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]University of Melbourne

ABSTRACT

Test-Time Adaptation (TTA) for black-box models accessible only via APIs remains largely unexplored. Existing approaches such as post-hoc output refinement offer limited adaptive capacity, while Zeroth-Order Optimization (ZOO) enables input-space adaptation but suffers from high query costs and instability in the unsupervised setting. We introduce **BETA** (Black-box Efficient Test-time Adaptation), a framework that uses a lightweight local steering model to create a tractable gradient pathway. Through prediction harmonization, consistency regularization, and prompt learning-oriented filtering, BETA enables stable adaptation with no additional API calls and negligible latency. On ImageNet-C, BETA achieves +7.1% accuracy gain on ViT-B/16 and +3.4% on CLIP, surpassing strong white-box methods. On a commercial API, BETA matches ZOO performance at 250× lower cost.

1 INTRODUCTION

Modern deep learning models often suffer performance degradation when deployed in the wild due to distribution shifts between training and test data (Recht et al., 2019; Koh et al., 2021). Test-Time Adaptation (TTA) (Sun et al., 2020; Wang et al., 2021; Niu et al., 2023; Wang et al., 2022; Manli et al., 2022; Zhang et al., 2025c; Maharana et al., 2026) has emerged as a crucial approach to address this challenge, adapting models on-the-fly using unlabeled target data. However, many state-of-the-art models are now deployed as opaque APIs (Hurst et al., 2024; Team et al., 2023), where users can only submit inputs and receive output predictions. We study TTA in this *strict black-box setting*: the user sends a raw image to the API and receives output probabilities, with the model's architecture, parameters, and data entirely unknown. Beyond the adaptation challenge, each API call incurs monetary cost and network latency, making query efficiency a first-class concern.

This setting is fundamentally harder than white-box TTA, where methods rely on backpropagation through model parameters (Wang et al., 2021; Niu et al., 2023; Wang et al., 2022; Zhang et al., 2025b), or supervised black-box adaptation using labeled support sets (Oh et al., 2023; Zhang et al., 2025a). Recent backpropagation-free methods (Niu et al., 2024; Zhou et al., 2025; Wang et al., 2024) improve efficiency but still require access to internal tokens or features, placing them in a "gray-box" category (Table 1). Truly black-box methods are scarce and face distinct limitations: output modification methods like LAME (Boudiaf et al., 2022) offer limited adaptive capacity; augmentation-based methods (Farina et al., 2024) linearly increase API costs; purification-based methods (Gao et al., 2023) require source domain training and introduce substantial latency; and ZOO-based prompting (Liu et al., 2018; Hansen & Ostermeier, 2001; Xiao et al., 2026) suffers from prohibitive query costs and catastrophic instability when guided by noisy unsupervised signals. For instance, accuracy on the Contrast corruption collapses from 32.6% to 4.1% with ZOO (Table 2).

We propose **BETA** (Black-box Efficient Test-time Adaptation), which achieves stable and efficient adaptation via a 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. Since naive gradient transfer between architectures is ineffective (gradient similarity ≈ 0.0006), BETA 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, we introduce *consistency regularization* and *prompt learning-*

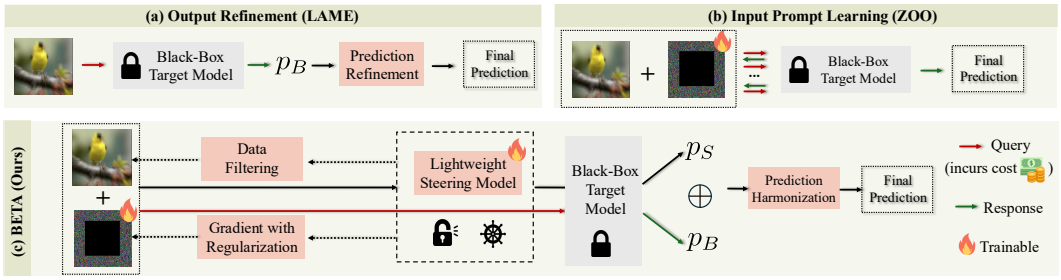


Figure 1: Comparison of black-box TTA strategies. **(a)** Output Refinement (LAME) is limited to post-processing. **(b)** ZOO requires multiple expensive API calls. **(c)** BETA achieves single-query adaptation via a lightweight steering model with prediction harmonization, stabilized through filtering and regularization.

Table 1: Comparison of TTA methods across key capabilities. BETA is the only method satisfying all desiderata.

Access	Method	w/o Params.	w/o Tokens	w/o Feats.	w/o Grad.	Arch-Agnostic	VMs	VLMs	1 API/Sample	Low Latency
▭	TENT (Wang et al., 2021)	✗	✗	✗	✗	✓	✓	✓	✓	✓
	TPT (Manli et al., 2022)	✗	✗	✗	✗	✓	✗	✓	✓	✓
▭	T3A (Iwasawa & Matsuo, 2021)	✗	✓	✗	✓	✓	✓	✓	✗	✓
	FOA (Niu et al., 2024)	✓	✗	✗	✓	ViT-only	✓	✓	✗	✗
	B^2TPT (Meng et al., 2025)	✓	✗	✓	✓	ViT-only	✗	✓	✗	✗
	BCA (Zhou et al., 2025)	✓	✓	✗	✗	✓	✓	✓	✓	✓
▭	LAME (Boudiaf et al., 2022)	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Augmentation (Farina et al., 2024)	✓	✓	✓	✓	✓	✓	✓	✗	✗
	Purification (Gao et al., 2023)	✓	✓	✓	✓	✓	✓	✓	✗	✗
	ZOO	✓	✓	✓	✓	✓	✓	✓	✗	✗
	BETA (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓

oriented filtering. Our contributions: (1) the first systematic study of TTA in the strict black-box setting, revealing limitations of existing approaches; (2) BETA, which bypasses expensive queries via prediction harmonization stabilized by consistency regularization and prompt-oriented filtering; (3) new state-of-the-art results surpassing white-box methods with a single API call per sample and a 250× cost advantage over ZOO on a commercial API.

2 METHOD

Problem Setup. TTA adapts a model to an unlabeled target domain $\mathcal{D}_T$ during inference. In the black-box setting, we have a *target model* f_B (the API) from which we only obtain predictions $p_B(x)$, and a *steering model* f_S (e.g., ViT-Small) with full local access. To adapt the black-box model without altering its weights, we learn an additive visual prompt $\delta \in \mathbb{R}^{H\times W\times C}$ added to inputs: $x' = x + \delta$. The goal is to optimize δ using gradients derived locally from f_S to improve the target model's predictions.

Prediction Harmonization. A straightforward approach is to minimize the entropy $\mathcal{H}(p_B(x'))$ via ZOO, but this is costly (e.g., CMA-ES requires 28 queries per sample) and fundamentally unstable under unsupervised signals. Alternatively, transferring the local gradient $g_{\text{Local}} = \nabla\mathcal{H}(p_S; f_S)$ from the steering model is ineffective, as cross-architecture gradient similarity is near zero (≈ 0.0006; Appendix Fig. 4). To overcome this, we relax the problem to finding a prompt that improves both models simultaneously. We define a *harmonized prediction* fusing the outputs:

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

and employ an asymmetric optimization strategy: we compute the gradient of the harmonized entropy but restrict gradient flow exclusively to the steering model's pathway, yielding our tractable proxy $g_{\text{BETA}} = \nabla_\delta\mathcal{H}(p_H; f_S)$. The parameter α navigates a trade-off between *Objective Relevance* (alignment with the true target gradient) and *Optimization Effectiveness* (tractability via f_S). 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. Empirically, $\alpha \in [0.3, 0.5]$ balances both well.

Stabilization. While prediction harmonization provides a tractable gradient pathway, optimizing the harmonized objective alone is inherently unstable when applied in isolation. As shown in Fig. 2a,

Table 2: Accuracy (%) on ImageNet-C (severity 5) using **ViT-B/16** as the black-box model. Within black-box methods, **bold** = best, underline = second best. *White/gray-box shown for reference.*

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	44.9	-10.6
■	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 (Niu et al., 2024) uses entropy minimization as the original activation discrepancy requires source statistics unavailable in the black-box setting.

†ZOO-SPSA-GC adapted from Oh et al. (2023); uses entropy instead of cross-entropy.

‡TT-Aug adapted from Shanmugam et al. (2021); we only aggregate predictions over augmented views.

§DDA requires training a diffusion model on source data (Gao et al., 2023); we use the released model.

five independent runs on ImageNet-C Contrast using solely Eq. 1 lead to either gradual performance decay or catastrophic collapse. This instability arises because noisy unsupervised signals cause the optimization to learn degenerate solutions that corrupt the input's semantic content. We introduce two complementary stabilization mechanisms.

(i) Prompt Learning-oriented Filtering. We update the prompt using only samples whose steering model entropy $\mathcal{H}(p_S(x))$ falls below a threshold ϵ, weighted by confidence: $\mathcal{L}_{\text{Harmon}}(x') = w_H(x') \cdot \mathcal{H}(p_H(x'))$, where $w_H(x) = \frac{1}{\exp[\mathcal{H}(p_S(x))-\epsilon]} \cdot \mathbb{I}_{\{\mathcal{H}(p_S(x))<\epsilon\}}$. Unlike prior filtering designed for pre-trained normalization parameters (Niu et al., 2022; 2023), we retain all reliable samples including redundant ones, since learning a visual prompt from random initialization is more data-hungry than fine-tuning existing parameters.

(ii) Consistency Regularization. Since prompts are randomly initialized, an unconstrained entropy objective can be minimized by learning degenerate solutions that destroy the model's representations. We introduce a consistency loss that anchors updates to pre-trained knowledge via KL-divergence between predictions on clean and prompted images: $\mathcal{L}_{\text{consist}}(x, x') = D_{\text{KL}}(p_S(x) \| p_S(x'))$.

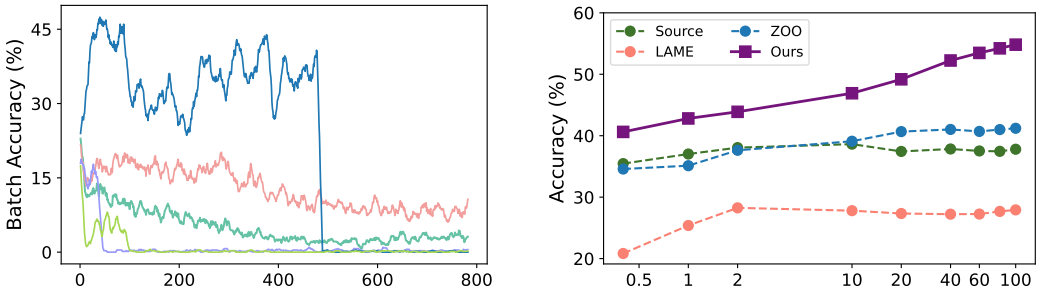
Final Objective. BETA operates online with a single gradient step per batch. The prompt δ is updated via the harmonization loss and consistency regularization, while the steering model's normalization layers are updated via a steering loss $\mathcal{L}_{\text{Steer}}$ using reliable, non-redundant samples (Niu et al., 2022):

$$\mathcal{L}_{\text{BETA}} = \mathbb{E}_{x \in B_t} \left[\mathcal{L}_{\text{Harmon}}(x') + \mathcal{L}_{\text{Steer}}(x') + \lambda \mathcal{L}_{\text{consist}}(x, x') \right]. \tag{2}$$

3 Experiments

Setup. We evaluate on ImageNet-C (severity 5) (Hendrycks & Dietterich, 2019), ImageNet-S (Wang et al., 2019), and ImageNet-R (Hendrycks et al., 2021). Black-box targets: ViT-B/16 (87M), ViT-L/16 (304M), CLIP-B/16 (150M), and a commercial Clarifai API ($0.0032/request). Steering model: ViT-S/16 (22M). We set α=0.4, λ=50, ϵ=0.9$\cdot \ln(1000)$, prompt width=16px.

Results on ImageNet-C. Table 2 shows full per-corruption results on ImageNet-C with ViT-B/16. Among black-box methods, LAME fails to improve upon the source (54.1% vs. 55.5%), confirming the limited capacity of output-only refinement. ZOO approaches exhibit severe instability on challenging corruptions: CMA-ES collapses to 4.1% on Contrast, and all three ZOO variants degrade substantially on Blur and Weather corruptions despite requiring 16× more API calls. In contrast, BETA achieves 62.6% average accuracy (+7.1% gain) with a single API call per sample, delivering consistent improvements across all 15 corruption types. Notably, BETA provides the largest gains on corruptions where other methods struggle most: +20.8% on Contrast, +16.6% on Glass blur, and +11.5% on Zoom blur. BETA outperforms white-box methods TENT (59.6%) and CoTTA (61.6%) despite operating under much stricter access constraints, and approaches the strong white-box ETA (65.8%). Results with the larger ViT-L/16 (304M) show a consistent +4.0% gain (Appendix Table 4).



(a) Optimization instability (b) Clarifai API cost

Figure 2: (a) Five runs using only Eq. 1 on ImageNet-C Contrast show collapse or failure. (b) BETA achieves 250× cost advantage over ZOO on the Clarifai API.

Results on ImageNet-S/R and VLMs. Table 3 shows results on ImageNet-Sketch and ImageNet-Rendition, which test adaptation to artistic and stylistic domain shifts rather than corruptions. On ViT-B/16, BETA achieves 56.3% average (+4.1% gain), surpassing all black-box baselines and outperforming white-box methods including TENT, SAR, and T3A. We further extend BETA to CLIP-B/16, representing—to our knowledge—the first work exploring TTA for VLMs in the strict black-box setting. BETA is the only black-box method that effectively improves CLIP, achieving 63.4% average accuracy. Remarkably, BETA surpasses specialized white-box VLM methods like TPT (62.5%) and DynaPrompt (63.2%), as well as gray-box methods such as TCA (63.0%) and B^{2}TPT (64.1%), demonstrating that BETA's prediction harmonization generalizes effectively beyond standard vision models.

Real-world API and Efficiency. On the commercial Clarifai API (Fig. 2b), BETA achieves +5.2% gain with just $0.4, while ZOO requires over $100 for comparable performance—a **250× cost advantage**. At $100, BETA delivers +17.1% gain. BETA adds only 3ms local overhead per image, while ZOO (16 calls) and TT-Aug (64 calls) are 9.4× and 37.5× slower. Under real-time streaming constraints (Alfarra et al.), ZOO drops to 54.3% while BETA maintains 62.5% (Appendix Table 5).

Ablation. We provide component analysis, hyperparameter sensitivity, steering model choice (including cross-architecture CNN→Transformer transfer), and robustness analyses in the Appendix. Key finding: output-only adaptation fails, and naively adding a prompt causes collapse; KL regularization and filtering each provide +4–5% gains, and combining both yields the full 62.6%.

Table 3: Accuracy (%) on ImageNet-S/R. "–": not applicable. BETA improves both VMs and VLMs.

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
■	**BETA (Ours)**	**49.3**	**63.3**	**56.3**	**50.9**	76.0	**63.4**

¶ZERO originally requires logits for re-scaling for CLIP (Farina et al., 2024); we adapt it to use output prediction probabilities.

4 Conclusion

We introduced BETA, a framework for efficient and stable test-time adaptation in the strict black-box setting via prediction harmonization with a lightweight steering model. By fusing outputs from the steering and target models, BETA creates a tractable gradient pathway that avoids the high query costs and instability of zeroth-order optimization. Combined with consistency regularization and prompt learning-oriented filtering, BETA achieves stable adaptation with a single API call per sample. Experiments across diverse benchmarks and model types—including vision models, vision-language models, and a commercial API—demonstrate that BETA surpasses strong white-box methods while maintaining a 250× cost advantage, making robust black-box TTA a practical reality for real-world deployment.

REFERENCES

Motasem Alfarra, Hani Itani, Alejandro Pardo, Merey Ramazanova, Juan Camilo Perez, Matthias Müller, Bernard Ghanem, et al. Evaluation of test-time adaptation under computational time constraints. In *Forty-first International Conference on Machine Learning*.

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)*, pp. 8334–8343, 2022. URL https://api.semanticscholar.org/CorpusID:246015836.

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.

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)*, pp. 11786–11796, June 2023.

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.

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

Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

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.

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

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.

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 (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2427–2440. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/1415fe9fea0fa1e45dddcff5682239a0-Paper.pdf.

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.

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*, pp. 5637–5664. PMLR, 2021.

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.

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.

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, February 2026. URL https://doi.org/10.5281/zenodo.18665186.

Shu Manli, Nie Weili, Huang De-An, Yu Zhiding, Goldstein Tom, Anandkumar Anima, and Xiao Chaowei. Test-time prompt tuning for zero-shot generalization in vision-language models. In *NeurIPS*, 2022.

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, pp. 6099–6107, 2025.

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

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 *Internetional Conference on Learning Representations*, 2023.

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.

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)*, pp. 24224–24235, June 2023.

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

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*, pp. 1214–1223, 2021.

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

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*, pp. 9229–9248. PMLR, 2020.

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.

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*, pp. 9614–9624. PMLR, 2020.

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.

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*, pp. 10506–10518, 2019.

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*, pp. 7201–7211, 2022.

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.

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.

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.

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.

Yunbei Zhang, Chengyi Cai, Feng Liu, and Jihun Hamm. Prime Once, then Reprogram Locally: An Efficient Alternative to Black-Box Model Reprogramming. *OpenReview*, 2025a. URL https://openreview.net/forum?id=Hic8PwzlBY.

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

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

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.

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*, pp. 29999–30009, 2025.

Appendix

This appendix provides supplementary material organized as follows: Appendix A formalizes the black-box setting and describes all baseline methods; Appendix B.5 presents additional quantitative results and comparisons including knowledge distillation (Appendix B.7) and zeroth-order optimization baselines; Appendix B.11–B.14 evaluate robustness under challenging conditions and computational efficiency; finally, Appendix C discusses limitations of our approach.

A Extended Related Work & Black-Box Setting Analysis

A.1 Detailed Analysis of Model Accessibility and Security Constraints

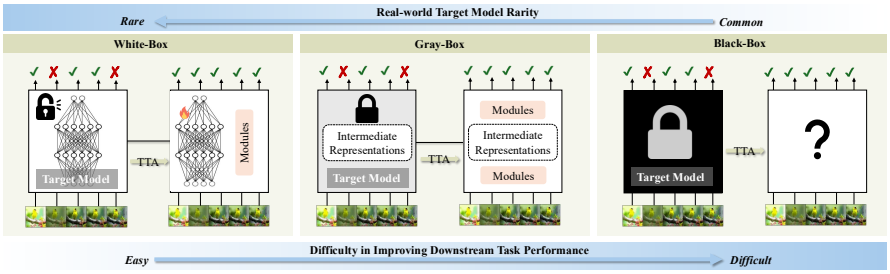


Figure 3: The black-box Test-time Adaptation setting studied in this work. From the client's perspective, the goal is to adapt a powerful server-side API model to a target distribution (e.g., corrupted or domain-shifted images) without any internal access. The client can only send raw input images and 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 (Hurst et al., 2024), 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 (Hinton et al., 2015). 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 (Farina et al., 2024)) as *Gray-Box*. While these methods do not require gradients, they rely on information often hidden in secure deployment environments.

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 (Wang et al., 2021) 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 (Niu et al., 2023) 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) (Wang et al., 2022) 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) (Iwasawa & Matsuo, 2021) 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) (Niu et al., 2024) 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 (Boudiaf et al., 2022) is a **black-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.

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

Test-Time Prompt Tuning (TPT) (Manli et al., 2022) 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) (Zhang et al., 2024) 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 (Xiao et al., 2025) 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 (Meng et al., 2025) 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) (Karmanov et al., 2024) 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) (Lee et al., 2025) 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) (Zhou et al., 2025) 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 em-

bedding 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) (Wang et al., 2024) 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.

B Comprehensive Quantitative Analysis

B.1 Gradient Analysis

To validate the prediction harmonization mechanism, we analyze the gradient alignment between different optimization strategies. We compute the cosine similarity between gradient vectors across four validation corruption domains using ViT-B/16 and ViT-L/16 as black-box targets and ViT-S/16 as the steering model. For this analysis only, we temporarily assume white-box access to the target models to compute the otherwise inaccessible reference gradients. As shown in Fig. 4, the local gradient $g_{\text{Local}} = \nabla\mathcal{H}(p_S; f_S)$ has near-zero cosine similarity with the target gradient $g_{\text{Black}} = \nabla\mathcal{H}(p_B; f_B)$ (≈ 0.0006), confirming that naive gradient transfer across architectures is ineffective. ZOO gradient estimates are equally noisy in the one-step setting despite their high query cost. In contrast, BETA's harmonized gradient g_{BETA} achieves meaningful alignment with the ideal gradient g_{Ideal} by navigating a trade-off between two competing factors: *Objective Relevance*, which measures how well the harmonized objective aligns with the true target ($\cos(g_{\text{Ideal}}, g_{\text{Black}})$), and *Optimization Effectiveness*, which measures how well the practical proxy can optimize this objective ($\cos(g_{\text{BETA}}, g_{\text{Ideal}})$). 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. The intersection of these curves identifies the optimal range $\alpha \in [0.3, 0.5]$ where both factors are balanced, confirming that BETA succeeds by constructing a shared optimization problem rather than directly approximating the target gradient.

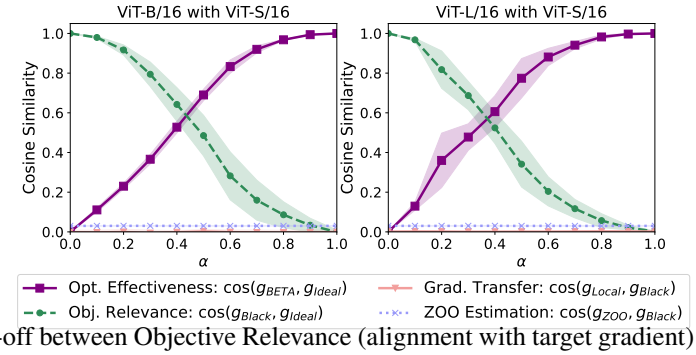


Figure 4: Trade-off between Objective Relevance (alignment with target gradient) and Optimization Effectiveness (alignment with steering gradient) as a function of α. The intersubsection identifies the optimal range ($\alpha \in [0.3, 0.5]$).

B.2 Extended Results

ImageNet-C with ViT-L/16. Table 4 shows results with ViT-L/16 (304M) as the black-box model. BETA again achieves the best black-box performance (+4.0% gain).

B.3 Efficiency Analysis

Table 5 compares the computational efficiency of black-box TTA methods on ImageNet-C with ViT-B/16. BETA requires only a single API call per image and adds just 3ms of local overhead for the

Table 4: 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

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

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

steering model's backward pass, resulting in a total wall-clock time of 48ms that effectively matches standard source inference (45ms). In contrast, ZOO-SPSA-GC requires 16 API calls per image (450ms, 9.4× slower), TT-Aug aggregates predictions over 64 augmented views (1,800ms, 37.5× slower), and DDA's iterative diffusion-based denoising incurs extreme latency (12,722ms) and GPU memory (23.4GB). The key insight is that local computation is negligible compared to API latency: the API forward pass (∼45ms) is dominated by network overhead, making the steering model's 3ms backward pass practically invisible. Under a strict real-time streaming protocol (Alfarra et al.), where methods that cannot keep pace with the data stream must skip samples, ZOO's accuracy drops to 54.3

B.4 Additional Ablation Studies

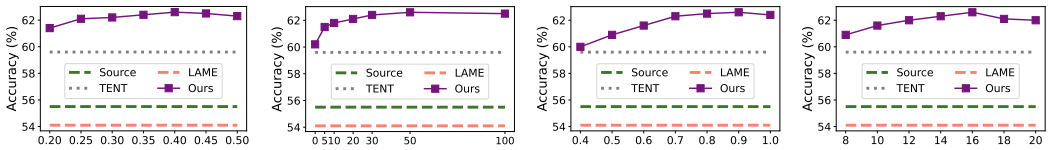
Component Analysis. Table 6 shows that output-only adaptation (Exp-1) fails, naively adding a prompt (Exp-2) causes collapse, and the full BETA with both KL regularization and filtering achieves the best 62.6%.

Table 6: Component analysis on ImageNet-C (ViT-B/16).

Method	Prompt	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

Hyperparameter Sensitivity. BETA performs stably across fusion weight $\alpha \in [0.3, 0.5]$, regularization weight $\lambda \in [20, 100]$, entropy margin ϵ, and prompt sizes (8–20 pixel width). See Figure 5.

Steering Model Choice. Table 7 shows BETA consistently improves across steering models of different sizes and architectures, including cross-architecture transfer (ResNet-50 steering Transformer-based models).



(a) α (b) λ (c) ϵ (d) Prompt Size

Figure 5: Hyperparameter sensitivity analysis showing stable performance across all key parameters.

Table 7: Effect of steering model choice. The Source and TENT-adapted accuracy of each local steering model are provided as a reference against the BETA accuracy on the large black-box models.

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) CLIP-B/16 (150M)	44.9 46.1	44.4 45.4	45.1 46.0	20.9	21.3	45.2 47.0	27.9	29.7	47.5 48.7	32.8	35.6	**49.3** **50.9**
Average	-	48.8	48.0	49.0	21.1	21.7	50.1	26.7	30.2	52.3	35.0	41.0	**54.3**

Robustness. Under label imbalance (Niu et al., 2023) and continual domain shifts (Wang et al., 2022) on ImageNet-C, BETA maintains 61.8% and 61.5% accuracy respectively (vs. 62.6% standard), demonstrating minimal degradation. BETA is also robust to batch size variations, achieving 59.3% even with batch size 4.

B.5 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** (Farina et al., 2024), 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** (Meng et al., 2025), a recent gray-box prompt tuning method for VLMs.

Table 8: 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**

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 8 and Table 9.

BETA consistently outperforms these query-intensive baselines while maintaining strict API efficiency. On the ImageNet variants (Table 8), 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 effi-

Table 9: 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^2TPT (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**

ciency gap is even more pronounced on EuroSAT (Table 9), 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.6 White-box TTA performance on 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 adaptation performance of the ViT-Small steering model in Table 10. 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.

Table 10: White-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

B.7 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 our framework is not merely performing Test-Time Knowledge Distillation (KD), we implemented a KD baseline following the protocol in (Zhao et al., 2024). 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 11, 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 11: 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.8 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 (Hansen & Ostermeier, 2001; Hansen et al., 2003; Niu et al., 2024; Meng et al., 2025). 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 (Liu et al., 2018; Tsai et al., 2020). 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:

$$g_t(\delta_t) = \frac{1}{q} \sum_{i=1}^{q} \frac{f(\delta_t + \mu u_i) - f(\delta_t)}{\mu} u_i \tag{3}$$

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 (Oh et al., 2023). SPSA is a highly efficient ZOO algorithm that estimates the gradient using only two queries per iteration (Spall, 1992). 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:

$$\hat{g}_t(\delta_t) = \frac{f(\delta_t + \mu\Delta_t) - f(\delta_t - \mu\Delta_t)}{2\mu} \Delta_t \tag{4}$$

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 (Oh et al., 2023). 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.9 API EFFICIENCY COMPARISON ACROSS BLACK-BOX METHODS

Table 12 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. LAME, 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.

B.10 ORTHOGONALITY OF CONTRIBUTION: UNSUPERVISED OBJECTIVE VS. ZOO ALGORITHMS

While we adopt the powerful ZOO algorithm like SPSA-GC (Oh et al., 2023) due to its superior efficiency, it is crucial to distinguish the role of the *ZOO algorithm* from the challenges inherent to the *adaptation objective*. The efficacy of SPSA-GC was originally demonstrated in BlackVIP (Oh et al., 2023) within a *supervised* few-shot transfer setting. In that context, the loss landscape is anchored by ground-truth labels via a Cross-Entropy loss, providing a consistent and convex directional signal for the zeroth-order estimator.

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, the Coordinator architecture, 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 12: API efficiency comparison across black-box TTA methods. 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.11 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** (Niu et al., 2023; Gong et al., 2022): Following the protocol established in SAR (Niu et al., 2023), we evaluate performance on data streams with highly skewed class distributions within each batch, simulating non-i.i.d. test streams.
- **Continual Domain Shifts** (Wang et al., 2022; Niu et al., 2022): Following the Continual Test-Time Adaptation (CoTTA) setting (Wang et al., 2022), the model adapts to the 15 corruption domains of ImageNet-C sequentially without resetting the model state between domains.

Table 13: 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

The results are summarized in Table 13. 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.12 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.

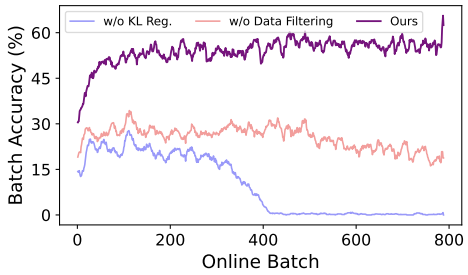


Figure 6: Online Batch Accuracy on ImageNet-C Contrast domain.

B.13 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 14, 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 14: 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.14 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 (Alfarra et al.).

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 12, we compare BETA against baselines including ZOO-SPSA-GC† and Test-Time Augmentation (TT-Aug) (Shanmugam et al., 2021).

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 (Alfarra et al.). This protocol penalizes methods that cannot keep pace with a data stream arriving at the API's maximum throughput speed ($r = 1$ 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. Since $T_{API} \gg T_{Local\_Fwd}$, 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 15 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 15: 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**

C Limitations

While BETA demonstrates strong performance and efficiency, its effectiveness is connected to the choice of the local steering model. In the current landscape, where most large-scale models are Transformer-based, our method is highly applicable, as finding a steering model with a similar ar-

chitecture is straightforward. However, the performance could be suboptimal if the architectures of the steering and target models differ significantly. Although our experiments show that cross-architecture adaptation is effective (e.g., a CNN steering a Transformer), the improvements are slightly less pronounced than when using architecturally similar models. Another avenue for future research is extending this framework beyond classification to more versatile, generative tasks. Investigating how to adapt the harmonized objective for generative outputs, where the prediction space is vast and unstructured, would be a valuable next step.