to supply a usable gradient signal rather than strong predictions on the target task.

Table 18: Zero-shot steering recipe: sensitivity to the choice of public CLIP. Both CLIP-RN50 and CLIP ViT-B/32 use text embeddings of the target label names as classifier weights over their visual features. Both deliver consistent gains on medical Derm7pt (with CLIP and BiomedCLIP targets) and remote-sensing EuroSAT. Main-text numbers use CLIP ViT-B/32.

Dataset	**Black-box Target**	Source	LAME	TT-Aug	**BETA (zero-shot f_S)**	
					CLIP-RN50	CLIP ViT-B/32
Derm7pt	CLIP ViT-B/16	55.9	56.0	56.6	57.1	**58.6**
	BiomedCLIP	60.9	60.4	61.0	61.2	**62.1**
EuroSAT	CLIP ViT-B/16	42.0	40.9	39.6	50.1	**53.3**

B.7 Isolating Prompt Learning and Steering Adaptation

We isolate the two update mechanisms using the same ViT-B/16 target, ViT-S/16 steering model, ImageNet-C evaluation protocol, and fixed hyperparameters. Table 19 compares target-only inference, fixed posterior harmonization with both models frozen, and the separate or joint adaptation of the prompt and steering model. Prompt learning with a fully frozen steering model improves accuracy from 55.5% to 60.7%; jointly updating the steering model's normalization parameters further improves it to 62.6%. Steering adaptation without prompt learning reaches 57.1%. The two mechanisms thus make complementary contributions, while fixed posterior harmonization alone does not explain the gain.

Table 19: Prompt learning and steering adaptation contribute complementary gains on ImageNet-C. The target model remains frozen in every configuration.

Configuration	Accuracy (%)
Target prediction only	55.5
Fixed harmonization; prompt and steering frozen	54.2
Prompt updated; steering fully frozen	60.7
Steering updated; no prompt update	57.1
Prompt and steering jointly updated (BETA)	**62.6**

The progressive component ablation in Table 8 addresses a different question: how KL regularization and filtering stabilize prompt learning. These components act on prompt updates, so their effects are

evaluated with input adaptation enabled. Appendix B.8 separately compares harmonization during optimization with inference-time ensembling.

B.8 Harmonization vs. Inference-Time Ensembling

A natural alternative to prediction harmonization is to adapt the steering model on its own and then ensemble its output with the black-box output at inference. We compare two such variants on ImageNet-C with ViT-B/16 as the target.

- **Adapt+Ensemble.** Update only the steering model's normalization parameters with $\mathcal{H}(p_S)$ and $\mathcal{L}_{\text{consist}}$, then ensemble with the black-box model at inference.
- **Prompt+Ensemble.** Additionally learn δ using the steering-only entropy objective, then ensemble.

Table 20: Harmonization vs. inference-time ensembling of the adapted steering model with the black-box model on ImageNet-C, ViT-B/16 target.

Setting	Local-only	Target-only	Adapt+Ens.	Prompt+Ens.	**BETA**
Avg. Acc. (%)	39.5	55.5	57.1	54.7	**62.6**

Prompt+Ensemble underperforms the target-only baseline ($54.7 < 55.5$): a prompt optimized for p_S alone can make the steering model confident on a class the target rejects, and averaging the two outputs at inference does not undo this effect. Adapt+Ensemble is capped by the weak local model, which contributes little informative probability mass to the mixture. BETA avoids both failure modes, because the JS term in Eq. (9) requires the two models to agree before the prompt is updated. The resulting accuracy is substantially higher at 62.6%.

B.9 Fixed versus Learnable Harmonization Weight

We compare the validation-selected default $\alpha = 0.4$ with a learnable weight initialized at 0.4 and jointly optimized during adaptation. With ViT-B/16 as the target and ViT-S/16 as the steering model on ImageNet-C, the learnable weight ends at 0.43 and achieves 62.7% accuracy (Table 21). The 0.1-point improvement over fixed α is small, so we retain the fixed value for a simpler adaptation procedure. Its selection uses only the disjoint validation corruptions described in Appendix A.3.

Table 21: Learning α changes ImageNet-C accuracy by only 0.1 percentage point relative to the fixed default.

Strategy	Initial α	Final α	Accuracy (%)
Fixed	0.40	0.40	62.6
Learnable	0.40	0.43	62.7

B.10 Adaptation Performance of the Steering Model

To demonstrate that BETA's improvement is non-trivial and not simply a result of relying on the steering model's outputs, we present the white-box and gray-box adaptation performance of the ViT-Small steering model in Table 22. There exists a substantial performance gap between the pre-trained steering model (39.5% accuracy on ImageNet-C) and the target black-box models (e.g., ViT-L/16 at 61.1% accuracy). Even when the steering model itself is fully adapted in a white-box setting with a strong method like SAR, its performance is capped at 57.4%. This is still well below the starting accuracy of the black-box model it is meant to guide. This highlights that BETA successfully leverages this weaker, suboptimal steering model not for its direct predictions, but to discover and transfer beneficial adaptation signals to the far more powerful black-box model without requiring any internal access.

B.11 Comparison with Test-Time Knowledge Distillation

A natural question arises as to whether BETA's improvements stem from simply distilling the powerful black-box model's knowledge into the local steering model. To investigate this, and to verify that

Table 22: White-box and gray-box TTA performance on the ViT-Small steering model (ImageNet-C). Even when fully adapted, the steering model's performance is capped well below that of the unadapted black-box target models (ViT-B/16: 55.5%, ViT-L/16: 61.1%).

	Source	TENT	T3A	SAR	CoTTA	LAME
Avg. Acc (%)	39.5	51.9	40.4	57.4	46.0	38.9
Gain (%)	0.0	+12.4	+0.9	+17.9	+6.5	-0.6

our framework is not merely performing Test-Time Knowledge Distillation (KD), we implemented a KD baseline following the protocol in [61]. Specifically, we employed the black-box ViT-B/16 as the teacher and the local ViT-S/16 as the student, optimizing the student to match the teacher's predictions on the target data.

The results, summarized in Table 23, reveal a fundamental distinction between the two approaches. Standard distillation is inherently limited by the capacity of the student model; the distilled ViT-S/16 achieves only 50.3% accuracy, failing to even match the original performance of the black-box teacher (55.5%). This result is expected, as KD aims to mimic the teacher's existing boundary rather than adapt it to the new domain.

In sharp contrast, BETA achieves 62.6% accuracy, significantly surpassing the original black-box model. This confirms that BETA is not a distillation process where a student mimics a fixed teacher. Instead, BETA utilizes the local model to actively *adapt* the input prompts for the black-box model, allowing the final system to break through the performance ceiling of the original pre-trained weights.

Table 23: Comparison between Test-Time Knowledge Distillation and BETA on ImageNet-C. While KD is upper-bounded by the teacher's performance, BETA successfully adapts the black-box model beyond its original baseline.

Model Role	**Architecture**	**Method**	**Avg. Acc (%)**
Local Steering	ViT-S/16	Source	39.5
		TENT	51.9
		KD (from ViT-B/16)	50.3
Black-Box Target	ViT-B/16	Source	55.5
		BETA (Ours)	**62.6**

B.12 Zeroth-Order Optimization Baselines

As a direct approach to adapting the visual prompt δ in a black-box setting, we evaluate several Zeroth-Order Optimization (ZOO) baselines. These derivative-free methods optimize the prompt by minimizing a fitness function, which we define as the Shannon entropy of the black-box model's predictions on the prompted input, $f(\delta) = \mathcal{H}(p_B(x + \delta))$. For a fair comparison, we configure all three ZOO methods to use 16 queries per test sample for their optimization process.

CMA-ES. As a representative ZOO method, **Covariance Matrix Adaptation Evolution Strategy (CMA-ES)** is a derivative-free algorithm used to optimize a high-dimensional visual prompt where gradients are inaccessible [21, 22, 11, 28]. In each iteration, CMA-ES samples a population of candidate prompts from a multivariate normal distribution and evaluates them using the fitness function. The goal is to find a prompt, δ, that minimizes this entropy, encouraging high-confidence predictions. Based on the performance of the sampled prompts, CMA-ES updates the mean and covariance matrix of the sampling distribution to guide the search towards more promising regions of the solution space.

RGF Random Gradient-Free (RGF) is a ZOO method that estimates the gradient of the fitness function by sampling multiple random directions from a standard Gaussian distribution [18, 39]. For a given visual prompt δ, RGF approximates the gradient by averaging the function's response to small perturbations along these random directions, allowing it to descend the loss landscape without direct gradient calculations. The gradient approximation at iteration t is computed as:

$$g_t(\delta_t) = \frac{1}{q} \sum_{i=1}^{q} \frac{f(\delta_t + \mu u_i) - f(\delta_t)}{\mu} u_i \tag{6}$$

where u_i is a random direction vector drawn from $\mathcal{N}(0, I)$, μ is a small smoothing parameter, and q is the number of directions sampled.

SPSA with Gradient Correction (SPSA-GC) To optimize the visual prompt under black-box constraints, we adopt the Simultaneous Perturbation Stochastic Approximation with Gradient Correction (SPSA-GC) algorithm, as utilized in BlackVIP [10]. SPSA is a highly efficient ZOO algorithm that estimates the gradient using only two queries per iteration [19]. Unlike RGF, which requires sampling multiple directions, SPSA perturbs the parameters in a single random direction and its opposite. The gradient approximation at iteration t for a visual prompt δ_t is computed as:

$$\hat{g}_t(\delta_t) = \frac{f(\delta_t + \mu\Delta_t) - f(\delta_t - \mu\Delta_t)}{2\mu}\Delta_t \tag{7}$$

where Δ_t is a random perturbation vector drawn from a Bernoulli distribution, and μ is a small step size.

While standard SPSA is query-efficient, the stochastic gradient estimate $\hat{g}_t$ can be noisy. To mitigate this, we employ the Gradient Correction mechanism proposed in BlackVIP [10]. This method integrates Nesterov's Accelerated Gradient (NAG) into the update rule, using a momentum accumulator to rectify the estimated gradient direction. By smoothing the optimization trajectory, SPSA-GC significantly enhances stability compared to vanilla SPSA, making it particularly suitable for the high-dimensional optimization of visual prompts.

B.13 API Efficiency across Adaptation Methods

Table 24 demonstrates BETA's superior efficiency compared to existing black-box TTA methods. While ZOO-based approaches (CMA, RGF, SPSA) require 16 API calls per test sample and achieve modest or negative performance gains ranging from -1.0% to +0.5%, BETA achieves a substantial +7.1% improvement with only a single API call per sample. This represents a 16× reduction in API usage while delivering significantly better adaptation performance. The gray-box LAME reference, though equally efficient with one API call, suffers from limited adaptive capacity due to its post-hoc output refinement approach, resulting in a -1.4% performance drop. These results highlight BETA's unique combination of query efficiency and adaptation effectiveness in the black-box setting.

Table 24: API efficiency comparison, with LAME included as a gray-box reference. BETA achieves the best accuracy-efficiency trade-off with a single API call per sample.

Method	#API/sample	Acc (%)	Gain
Source (Inference)	1	55.5	0.0
LAME	1	54.1	-1.4
ZOO-CMA	16	54.5	-1.0
ZOO-RGF	16	56.0	+0.5
ZOO-SPSA-GC	16	55.1	-0.4
TT-Aug	64	55.6	+0.1
DDA	2	56.9	+1.4
BETA (Ours)	**1**	**62.6**	**+7.1**

B.14 Orthogonality of Contribution: Unsupervised Objective vs. ZOO Algorithms

Our ZOO baselines include SPSA-GC [10]; its choice should be distinguished from the challenges inherent to the *adaptation objective*. The efficacy of SPSA-GC was originally demonstrated in BlackVIP [10] within a *supervised* few-shot transfer setting, where ground-truth labels guide optimization through a cross-entropy loss.

In contrast, our strictly **unsupervised online setting** relies on objectives such as entropy minimization. We observe that replacing the supervised loss with an unsupervised one fundamentally alters the optimization landscape, making it prone to trivial solutions. As evidenced in our experimental results, naively applying even a robust ZOO algorithm like SPSA-GC to this unsupervised objective leads to prompt collapse, where the model exploits high-frequency patterns to minimize entropy without preserving semantic integrity. Therefore, we clarify that our primary contribution does not lie in the ZOO algorithm itself. Rather, our contribution is the **unsupervised stabilization framework**: comprising Prediction Harmonization, Prompt Learning-Oriented Data Filtering, and Consistency Regularization. These mechanisms effectively constrain the optimization space, preventing the instability inherent to source-free black-box adaptation and enabling effective Test-Time Adaptation.

Table 25: Robustness analysis on ImageNet-C (ViT-B/16) under Label Imbalance and Continual Domain Shift settings. BETA demonstrates minimal degradation compared to the standard setting, highlighting its stability in dynamic environments.

Method	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Bright.	Contr.	Elastic	Pixel.	JPEG	Avg.
Source	56.8	56.8	57.5	46.9	35.6	53.1	44.8	62.2	62.5	65.7	77.7	32.6	46.0	67.0	67.6	55.5
BETA (Standard)	60.5	60.7	61.1	54.5	52.2	59.9	56.3	63.6	64.7	66.1	78.1	53.4	62.1	73.3	72.0	62.6
BETA (**Label Imbalance**)	59.0	59.9	59.5	53.9	51.1	59.1	55.5	62.9	64.3	65.4	77.9	52.4	61.2	73.1	72.1	61.8
BETA (**Continual Shifts**)	59.5	61.0	60.4	52.3	51.4	58.4	55.2	61.8	63.3	63.8	77.4	51.8	61.7	72.5	71.3	61.5

B.15 Robustness to Label Imbalance and Continual Shifts

While our primary evaluation follows the standard episodic adaptation setting, real-world data streams often exhibit temporal correlations or non-stationary distributions. To validate the stability of BETA in dynamic environments, we extend our evaluation on ImageNet-C (using ViT-B/16) to include two challenging scenarios:

- **Label Imbalance** [5, 62]: Following the protocol established in SAR [5], we evaluate performance on data streams with highly skewed class distributions within each batch, simulating non-i.i.d. test streams.
- **Continual Domain Shifts** [6, 30]: Following the Continual Test-Time Adaptation (CoTTA) setting [6], the model adapts to the 15 corruption domains of ImageNet-C sequentially without resetting the model state between domains.

The results are summarized in Table 25. BETA exhibits remarkable stability, maintaining high performance even under these challenging conditions. In the label imbalance setting, BETA achieves an average accuracy of 61.8%, and under continual shifts, it maintains 61.5%. This represents minimal degradation compared to the standard i.i.d. setting (62.6%).

Why is BETA robust? This robustness is intuitive given our framework's design. Unlike white-box methods that directly update internal model parameters—a process known to risk catastrophic forgetting or overfitting to biased batches—BETA keeps the parameters of the black-box target model frozen. We exclusively learn an additive input prompt. Furthermore, the local steering model is updated with a conservative learning rate and strong consistency regularization, preventing the optimization trajectory from over-fitting to the dynamic changes or local biases in the data stream. This makes BETA naturally resilient to the instability often observed in dynamic test-time adaptation.

B.16 Analysis on Stabilization Mechanisms

We conduct a component analysis to demonstrate the importance of our two stabilization mechanisms, visualizing the online batch accuracy on the challenging ImageNet-C Contrast domain. The figure shows that the full BETA framework ("Ours") rapidly achieves high accuracy and maintains stable performance across all 800 online batches. In contrast, removing the data filtering component ("w/o Data Filtering") results in significantly lower and gradually decaying performance. More critically, removing the consistency regularization ("w/o KL Reg.") leads to catastrophic collapse, with the model's accuracy plummeting to near zero after approximately 400 batches. This analysis empirically validates that both the consistency regularization and the data filtering are essential for the stable and effective performance of BETA.

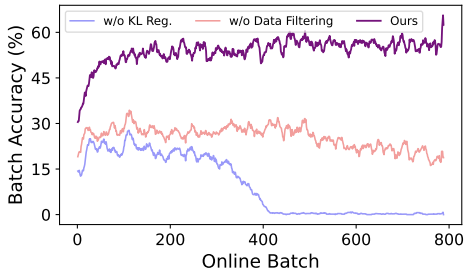


Figure 7: Online Batch Accuracy on ImageNet-C Contrast domain.

B.17 Robustness to Batch Size

In practical online deployment, the number of samples available for adaptation at any given time step can vary significantly. To assess BETA's sensitivity to this factor, we evaluated its performance on ImageNet-C (ViT-B/16) using batch sizes ranging from 4 to 128. As shown in Table 26, BETA demonstrates high robustness to batch size variations. Even with a very small batch size of 4, where gradient estimates are typically noisy, BETA achieves an average accuracy of 59.3%, significantly outperforming the source model baseline of 55.5%. The performance consistently improves as the batch size increases, saturating at 62.6% for batch sizes of 64 and above. This indicates that while larger batches provide more stable gradients, BETA remains effective even in low-data regimes.

Table 26: Effect of batch size on average accuracy (%) on ImageNet-C. BETA consistently improves upon Source (55.5%) even with extremely small batch sizes.

Batch Size	**Source**	**4**	**8**	**16**	**32**	**64**	**128**
Avg. Acc (%)	55.5	59.3	60.1	62.3	62.5	62.6	62.6

B.18 Multi-Query Sensitivity

The main-body protocol allows BETA one API query per sample. We verified that the single-query design is the Pareto-optimal operating point and not a hard ceiling. On ImageNet-C with ViT-B/16 as the target, allowing $\{1, 2, 3\}$ queries per sample yields $\{62.6,\ 63.1,\ 63.3\}\%$ accuracy, so most of the gain is captured at the first query and diminishing returns set in immediately. This supports our default of one query per sample.

C Analytical Derivation and Algorithm

This appendix expands the analytical grounding of Sec. 2.4 and provides the full gradient-level view, a pseudocode listing, and a discussion of why the analysis is insensitive to the choice of steering or target architecture.

C.1 Entropy decomposition of the harmonized objective

The decomposition. The harmonized prediction $p_H = \alpha p_S + (1-\alpha)p_B$ is a convex combination of the two distributions. The standard mixture-entropy identity gives

$$\mathcal{H}(p_H) = \alpha\,\mathcal{H}(p_S) + (1-\alpha)\,\mathcal{H}(p_B) + \mathrm{JS}_\alpha(p_S \,\|\, p_B), \tag{8}$$

where $\mathrm{JS}_\alpha(p_S\|p_B) = \alpha D_{\mathrm{KL}}(p_S\|p_H) + (1-\alpha)D_{\mathrm{KL}}(p_B\|p_H) \geq 0$ is the α-weighted Jensen–Shannon divergence and equals zero if and only if $p_S = p_B$.

Why $\mathcal{H}(p_B)$ drops out. Because f_B is a black-box and admits no gradient with respect to the prompt δ, the term $(1-\alpha)\mathcal{H}(p_B)$ contributes no gradient signal during optimization. Dropping this constant gives the effective prompt loss

$$\nabla_\delta\,\mathcal{H}(p_H) = \nabla_\delta[\,\alpha\,\mathcal{H}(p_S) + \mathrm{JS}_\alpha(p_S \,\|\, p_B)\,]\,. \tag{9}$$

The harmonized objective therefore reduces to a sum of two terms that are both differentiable with respect to δ through the steering pathway: the steering entropy and the JS alignment.

The JS term as a directional anchor. The JS divergence depends on δ only through $p_S(x')$, but its second argument $p_B(x')$ enters as a fixed reference at every step. Minimizing $\mathrm{JS}_\alpha(p_S\|p_B)$ therefore pulls p_S toward p_B and biases the prompt update toward classes the target already favors. Without this term, i.e., when only $\mathcal{H}(p_S)$ is minimized, the optimizer has no information about the target's belief and can drive p_S to high confidence on a class that p_B rejects. We observe this failure mode whenever harmonization is disabled (Fig. 3); adding the JS term restores convergence.

Distinction from inference-time ensembling. A natural alternative is to train p_S on its own and average the two predictions at inference. This does not reproduce the directional anchoring above: at inference the prompt has already been shaped by the steering-only objective, so any prompt-level corruption that drives p_S to a wrong class persists into the averaged prediction. Eq. (9) instead couples p_B into the optimization itself, which Sec. B.8 confirms numerically (Adapt+Ensemble: 57.1%, Prompt+Ensemble: 54.7%, BETA: 62.6%).

Algorithm 1 BETA: Black-box Efficient TTA (one online step).

Require: batch B_t, black-box API f_B, steering model f_S with normalization parameters θ, prompt δ, weight α, filter margin ϵ, KL weight λ, EMA $\bar{p}_{t-1}$.

$x' \leftarrow x + \delta$ {shared prompted input}

$p_B \leftarrow f_B(x')$ {single API call; detached, no gradient}

$p_S \leftarrow f_S(x'; \theta)$ {local, differentiable}

$p_H \leftarrow \alpha p_S + (1-\alpha)\, p_B$ {Eq. (1)}

Compute filter weights w_H, w_S from $\mathcal{H}(p_S)$ and the EMA $\bar{p}_{t-1}$

$\mathcal{L} \leftarrow w_H\, \mathcal{H}(p_H) + w_S\, \mathcal{H}(p_S) + \lambda\, D_{\mathrm{KL}}\big(p_S(x) \,\|\, p_S(x')\big)$

$\delta \leftarrow \delta - \eta_\delta\, \nabla_\delta \mathcal{L}$ {gradient flows through f_S only}

$\theta \leftarrow \theta - \eta_\theta\, \nabla_\theta\big(w_S\, \mathcal{H}(p_S)\big)$

Return prediction: $\arg\max_k\, p_H^k$ {use this batch's harmonized forward-pass output}

C.2 Gradient-level derivation

Starting from Eq. (8) and dropping the $(1-\alpha)\mathcal{H}(p_B)$ term, which is constant in δ, the prompt gradient of the harmonized loss is

$$\nabla_\delta\, \mathcal{H}(p_H) \;=\; -\alpha \sum_{k=1}^{K} \big(1 + \log p_H^k\big)\, \tilde{p}_S^k, \tag{10}$$

where $p_H^k = \alpha p_S^k + (1-\alpha) p_B^k$ and $\tilde{p}_S^k = \nabla_\delta p_S^k$. Compared to the steering-only gradient $-\alpha \sum_k (1 + \log p_S^k)\, \tilde{p}_S^k$, the only change is the reweighting $\log p_S^k \mapsto \log p_H^k$. When p_B^k is large the factor $|1 + \log p_H^k|$ shrinks, so the negative push toward class k is reduced and the prompt update is biased toward classes the target already favors. This directional anchoring cannot be recovered by an inference-time ensemble, because inference-time combination acts after the prompt has been learned without any guidance from p_B. Sec. B.8 confirms this numerically.

C.3 Why cross-architecture steering works

BETA optimizes in pixel space, not in any network's internal representation. Because the learnable shift δ is added directly to the input and both f_S and f_B consume $x' = x + \delta$, any direction that makes an image easier for f_S to classify correctly tends to move it closer to the shared low-level data manifold on which both networks were trained. This geometric effect is insensitive to whether the target or the steering model is convolutional or attention-based. The JS term in Eq. (9) then rules out prompts that only help the steering model, because such prompts raise $\mathrm{JS}_\alpha(p_S \| p_B)$.

C.4 Algorithm

Algorithm 1 gives the full per-batch update in pseudocode. It makes explicit that p_B is detached (no gradient), the visual prompt δ receives gradient only through f_S, and the local normalization parameters θ are updated separately.

D Failure Modes and Limitations

Label Semantics and Steering Capacity. BETA needs a local steering model with task-relevant representations [63] and meaningful correspondence to the target label space. Public CLIP text encoders may provide weak semantic representations for specialized pathology codes or rare instrument identifiers, yielding a less informative gradient for the visual prompt. In the extreme case of a randomly initialized steering model, predictions have nearly maximal entropy and the filter rejects almost all samples, suppressing uninformative updates and keeping performance close to Source. This is a qualitative failure regime rather than a guarantee for every mismatched model. Closed-set class correspondence is described in Appendix A.1; partially overlapping and open-vocabulary outputs require additional mapping mechanisms.

Local steering capacity also affects the attainable gain. On ImageNet-C with a ViT-B/16 target, the improvement decreases from 7.1 percentage points with ViT-S/16 to 2.7 points with ViT-T/16 (Table 7). More severely compressed steering models may provide less useful gradients. The CPU

profiles in Table 17 quantify the computational trade-off for the tested models; real-time feasibility depends on the local hardware, batch size, and API latency.

Confidently Incorrect Target Predictions. When the target assigns high confidence to a wrong class, the harmonization objective may anchor the prompt toward that incorrect prediction. Unlabeled output probabilities alone may not expose this error, and the entropy filter does not guarantee its correction. This limits adaptation under shifts that systematically induce confident target errors.

Architectural and Task Scope. Our experiments demonstrate cross-architecture steering between CNNs and Transformers, including ResNet-50 steering a ViT-B/16 target to 60.8% accuracy (Table 7). Architectural matching is therefore not required, although the quality of the available steering signal affects performance. We have not evaluated Mamba targets or established performance across all architecture families.

The empirical evaluation follows standard TTA protocols and focuses on classification under distribution shift. Extending the framework to detection or segmentation would require object-level or per-pixel output correspondence and harmonization. Language and generative tasks would require suitable token-level outputs and a reformulated adaptation objective. These extensions remain future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Sec. 1) state the three contributions (first systematic study of strict black-box TTA, the BETA framework with prediction harmonization and stabilization mechanisms, and SOTA black-box results), and each is supported by experiments in Sec. 3, Tables 2–4, and the Clarifai evaluation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations and failure modes in Appx. D, including three concrete failure regimes for the zero-shot steering recipe (semantic mismatch with public CLIP, severely constrained local compute, and adversarial overconfidence of the target), as well as architectural and task-scope limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The analytical insight in Sec. 2.4 is grounded in the entropy decomposition of the harmonized objective. The full derivation, gradient-level form, and discussion of architecture-insensitivity appear in Appx. C, with all assumptions (f_B is non-differentiable, $\alpha \in [0, 1]$) stated explicitly.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the BETA algorithm fully in Sec. 2.5, provide pseudocode in Algorithm 1 (Appx. C), specify all hyperparameters (α, λ, ϵ, optimizer choices, learning rates, prompt structure) in the Implementation Details paragraph of Sec. 3, and use only publicly available datasets (ImageNet-C/S/R, EuroSAT, Derm7pt) and models (ViT-B/16, ViT-L/16, CLIP, BiomedCLIP, ViT-S/16).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and setup instructions are available at https://github.com/yunbeizhang/BETA, linked from the project website in the abstract. All datasets used (ImageNet-C/S/R, EuroSAT, Derm7pt) and pre-trained models (ViT, CLIP, BiomedCLIP) are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The Implementation Details paragraph of Sec. 3 specifies optimizer (AdamW for the prompt, SGD for steering normalization), learning rates (0.01 and 2×10^{-5}), $\alpha = 0.4$, $\lambda = 50$, entropy threshold $\epsilon = 0.9 \log K$ for K classes, and prompt structure (16-pixel padded frame, 39,936 parameters, Gaussian initialization). Baseline protocols appear in Appx. A.2; the disjoint validation procedure and class-count scaling of ϵ are specified in Appx. A.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Tables 10 and 14 report five independent runs of the full method on ImageNet-C and the Clarifai Contrast domain, including individual accuracies and mean ± standard deviation. ImageNet-C statistics are computed from unrounded accuracies across independent random-seed runs. Main accuracy tables retain the original single-run results for comparability.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Table 5 reports wall-clock latency and GPU memory on a single NVIDIA RTX 3090. Table 17 reports complete local-update latency and peak process memory on an AMD Ryzen Threadripper PRO 5955WX with eight CPU threads, excluding API and network latency. The Clarifai experiments report API budgets explicitly.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

Justification: The research uses only publicly available academic benchmarks and pre-trained models, performs no human-subjects experiments, and adheres to the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: BETA enables users of commercial black-box vision APIs to adapt to distribution shifts at low cost, with positive impact on accessibility (lower API spend) and on specialized domains such as medical imaging where robust adaptation matters. We do not foresee direct negative societal impacts beyond those already associated with the underlying pre-trained models, since BETA does not modify or replace those models.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release new pre-trained models, datasets, or generative systems with high misuse risk. BETA is an inference-time adaptation method that operates on top of existing public models and benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets (ImageNet, ImageNet-C/S/R, EuroSAT, Derm7pt) and pre-trained models (ViT, CLIP, BiomedCLIP, DDA) are cited to their original publications and used under their respective public licenses. The Clarifai API is used as a paid commercial service under its standard terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The BETA code is publicly available with installation, data preparation, and usage instructions at `https://github.com/yunbeizhang/BETA`. No new datasets or pre-trained model assets are released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The work involves no crowdsourcing or human-subjects research; all evaluations use existing public benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The work involves no human-subjects research and therefore requires no IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were used only for minor language polishing and do not contribute to the core methodology, experimental design, or scientific claims of this work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.