LoRA-TTT: Low-Rank Test-Time Training for Vision-Language Models

Yuto Kojima [1 2] **Jiarui Xu** [2] **Xueyan Zou** [2] **Xiaolong Wang** [2]

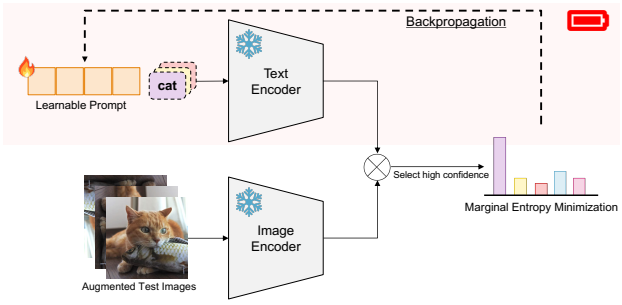
Abstract

We propose LoRA-TTT, a novel test-time training (TTT) method for vision-language models (VLMs) that leverages Low-Rank Adaptation (LoRA), applied exclusively to the image encoder. Unlike prior TTT approaches that rely on computationally intensive text prompt tuning and entropy-based loss, LoRA-TTT updates only LoRA parameters at test time, achieving substantial performance gains with minimal memory and runtime overhead. We also introduce an efficient reconstruction loss tailored for TTT. Experiments on 15 datasets show that LoRA-TTT improves zero-shot top-1 accuracy of CLIP-ViT-B/16 by 5.79% on OOD and 1.36% on fine-grained benchmarks, without using external models or caches.

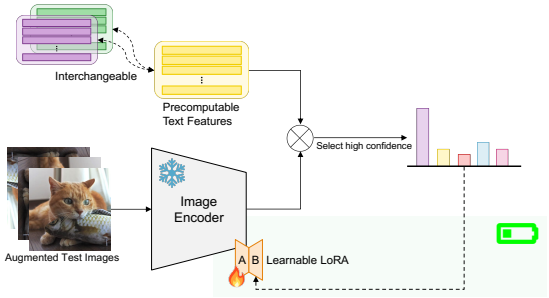
1. Introduction

Recent advances in large-scale Vision-Language Models (VLMs) have enabled strong zero-shot generalization. However, their performance often degrades under domain shifts (Shu et al., 2023; Xiao et al., 2024), which are common in real-world scenarios.

Inspired by recent advancements in Parameter-Efficient Fine-Tuning (PEFT) (Han et al., 2024; Xu et al., 2023; Ding et al., 2023), we propose **Lo**w-**Ra**nk **T**est-**T**ime **T**raining (LoRA-TTT), a test-time training (TTT) method that efficiently adapts VLMs to distribution shifts by updating only Low-Rank Adaptation (LoRA) (Hu et al., 2021) parameters in the image encoder, reducing memory usage and mitigating catastrophic forgetting. As shown in Figure 1, LoRA-TTT replaces prompt tuning with LoRA tuning, updating LoRA parameters during test time using the marginal entropy minimization (MEM) loss (Zhang et al., 2022; Shu et al., 2022). Because LoRA-TTT focuses solely on vision-side parameters and precomputes text features, it eliminates the need for the text encoder at test time, significantly reducing runtime and memory. It also offers the flexibility of interchangeable text prompts.



(a) Test-time Prompt Tuning (Shu et al., 2022)



(b) LoRA-TTT (Ours)

Figure 1: **Comparison of our proposed LoRA-TTT and Test-time Prompt Tuning (TPT).** (a) TPT optimizes the learnable text prompt via backpropagation, which results in high memory consumption, long runtime, and non-interchangeable text prompts. (b) LoRA-TTT requires only the image encoder and tunes only the LoRA parameters, demonstrating high performance while minimizing memory consumption and ensuring faster runtime.

In addition, we introduce a lightweight reconstruction loss inspired by masked image modeling (Gandelsman et al., 2022; Wang et al., 2023; Liu et al., 2024), which improves calibration by mitigating the overconfidence caused by MEM loss (Guo et al., 2017; Yoon et al., 2024). The combination of MEM and reconstruction losses enables robust, resource-efficient adaptation in real-world scenarios, including high-stakes and memory-constrained environments (Wang et al., 2022; Liu et al., 2023; Dorbala et al., 2022;

[1]Sony Semiconductor Solutions Corporation, Tokyo, Japan [2]UC San Diego. Correspondence to: Yuto Kojima <yuto.kojima@sony.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

Khandelwal et al., 2022).

2. Method

2.1. LoRA-TTT

Application of LoRA for TTT. Focusing on the current mainstream method, TPT (Shu et al., 2022), our approach shifts the target of parameter updates from the text prompt to the image encoder, while leveraging the MEM loss. However, directly updating the entire image encoder is expected to result in excessive memory consumption and domain-specific behaviors that lose the out-of-distribution generalization and robustness of foundation models (Wortsman et al., 2022; Kumar et al., 2022). As shown in Figure 2, inspired by the effectiveness of LoRA in the large language model field, we apply LoRA to layers of the image encoder in VLMs, updating only the LoRA parameters during test time. The original LoRA paper shows that the change in weights during model adaptation has a low intrinsic rank and we hypothesize that LoRA can adapt to unique domain-specific features of each instance even with unlabeled data similar to fine-tuning in downstream tasks. By applying LoRA, we adjust only a small number of parameters without altering the original well pre-trained weights in VLMs. This approach allows individual adaptation to each test instance while preserving the strong zero-shot capability of the original VLMs and reducing memory consumption during backpropagation. Furthermore, LoRA-TTT applies LoRA exclusively to the vision encoder of VLMs, rendering the text encoder unnecessary during TTT. This allows the model to be tuned independently of specific text prompts. Following the approach of Episodic TTT (Wang et al., 2020; Shu et al., 2022; Zhao et al., 2023), we update parameters using only a single test instance and reset them afterward, ensuring a certain level of robustness against sequence data.

Masked Image Reconstruction for VLMs. LoRA-TTT is not limited by the type of loss function. We are able to leverage a self-supervised reconstruction loss based on masked autoencoders (MAE) (He et al., 2022; Gandelsman et al., 2022), owing to the benefits of parameterizing the image encoder. LoRA-TTT differs from conventional TTT methods based on MAE (Gandelsman et al., 2022; Wang et al., 2023) and offers an efficient solution suitable for TTT, as it requires neither an image decoder nor fine-tuning of the model prior to TTT. LoRA-TTT takes both augmented images and their randomly masked versions as input to the image encoder and calculates the mean squared error of only the encoded class tokens as the loss. These images are selected from the top 10% of views with the highest confidence, similar to TPT. We optimize the following loss:

$$\mathcal{L}_{\text{MAE}} = \text{MSE}(g(X)_{\text{cls}}, g(\text{mask}(X))_{\text{cls}}), \tag{1}$$

where $\text{MSE}(\cdot, \cdot)$ represents the mean squared error between the encoded class tokens of the masked and unmasked images, $\text{mask}()$ randomly masks out majority of the input image patches (*e.g.*, 50%). This loss encourages the model to reconstruct the original global features by leveraging the remaining visual clues, enhancing visual understanding to better support downstream tasks. The total loss can be expressed as $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{MEM}} + \lambda_2 \mathcal{L}_{\text{MAE}}$, where λ_1 and λ_2 are coefficients that balance the two losses.

3. Experiments

3.1. Experimental setup

Datasets. Following prior work (Shu et al., 2022; Feng et al., 2023; Karmanov et al., 2024), we evaluate out-of-distribution (OOD) performance on 4 datasets derived from ImageNet (Deng et al., 2009) and fine-grained (FG) classification on 10 diverse datasets spanning categories such as animals, scenes, and actions. Full dataset details are provided in the Appendix.

Baselines. We compare our method with the baseline CLIP-ViT-B/16 and few-shot learning methods — CoOp (Zhou et al., 2022b), CoCoOp (Zhou et al., 2022a) — as well as existing test-time prompt tuning methods — TPT (Shu et al., 2022) and C-TPT (Yoon et al., 2024). Additionally, we compare Image Encoder Tuning, which tunes the attention weight matrices of the image encoder without relying on LoRA; Layer Normalization Tuning, which tunes the normalization layer parameters of the image encoder, similar to WATT (Osowiechi et al., 2024) and CLIPArTT (Hakim et al., 2024); and MTA (Zanella & Ben Ayed, 2024b), which operates without backpropagation.

Implementation details. We follow a strict episodic test-time training setting, where the model is reset after each test instance and updated with at most one epoch of back-propagation, without using external models or cached data. We adopt CLIP-ViT-B/16 as the common backbone and apply LoRA only to the 11th and 12th transformer layers of the image encoder. The LoRA rank is set to 16, and the scale factor γ is set to 12 for the OOD benchmark and 2 for the fine-grained benchmark. We evaluate three variants of our method: **LoRA-TTT-M** (using the MEM loss only), **LoRA-TTT-A** (using the MAE loss only), and **LoRA-TTT**, which combines both with fixed weights, where the losses are weighted by $\lambda_1 = 1$ and $\lambda_2 = 16$. Additional implementation details are provided in the Appendix.

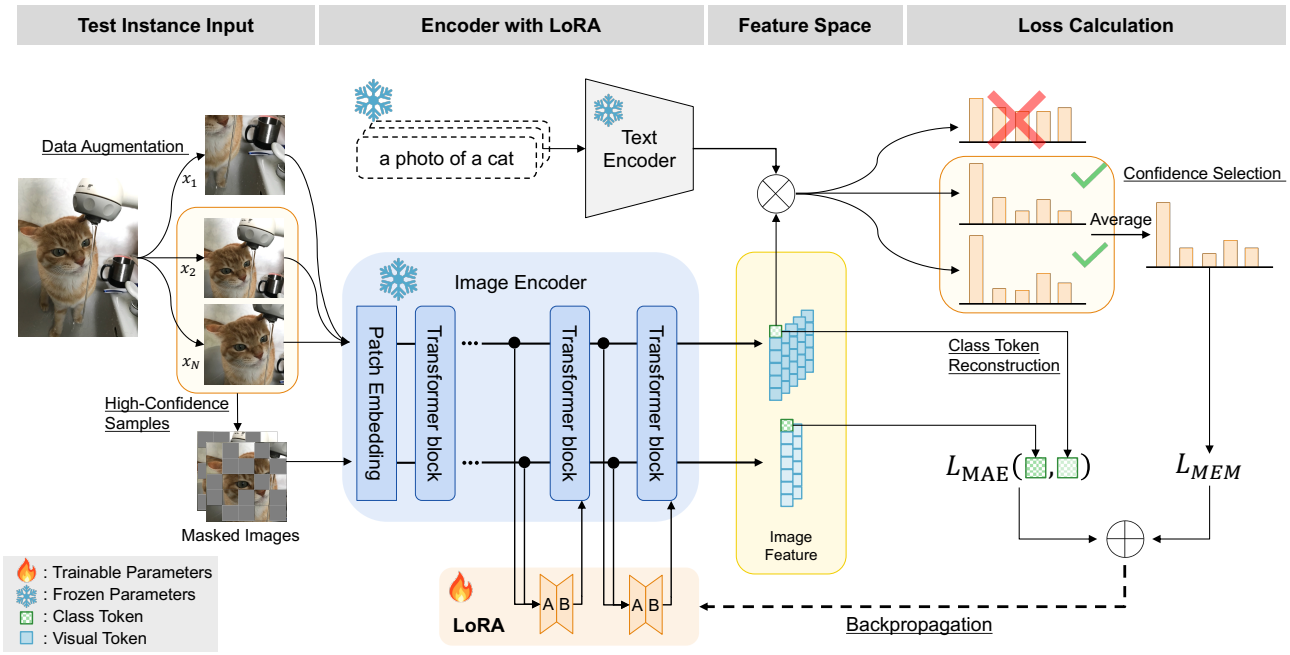


Figure 2: **LoRA-TTT** for zero-shot image classification. LoRA-TTT updates the LoRA parameters using MEM loss and MAE loss, calculated from the top 10% of high-confidence augmented views. This approach allows adaptation to domain shifts with low memory consumption while maintaining generalization ability.

3.2. Results

Zero-shot classification. In Table 1, LoRA-TTT improves the zero-shot generalization of CLIP-ViT-B/16, outperforming few-shot and text prompt tuning methods on both benchmarks. It achieves state-of-the-art results without using domain knowledge, external models, or cache, and consistently outperforms alternatives such as Image Encoder Tuning and Layer Normalization Tuning. These results highlight the effectiveness of LoRA tuning for bridging domain gaps in VLMs.

In the comparison of the two test-time losses, LoRA-TTT-M and LoRA-TTT-A, both methods show performance improvements over the baseline. LoRA-TTT-A (*i.e.*, MAE loss) achieves comparable or better results than LoRA-TTT-M (*i.e.*, MEM loss) across a wide range of category domains in the fine-grained benchmark, particularly demonstrating the versatility of MAE as a test-time loss in VLMs. Although our use of MAE loss involves only an image encoder without a decoder, it demonstrates that restoring global features of masked images contributes to understanding image context in VLMs. Combining the two losses further enhances both versatility and performance, surpassing TPT. For example, a comparison of TPT with LoRA-TTT-M on EuroSAT highlights the importance of tuning the text prompt when using only MEM loss, but the combination with MAE loss helps overcome this weakness.

Generalization in prompts. Table 2 shows the generalization performance when using either the ensemble of text prompts or CoOp. TPT is designed to initialize a single hard prompt, making it difficult to leverage performance gains from an ensemble of prompts. By combining our method with the ensemble, we achieve improvements comparable to CLIP-ViT-B/16 with the ensemble, while maintaining zero-shot conditions and surpassing prior prompt tuning methods. Even with CoOp, LoRA-TTT shows similar gains to CLIP-ViT-B/16 + CoOp, indicating it improves performance independently of prompt. As our method tunes only the image encoder, it allows flexible prompt design—an advantage in practical applications where freely choosing prompts is beneficial (Gu et al., 2023).

Calibration. As shown in Figure 3, $\mathcal{L}_{\text{MEM}}$, commonly used as a test-time loss in VLMs, is known to induce overconfidence (Guo et al., 2017; Yoon et al., 2024), where the model's predicted confidence exceeds its actual accuracy. Table 3 compares Expected Calibration Error (ECE) for our loss functions, TPT, and C-TPT. Since LoRA-TTT-A (*i.e.*, MAE loss) is not explicitly designed for confidence estimation, it preserves the model's original calibration and achieves ECE comparable to or better than C-TPT, which is specifically designed to improve TPT's calibration. From a TTT perspective, the MAE loss offers benefits not only in domain generalization but

Table 1: **Top-1 accuracy of zero-shot image classification on ImageNet, the OOD benchmark, and the fine-grained benchmark** using the default hard prompt. The results of CoCoOp are obtained from the TPT paper, while others are reproduced with our code. The best results under zero-shot conditions are highlighted in **bold**. Performance improvements over the zero-shot CLIP-ViT-B/16 are indicated with an upward blue arrow (↑blue) and a downward red arrow (↓red)

Method	ImageNet	OOD Avg.	Flower102	DTD	Pets	Cars	UCF101	Caltech	Food101	SUN397	Aircraft	EuroSAT	FG Avg.
CLIP-ViT-B/16	66.71	57.14	67.40	44.39	88.25	65.51	65.24	93.31	83.64	62.56	23.91	42.22	63.64
CoOp (Zhou et al., 2022b)	71.75	59.46	68.30	42.34	89.35	63.30	67.19	92.85	83.72	64.53	19.96	40.19	63.17
CoCoOp (Zhou et al., 2022a)	71.02	59.91	70.85	45.45	90.46	64.90	68.44	93.79	83.97	66.89	22.29	39.23	64.63
TPT (Shu et al., 2022)	69.02	60.89	68.98	45.92	87.27	67.02	**68.99**	93.55	**85.00**	**65.11**	23.76	43.44	64.91
C-TPT (Yoon et al., 2024)	68.50	59.55	**69.67**	44.80	88.47	65.97	65.27	93.35	83.23	64.28	23.97	42.21	64.12
MTA (Zanella & Ben Ayed, 2024b)	69.23	61.49	68.29	45.33	88.17	**68.08**	68.07	**94.12**	84.88	64.72	25.38	40.91	64.79
Image Encoder Tuning	64.26	59.89	59.03	42.91	83.81	66.60	67.57	88.64	82.11	62.71	23.67	36.67	61.37
Layer Normalization Tuning	66.93	57.45	67.48	44.33	88.36	65.80	65.45	93.31	83.79	62.83	24.00	42.22	63.76
LoRA-TTT-M (Ours)	69.21 (↑2.49)	62.78 (↑5.64)	67.60 (↑0.20)	**46.04** (↑1.65)	87.11 (↓1.14)	67.81 (↑2.30)	68.38 (↑3.15)	93.59 (↑0.28)	84.83 (↑1.19)	64.61 (↑2.05)	25.68 (↑1.77)	39.27 (↓2.95)	64.49 (↑0.85)
LoRA-TTT-A (Ours)	66.27 (↓0.45)	59.00 (↑1.86)	68.33 (↑0.93)	45.21 (↑0.83)	**88.72** (↑0.46)	66.94 (↑1.43)	66.35 (↑1.11)	93.71 (↑0.41)	84.39 (↑0.75)	63.63 (↑1.07)	25.38 (↑1.47)	**44.52** (↑2.30)	64.72 (↑1.08)
LoRA-TTT (Ours)	**69.40** (↑2.68)	**62.93** (↑5.79)	67.88 (↑0.49)	45.86 (↑1.48)	87.63 (↓0.63)	67.72 (↑2.20)	68.38 (↑3.15)	93.83 (↑0.53)	84.99 (↑1.35)	64.59 (↑2.03)	**25.92** (↑2.01)	43.23 (↑1.01)	**65.00** (↑1.36)

Table 2: **Generalization in prompts.** Performance differences from each hard prompt are indicated with an upward blue arrow (↑blue) and a downward red arrow (↓red).

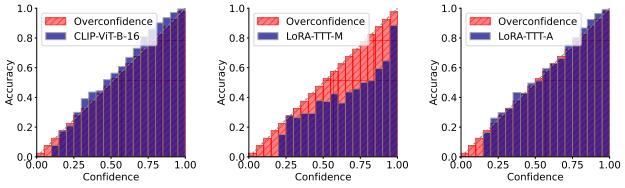
Method	ImageNet	OOD Average	FG Average
CLIP-ViT-B/16 + Ensemble	68.31 (↑1.59)	59.52 (↑2.38)	64.68 (↑1.04)
TPT + Ensemble	67.21 (↓1.81)	59.93 (↓0.96)	63.37 (↓1.54)
LoRA-TTT+ Ensemble (Ours)	**70.67** (↑1.27)	**64.99** (↑2.06)	**65.95** (↑0.95)
CLIP-ViT-B/16 + CoOp	71.75 (↑5.03)	59.46 (↑2.32)	63.17 (↓0.47)
TPT + CoOp	73.63 (↑4.62)	62.98 (↑2.09)	63.95 (↓0.95)
LoRA-TTT+ CoOp (Ours)	**74.03** (↑4.63)	**64.35** (↑1.42)	**64.00** (↓1.01)

also in calibration — an essential property in real-world applications such as healthcare and autonomous driving (Wang et al., 2022; Liu et al., 2023; Dorbala et al., 2022).

Efficiency. Runtime and memory consumption, shown in Figure 4a and Figure 4b, highlight the efficiency of our approach compared to TPT. By precomputing text features, LoRA-TTT removes the need for a text encoder during TTT, reducing model size and avoiding the bottleneck caused by text encoding. Although LoRA-TTT has more trainable parameters, applying LoRA only to the deeper layers of the image encoder reduces memory usage. LoRA-TTT-A further lowers memory consumption by computing loss on only 10% of augmented samples and masking half of the tokens. As a result, LoRA-TTT requires minimal overhead even with the MAE loss and achieves higher efficiency than TPT while delivering better performance. These properties make LoRA-TTT suitable for domains such as streaming data (Wang et al., 2023; Azimi et al., 2022), high-stakes applications (Wang et al., 2022; Liu et al., 2023; Dorbala et al., 2022; Khandelwal et al., 2022), and edge devices (Cai et al., 2020; Song et al., 2023).

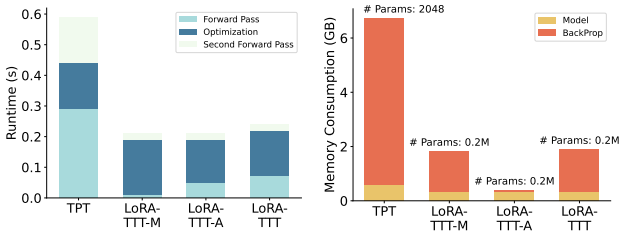
Table 3: **Expected Calibration Error** (↓).

Method	ImageNet	OOD Average	FG Average
CLIP-ViT-B/16	1.93	4.80	4.53
TPT	10.61	12.08	11.71
C-TPT	3.11	5.38	5.29
LoRA-TTT-M	20.32	22.76	19.73
LoRA-TTT-A	2.97	5.59	4.80
LoRA-TTT	14.04	16.49	12.75



(a) CLIP-ViT-B/16 (b) $\mathcal{L}_{\text{MEM}}$ (c) $\mathcal{L}_{\text{MAE}}$

Figure 3: **Comparison of calibration performance on the Cars dataset.** The MAE loss can improve performance while preserving the baseline model's output characteristics.



(a) Runtime (b) Memory Consumption

Figure 4: **TTT efficiency in ImageNet evaluation.**

4. Conclusion

This paper introduces LoRA-TTT, a test-time training method for VLMs that applies LoRA to adapt to distribution shifts without catastrophic forgetting. A lightweight reconstruction loss further improves generalization and calibration. Experiments show that LoRA-TTT outperforms text prompt tuning methods with lower memory and runtime costs, and without external resources, making it suitable for both high-stakes and resource-constrained environments.

Acknowledgements

This project was supported by Sony under the Visiting Industry Fellow program at UC San Diego. We also thank Jiteng Mu and Yinbo Chen for their valuable comments.

References

Azimi, F., Palacio, S., Raue, F., Hees, J., Bertinetto, L., and Dengel, A. Self-supervised test-time adaptation on video data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3439–3448, 2022.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pp. 446–461. Springer, 2014.

Cai, H., Gan, C., Zhu, L., and Han, S. Tinytl: Reduce memory, not parameters for efficient on-device learning. *Advances in Neural Information Processing Systems*, 33: 11285–11297, 2020.

Chen, D., Wang, D., Darrell, T., and Ebrahimi, S. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305, 2022.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

Dorbala, V. S., Sigurdsson, G., Piramuthu, R., Thomason, J., and Sukhatme, G. S. Clip-nav: Using clip for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2211.16649*, 2022.

Farina, M., Franchi, G., Iacca, G., Mancini, M., and Ricci, E. Frustratingly easy test-time adaptation of vision-language models. *arXiv preprint arXiv:2405.18330*, 2024.

Feng, C.-M., Yu, K., Liu, Y., Khan, S., and Zuo, W. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2704–2714, 2023.

Gandelsman, Y., Sun, Y., Chen, X., and Efros, A. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 35:29374–29385, 2022.

Gao, P., Lin, Z., Zhang, R., Fang, R., Li, H., Li, H., and Qiao, Y. Mimic before reconstruct: Enhancing masked autoencoders with feature mimicking. *International Journal of Computer Vision*, 132(5):1546–1556, 2024.

Gu, J., Han, Z., Chen, S., Beirami, A., He, B., Zhang, G., Liao, R., Qin, Y., Tresp, V., and Torr, P. A systematic survey of prompt engineering on vision-language foundation models. *arXiv preprint arXiv:2307.12980*, 2023.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Hakim, G. A. V., Osowiechi, D., Noori, M., Cheraghalikhani, M., Bahri, A., Yazdanpanah, M., Ayed, I. B., and Desrosiers, C. Clipartt: Adaptation of clip to new domains at test time. *arXiv preprint arXiv:2405.00754*, 2024.

Han, Z., Gao, C., Liu, J., Zhang, S. Q., et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Helber, P., Bischke, B., Dengel, A., and Borth, D. 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.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021a.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15262–15271, 2021b.

Hondru, V., Croitoru, F. A., Minaee, S., Ionescu, R. T., and Sebe, N. Masked image modeling: A survey. *arXiv preprint arXiv:2408.06687*, 2024.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Karmanov, A., Guan, D., Lu, S., El Saddik, A., and Xing, E. Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14162–14171, 2024.

Khandelwal, A., Weihs, L., Mottaghi, R., and Kembhavi, A. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14829–14838, 2022.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Kumar, A., Raghunathan, A., Jones, R., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

Li, F.-F., Andreeto, M., Ranzato, M., and Perona, P. Caltech 101. *CaltechDATA: Pasadena, CA, USA*, 2022.

Liang, J., He, R., and Tan, T. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, pp. 1–34, 2024.

Liu, J., Zhang, Y., Chen, J.-N., Xiao, J., Lu, Y., A Landman, B., Yuan, Y., Yuille, A., Tang, Y., and Zhou, Z. Clip-driven universal model for organ segmentation and tumor detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21152–21164, 2023.

Liu, J., Xu, R., Yang, S., Zhang, R., Zhang, Q., Chen, Z., Guo, Y., and Zhang, S. Continual-mae: Adaptive distribution masked autoencoders for continual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28653–28663, 2024.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Osowiechi, D., Noori, M., Hakim, G. A. V., Yazdanpanah, M., Bahri, A., Cheraghalikhani, M., Dastani, S., Beizaee, F., Ayed, I. B., and Desrosiers, C. Watt: Weight average test-time adaptation of clip. *arXiv preprint arXiv:2406.13875*, 2024.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, 2018.

Shu, M., Nie, W., Huang, D.-A., Yu, Z., Goldstein, T., Anandkumar, A., and Xiao, C. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35: 14274–14289, 2022.

Shu, Y., Guo, X., Wu, J., Wang, X., Wang, J., and Long, M. Clipood: Generalizing clip to out-of-distributions. In *International Conference on Machine Learning*, pp. 31716–31731. PMLR, 2023.

Song, J., Lee, J., Kweon, I. S., and Choi, S. Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11920–11929, 2023.

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

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019.

Wang, R., Sun, Y., Gandelsman, Y., Chen, X., Efros, A. A., and Wang, X. Test-time training on video streams. *arXiv preprint arXiv:2307.05014*, 2023.

Wang, Z., Wu, Z., Agarwal, D., and Sun, J. Medclip: Contrastive learning from unpaired medical images and text. *arXiv preprint arXiv:2210.10163*, 2022.

Wang, Z., Luo, Y., Zheng, L., Chen, Z., Wang, S., and Huang, Z. In search of lost online test-time adaptation: A survey. *International Journal of Computer Vision*, pp. 1–34, 2024.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7959–7971, 2022.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Xiao, Z., Shen, J., Derakhshani, M. M., Liao, S., and Snoek, C. G. Any-shift prompting for generalization over distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13849–13860, 2024.

Xin, Y., Luo, S., Zhou, H., Du, J., Liu, X., Fan, Y., Li, Q., and Du, Y. Parameter-efficient fine-tuning for pre-trained vision models: A survey. *arXiv preprint arXiv:2402.02242*, 2024.

Xu, L., Xie, H., Qin, S.-Z. J., Tao, X., and Wang, F. L. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*, 2023.

Yoon, H. S., Yoon, E., Tee, J. T. J., Hasegawa-Johnson, M., Li, Y., and Yoo, C. D. C-tpt: Calibrated test-time prompt tuning for vision-language models via text feature dispersion. *arXiv preprint arXiv:2403.14119*, 2024.

Zanella, M. and Ben Ayed, I. Low-rank few-shot adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1593–1603, 2024a.

Zanella, M. and Ben Ayed, I. On the test-time zero-shot generalization of vision-language models: Do we really need prompt learning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23783–23793, 2024b.

Zhang, M., Levine, S., and Finn, C. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

Zhang, Q., Chen, M., Bukharin, A., Karampatziakis, N., He, P., Cheng, Y., Chen, W., and Zhao, T. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

Zhao, S., Wang, X., Zhu, L., and Yang, Y. Test-time adaptation with clip reward for zero-shot generalization in vision-language models. *arXiv preprint arXiv:2305.18010*, 2023.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16816–16825, 2022a.

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

Zhu, Y., Shen, Z., Zhao, Z., Wang, S., Wang, X., Zhao, X., Shen, D., and Wang, Q. Melo: Low-rank adaptation is better than fine-tuning for medical image diagnosis. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5. IEEE, 2024.

Table 4: Comparison of VLM-focused TTT methods.

Method	External Module	Text Template	TTT Iterations	Trainable Parameter	Test Instances
ZERO	None	No restriction	0	None	1
TDA	Key-value Cache	No restriction	0	None	1
CLIPArTT	None	No restriction	10 (default)	Layer Normalization	128 (Multiple required)
WATT	None	Diverse set of templates	Multiple required	Layer Normalization	128 (default)
TPT	None	single template	1	Text Prompt	1
LoRA-TTT	None	No restriction	1	LoRA	1

A. Broader Impact

Test-Time Training (TTT) for Vision-Language Models (VLMs) is crucial for enhancing their generalization ability and broadening their applicability to real-world AI applications. This study introduces a novel method that achieves strong zero-shot generalization across diverse categories. Our approach enables the development of systems that can adapt to various environments, ranging from memory-constrained edge devices to high-stakes applications, thereby making VLMs more versatile and practical in real-world scenarios. We hope that Parameter-Efficient Fine-Tuning (*e.g.*, LoRA) will play a pivotal role in TTT, inspiring future research aimed at improving the performance of foundation models.

B. Limitations

Our method has limitations that should be addressed in future work. The MEM loss is primarily designed for image classification, making its adaptation to other tasks, such as object detection or segmentation, challenging. In contrast, the MAE loss is task-agnostic, and extending it to such tasks is a promising direction. Additionally, LoRA hyperparameters (e.g., r and γ) require careful tuning as their optimal values depend on the target domain. Developing a mechanism to dynamically adjust these parameters based on domain characteristics could improve adaptability and performance.

C. Related Work

Test-Time Training (TTT) allows models to adapt to distribution shifts between training and test data during inference through dynamic parameter updates (Liang et al., 2024; Wang et al., 2024; Chen et al., 2022). The challenges in this area lie in designing an effective test-time objective without labels and developing an efficient system suitable for real-world deployment. For example, TENT (Wang et al., 2020) tunes batch normalization statistics at test time using entropy loss; however, this approach requires batch processing rather than instance-level processing, making it challenging to handle sequential data in real-time. In contrast, MEMO (Zhang et al., 2022) computes test loss from a single instance, a strategy we extend to VLMs. Sun *et al.* (Sun et al., 2020) and Gandelsman *et al.* (Gandelsman et al., 2022) update the image encoder by introducing auxiliary tasks and applying self-supervision; however, these methods require fine-tuning the model with auxiliary tasks beforehand for TTT. Our approach eliminates this need, allowing for direct adaptation of pre-trained VLMs without additional pre-training steps. We demonstrate that our reconstruction loss enhances performance on foundation models like CLIP, offering a simple yet effective alternative to prior methods.

TTT for VLMs. TPT (Shu et al., 2022) focuses on optimizing a text prompt at test time, valued for its simplicity and effectiveness. It demonstrates that augmenting a single test instance and calculating marginal entropy minimization (Zhang et al., 2022) serves as an effective loss for VLMs. DiffTPT (Feng et al., 2023) utilizes stable diffusion to enhance data augmentation quality, while C-TPT (Yoon et al., 2024) is a technique that calibrates TPT to improve reliability. While text prompt tuning remains the predominant approach in TTT for VLMs, some methods instead focus on adapting the image encoder. RLCF (Zhao et al., 2023) tunes the image encoder and demonstrates that CLIP-ViT-B can achieve performance comparable to CLIP-ViT-L but requires CLIP-ViT-L as a feedback source, which poses challenges in memory-constrained environments. As shown in Table 4, WATT (Osowiechi et al., 2024) and CLIPArTT (Hakim et al., 2024) tune the layer normalization parameters of the vision encoder; however, WATT relies on a diverse set of text templates, while CLIPArTT requires multiple test instances, imposing significant constraints on real-world applicability. Moreover, both methods update these parameters across all layers, leading to high computational costs and requiring multiple backpropagation steps. In contrast, our method tunes only the two layers closest to the output, significantly improving computational efficiency and

Table 5: **The details of the datasets used in the experiments.**

Dataset	# Classes	Test set size
ImageNet (Deng et al., 2009)	1,000	50,000
ImageNet-A (Hendrycks et al., 2021b)	200	7,500
ImageNetV2 (Recht et al., 2019)	1,000	10,000
ImageNet-R (Hendrycks et al., 2021a)	200	30,000
ImageNet-Sketch (Wang et al., 2019)	1,000	50,889
Flowers102 (Nilsback & Zisserman, 2008)	102	2,463
DTD (Cimpoi et al., 2014)	47	1,692
OxfordPets (Parkhi et al., 2012)	37	3,669
StanfordCars (Krause et al., 2013)	196	8,041
UCF101 (Soomro et al., 2012)	101	3,783
Caltech101 (Li et al., 2022)	100	2,465
Food101 (Bossard et al., 2014)	101	30,300
SUN397 (Xiao et al., 2010)	397	19,850
FGVCAircraft (Maji et al., 2013)	100	3,333
EuroSAT (Helber et al., 2019)	10	8,100

enabling faster backpropagation. Additionally, lightweight, backpropagation-free methods such as ZERO (Farina et al., 2024) and TDA (Karmanov et al., 2024) have also been proposed. ZERO offers low computational overhead but struggles with generalization performance compared to TPT. While TDA is efficient, it relies on a key-value cache. In contrast, our method adapts to a single test instance in one step, without relying on external modules. This ensures feasibility even in closed, memory-constrained environments such as edge devices, where external resources and cached data are unavailable.

Application of Low-rank adaptation (LoRA) aims to achieve efficient fine-tuning of large models with vast numbers of parameters in memory-constrained environments by introducing trainable low-rank matrices into each layer of the Transformer architecture, allowing the pre-trained parameters to remain frozen (Hu et al., 2021; Han et al., 2024; Xin et al., 2024). MeLo (Zhu et al., 2024) demonstrates that applying LoRA to vision transformers (ViT) for downstream medical image diagnosis achieves comparable performance to fully fine-tuned ViT models while significantly reducing memory consumption. CLIP-LoRA (Zanella & Ben Ayed, 2024a) demonstrate significant performance improvements in few-shot learning by applying LoRA to the vision encoder of CLIP. However, CLIP-LoRA requires a few labeled samples from the target downstream task.

D. Experiments Details

D.1. Datasets

The evaluation includes out-of-distribution testing on ImageNet and its four variants, as well as fine-grained classification assessments across categories derived from 10 different datasets. The details of the datasets are provided in Table 5.

D.2. Detailed Implementation Settings

Backbone and Optimization. We adopt the pre-trained CLIP-ViT-B/16 as the common backbone architecture. LoRA parameters are optimized in a single step using the AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 0.001 and weight decay of 0.2. All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24GB of memory.

LoRA Configuration. LoRA is applied exclusively to the transformer architecture in layers 11 and 12 of the image encoder with a rank of 16, targeting the key, query, value, and output projection matrices. The scale factor γ is set to 12 for the OOD benchmark and 2 for the fine-grained benchmark. Matrix $\mathbf{A}$ is initialized using Kaiming-uniform (He et al., 2015), while matrix $\mathbf{B}$ is initialized to zero.

Table 6: The 80 hand-crafted text prompts.

“a bad photo of a { class }”, “a photo of many { class }”, “a sculpture of a { class }”, “a photo of the hard to see { class }”, “a low resolution photo of the { class }”, “a rendering of a { class }”, “graffiti of a { class }”, “a bad photo of the { class }”, “a cropped photo of the { class }”, “a tattoo of a { class }”, “the embroidered { class }”, “a photo of a hard to see { class }”, “a bright photo of a { class }”, “a photo of a clean { class }”, “a photo of a dirty { class }”, “a dark photo of the { class }”, “a drawing of a { class }”, “a photo of my { class }”, “the plastic { class }”, “a photo of the cool { class }”, “a close-up photo of a { class }”, “a black and white photo of the { class }”, “a painting of the { class }”, “a painting of a { class }”, “a pixelated photo of the { class }”, “a sculpture of the { class }”, “a bright photo of the { class }”, “a cropped photo of a { class }”, “a plastic { class }”, “a photo of the dirty { class }”, “a jpeg corrupted photo of a { class }”, “a blurry photo of the { class }”, “a photo of the { class }”, “a good photo of the { class }”, “a rendering of the { class }”, “a { class } in a video game”, “a photo of one { class }”, “a doodle of a { class }”, “a close-up photo of the { class }”, “a photo of a { class }”, “the origami { class }”, “the { class } in a video game”, “a sketch of a { class }”, “a doodle of the { class }”, “a origami { class }”, “a low resolution photo of a { class }”, “the toy { class }”, “a rendition of the { class }”, “a photo of the clean { class }”, “a photo of a large { class }”, “a rendition of a { class }”, “a photo of a nice { class }”, “a photo of a weird { class }”, “a blurry photo of a { class }”, “a cartoon { class }”, “art of a { class }”, “a sketch of the { class }”, “a embroidered { class }”, “a pixelated photo of a { class }”, “itap of the { class }”, “a jpeg corrupted photo of the { class }”, “a good photo of a { class }”, “a plushie { class }”, “a photo of the nice { class }”, “a photo of the small { class }”, “a photo of the weird { class }”, “the cartoon { class }”, “art of the { class }”, “a drawing of the { class }”, “a photo of the large { class }”, “a black and white photo of a { class }”, “the plushie { class }”, “a dark photo of a { class }”, “itap of a { class }”, “graffiti of the { class }”, “a toy { class }”, “itap of my { class }”, “a photo of a cool { class }”, “a photo of a small { class }”, “a tattoo of the { class }”.

Baselines. For text prompt tuning baselines, we use 4 trainable text tokens initialized with the hard prompt “a photo of a”. We prepare three versions of precomputed prompts: (1) the default hard prompt, (2) an ensemble of 80 hand-crafted prompts (Radford et al., 2021), and (3) CoOp (Zhou et al., 2022b), which uses 4 tokens and is pre-trained on ImageNet with 16-shot supervision. The 80 hand-crafted prompts are listed in Table 6.

Other Tuning Methods. For Image Encoder Tuning, we directly tune the key, query, value, and output matrices in layers 11 and 12 of the image encoder using the same optimizer and loss configuration as LoRA-TTT. For Layer Normalization Tuning, we tune only the layer normalization parameters in the same layers with identical settings.

Test-time Augmentation. Following TPT (Shu et al., 2022), we expand a single test instance into a batch of 64 using random resized crops (including the original instance). To suppress noise, we select the top 10% of high-confidence samples from the batch for computing the test loss.

D.3. MAE loss variants

In this section, we provide details about the variants of the MAE loss. In addition to the MAE loss applied in LoRA-TTT, we explore the following approaches, as illustrated in Figure 5. The loss $L_{\text{MAE}}^{\text{vis, enc}}$ calculates the mean squared error of unmasked visual tokens after image encoding. The loss $L_{\text{MAE}}^{\text{cls, dec}}$ reconstructs class tokens following the decoding process. The loss $L_{\text{MAE}}^{\text{pix, dec}}$ rearranges the visual tokens obtained after decoding back into image pixels, enabling pixel-level reconstruction. This method of calculating $L_{\text{MAE}}^{\text{pix, dec}}$ is consistent with traditional TTT approaches based on MAE (Gandelsman et al., 2022; Wang et al., 2023). These methodologies provide diverse perspectives on leveraging MAE loss for effective reconstruction.

D.4. Evaluation metric

We use the Expected Calibration Error (ECE) (Naeini et al., 2015; Yoon et al., 2024) as a metric to evaluate the calibration performance of the model in image classification. ECE is calculated on a given evaluation dataset by dividing the model's outputs into equally sized bins based on prediction confidence and measuring the discrepancy between the predicted probabilities and the true probabilities within each bin. A well-calibrated model exhibits a smaller gap between predicted

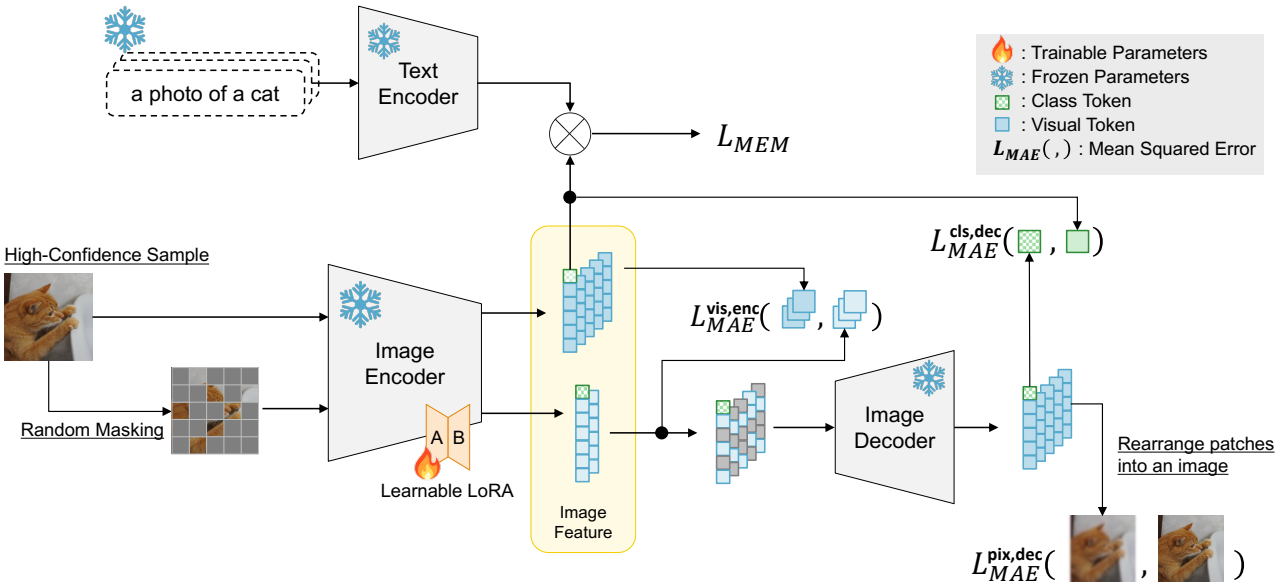


Figure 5: **Variants of MAE Loss.**

Table 7: **Top1 accuracy of zero-shot image classification on the OOD benchmark** when using the default hard prompt. The results of CoCoOp are obtained from the TPT paper, while others are reproduced with our code. The best results under zero-shot conditions are highlighted in **bold**. Performance improvements over the zero-shot CLIP-ViT-B/16 are indicated with an upward blue arrow (↑blue) and a downward red arrow (↓red).

Method	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-Sketch	Average	OOD Avg.
CLIP-ViT-B/16	66.71	47.80	60.63	73.99	46.15	59.06	57.14
CoOp (Zhou et al., 2022b)	71.75	50.13	64.51	75.28	47.92	61.92	59.46
CoCoOp (Zhou et al., 2022a)	71.02	50.63	64.07	76.18	48.75	62.13	59.91
TPT (Shu et al., 2022)	69.02	54.73	63.70	77.15	47.99	62.52	60.89
C-TPT (Yoon et al., 2024)	68.50	51.60	62.70	76.00	47.90	61.34	59.55
MTA (Zanella & Ben Ayed, 2024b)	69.23	56.87	63.67	76.88	48.54	63.04	61.49
Image Encoder Tuning	64.26	56.31	59.70	75.89	47.65	60.76	59.89
Layer Normalization Tuning	66.93	48.24	60.94	74.31	46.31	59.35	57.45
LoRA-TTT-M (Ours)	69.21 (↑2.49)	**60.57** (↑12.77)	64.28 (↑3.65)	77.53 (↑3.54)	48.73 (↑2.57)	64.06 (↑5.01)	62.78 (↑5.64)
LoRA-TTT-A (Ours)	66.27 (↓0.45)	52.55 (↑4.75)	60.87 (↑0.24)	75.57 (↑1.58)	47.01 (↑0.85)	60.45 (↑1.39)	59.00 (↑1.86)
LoRA-TTT (Ours)	**69.40** (↑2.68)	60.52 (↑12.72)	**64.43** (↑3.80)	**77.84** (↑3.85)	**48.94** (↑2.79)	**64.23** (↑5.17)	**62.93** (↑5.79)

confidence and actual accuracy, resulting in a lower ECE value. The ECE is computed as follows:

$$\mathrm{ECE} = \sum_{k=1}^{K} \frac{|B_k|}{m} \left|\mathrm{acc}(B_k) - \mathrm{conf}(B_k)\right|, \tag{2}$$

where K represents the number of bins, $|B_k|$ denotes the number of samples in bin k, $\mathrm{acc}(B_k)$ is the average accuracy of the samples in bin k, and $\mathrm{conf}(B_k)$ represents the average prediction confidence of the samples in bin k. In our experiments, the number of bins is set to 20.

E. Additional Experiments

E.1. Zero-shot classification

Table 7 shows the results on all datasets in the OOD benchmark.

Table 8: Error analysis of top-1 accuracy in zero-shot image classification on the OOD benchmark.

Method	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-Sketch	Average	OOD Avg.
CLIP-ViT-B/16	66.71	47.80	60.63	73.99	46.15	59.06	57.14
TPT (Shu et al., 2022)	69.02 (±.14)	54.73 (±.11)	63.70 (±.09)	77.15 (±.06)	47.99 (±.04)	62.52 (±.03)	60.89 (±.04)
LoRA-TTT-M (Ours)	69.21 (±.05)	60.57 (±.16)	64.28 (±.08)	77.53 (±.09)	48.73 (±.04)	64.06 (±.02)	62.78 (±.02)
LoRA-TTT-A (Ours)	66.27 (±.11)	52.55 (±.35)	60.87 (±.19)	75.57 (±.09)	47.01 (±.08)	60.45 (±.06)	59.00 (±.08)
LoRA-TTT (Ours)	69.40 (±.08)	60.52 (±.19)	64.43 (±.12)	77.84 (±.03)	48.94 (±.05)	64.23 (±.01)	62.93 (±.02)

Table 9: Error analysis of top-1 accuracy in zero-shot image classification on the fine-grained benchmark.

Method	Flower102	DTD	Pets	Cars	UCF101	Caltech	Food101	SUN397	Aircraft	EuroSAT	FG Avg.
CLIP-ViT-B/16	67.40	44.39	88.25	65.51	65.24	93.31	83.64	62.56	23.91	42.22	63.64
TPT (Shu et al., 2022)	68.98 (±.18)	45.92 (±.33)	87.27 (±.20)	67.02 (±.14)	68.99 (±.15)	93.55 (±.22)	85.00 (±.06)	65.11 (±.08)	23.76 (±.36)	43.44 (±.08)	64.91 (±.04)
LoRA-TTT-M (Ours)	67.60 (±.33)	46.04 (±.25)	87.11 (±.19)	67.81 (±.16)	68.38 (±.07)	93.59 (±.13)	84.83 (±.13)	64.61 (±.09)	25.68 (±.12)	39.27 (±.23)	64.49 (±.08)
LoRA-TTT-A (Ours)	68.33 (±.02)	45.21 (±.07)	88.72 (±.13)	66.94 (±.02)	66.35 (±.34)	93.71 (±.02)	84.39 (±.05)	63.63 (±.12)	25.38 (±.20)	44.52 (±.15)	64.72 (±.04)
LoRA-TTT (Ours)	67.88 (±.22)	45.86 (±.12)	87.63 (±.06)	67.72 (±.03)	68.38 (±.12)	93.83 (±.16)	84.99 (±.05)	64.59 (±.11)	25.92 (±.39)	43.23 (±.33)	65.00 (±.06)

E.2. Error analysis

Table 8 and Table 9 present the standard deviation of three random runs with different seeds for zero-shot image classification on the OOD and fine-grained benchmarks, respectively. The randomness of LoRA-TTT-M mainly stems from random data augmentation and one-step optimization, similar to TPT. Additionally, LoRA-TTT-A introduces an additional source of randomness through its masking strategy. Nevertheless, our method achieves an error magnitude comparable to that of TPT.

E.3. Expected Calibration Error

Table 10 presents the calibration results on the OOD benchmark, while Table 11 shows the results on the fine-grained benchmark for each dataset. The comparison includes our method, TPT (Shu et al., 2022), and C-TPT (Yoon et al., 2024). The results show that LoRA-TTT-A (*i.e.*, MAE loss) achieves calibration performance comparable to or surpassing that of C-TPT across a wide range of categories, highlighting the effective calibration properties of MAE loss.

E.4. Hyper-parameter tuning and sensitivity

Figure 6a shows that adding the MAE loss improves performance on fine-grained datasets without degrading performance on OOD datasets. We chose $\lambda_1 = 1$ and $\lambda_2 = 16$ for their consistent strong results across datasets. In Figures 6b to 6d, increasing the number of data augmentations tends to enhance performance; however, for efficiency, we chose 64, aligning with TPT. As N_p increases, performance tends to decrease, which is consistent with TPT's results. Therefore, following TPT, we select the top 10% ($N_p = 6$). Additionally, AugMix proves to be effective for data augmentation.

We chose different LoRA scales for the two benchmarks because only ImageNet-A exhibited distinct behavior depending on γ, as shown in Table 12. The relationship between this dataset and LoRA parameters requires further investigation. Our method consistently achieves strong performance and outperforms TPT on both benchmarks with $r = 16$ and $\gamma = 2$, while also using the same masking ratio and LoRA layer settings. Hyperparameter sensitivity is inherent in TTT methods. For example, the optimal iteration count in WATT (Osowiechi et al., 2024) and CLIPArTT (Hakim et al., 2024) varies by domain. Our approach generalizes well across multiple benchmarks, highlighting its stability with the fixed configuration.

E.5. Scalability Analysis of Our Method

In this section, we evaluate the scalability of our proposed method by applying it to a larger baseline model. Table 13 and Table 14 show the results obtained using the pretrained CLIP-ViT-L/14 on the OOD and fine-grained benchmarks, respectively. LoRA is applied exclusively to the transformer architecture in layers 23 and 24 of the image encoder, targeting the key, query, value, and output matrices with a rank of 16, and the LoRA scale γ is set to 2. All other experimental parameters are consistent with those in the main paper. The results demonstrate that LoRA-TTT consistently outperforms the baseline CLIP-ViT-L/14 across both benchmarks and multiple categories while maintaining the zero-shot setting. It also demonstrates performance improvements when combined with the ensemble text prompts, exhibiting generalization properties to text prompts similar to those observed with CLIP-ViT-B/16. Performance improvements are observed with

Table 10: **Expected Calibration Error (ECE↓)** of zero-shot image classification with TTT on the OOD benchmark. The best results, except for the baseline, are highlighted in **bold**.

Method	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-Sketch	Average	OOD Avg.
CLIP-ViT-B/16	1.93	8.37	2.51	3.53	4.79	4.23	4.80
TPT (Shu et al., 2022)	10.61	15.35	11.85	4.97	16.14	11.78	12.08
C-TPT (Yoon et al., 2024)	3.11	**6.40**	4.64	2.80	7.69	**4.93**	**5.38**
LoRA-TTT-M (Ours)	20.32	25.47	22.66	12.65	30.25	22.27	22.76
LoRA-TTT-A (Ours)	**2.97**	9.60	**4.13**	**1.35**	**7.28**	5.07	5.59
LoRA-TTT (Ours)	14.04	19.27	16.19	8.08	22.45	16.00	16.49

Table 11: **Expected Calibration Error (ECE↓)** of zero-shot image classification with TTT on the fine-grained benchmark. The best results, except for the baseline, are highlighted in **bold**.

Method	Flower102	DTD	Pets	Cars	UCF101	Caltech	Food101	SUN397	Aircraft	EuroSAT	Average
CLIP-ViT-B/16	3.21	8.23	4.41	4.45	2.93	5.12	2.03	2.24	5.50	7.16	4.53
TPT (Shu et al., 2022)	13.57	23.45	6.18	5.92	11.65	3.60	4.49	11.94	17.81	18.48	11.71
C-TPT (Yoon et al., 2024)	5.24	13.77	**1.56**	**1.56**	**2.30**	**3.27**	3.31	5.02	**4.41**	12.47	5.29
LoRA-TTT-M (Ours)	24.27	34.64	10.97	16.96	18.91	4.61	11.96	20.80	25.45	28.70	19.73
LoRA-TTT-A (Ours)	**4.10**	**12.27**	3.08	2.20	3.52	4.09	**1.83**	**3.01**	6.51	**7.34**	**4.80**
LoRA-TTT (Ours)	19.54	26.05	6.68	7.73	11.30	2.31	7.94	13.15	16.76	16.02	12.75

both types of loss (*i.e.*, LoRA-TTT-M and LoRA-TTT-A), highlighting the robustness of our method and its scalability to larger baseline models.

F. Ablation Study

F.1. How to apply LoRA for TTT

In this section, we explore the utilization of LoRA for TTT. We investigate the key factors for effectively applying LoRA, including: (1) determining the optimal layers and the extent of LoRA application within the transformer model, (2) understanding the relationship between the appropriate rank and scale, and (3) selecting the attention matrices for tuning.

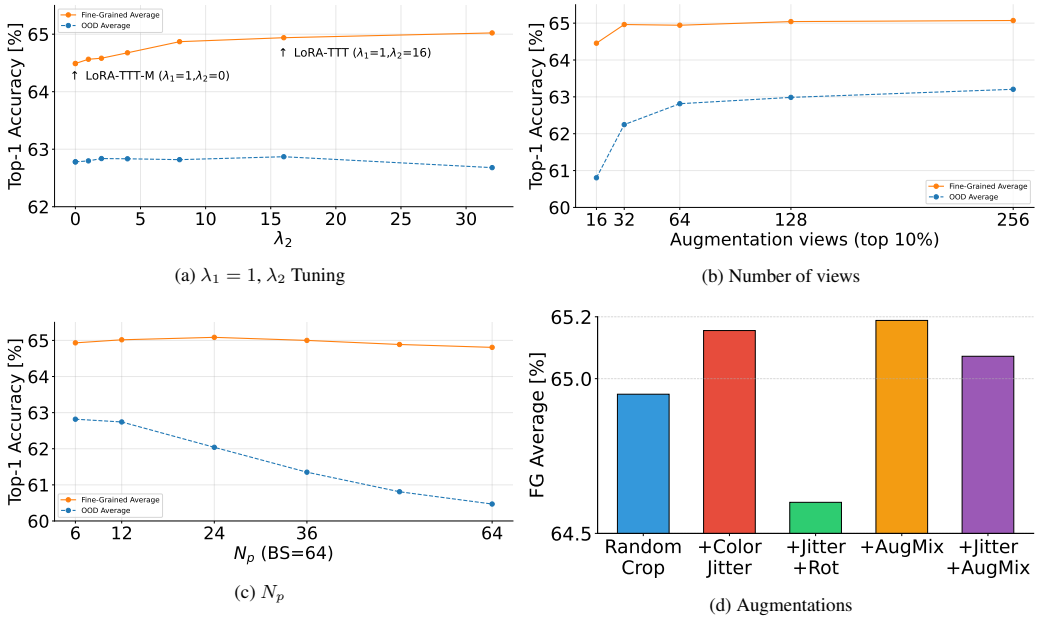
Which layers should we apply LoRA to? Table 15 presents the zero-shot classification performance when LoRA is applied to specific layers of the image encoder in CLIP-ViT-B/16. Our results indicate that applying LoRA to deeper layers is more effective than to shallower ones, aligning with trends observed in fine-tuning language models (Zhang et al., 2023). Additionally, applying LoRA to more layers does not necessarily improve performance. Limiting its application to the 11th and 12th layers not only outperforms applying it across all layers in terms of performance but also reduces memory consumption and runtime, making our approach more efficient for TTT.

LoRA rank and scale. As shown in Figure 7a, increasing the rank does not directly lead to performance gains. Each rank has an optimal scale, and as the rank increases, the corresponding optimal scale tends to decrease. When the rank is small (*e.g.*, rank 4), performance remains stable across different scales, reducing the need for extensive hyperparameter tuning.

LoRA rank and attention matrices. We investigate the optimal application of LoRA to different attention matrices in CLIP-ViT-B/16. In Figure 7b, we observe that applying LoRA to $\boldsymbol{W}_{\text{v}}$ at the same rank achieves the best results among the 4 matrices ($\boldsymbol{W}_{\text{o}}$, $\boldsymbol{W}_{\text{v}}$, $\boldsymbol{W}_{\text{q}}$, and $\boldsymbol{W}_{\text{k}}$). This trend aligns with previous research (Zhang et al., 2023; Zanella & Ben Ayed, 2024a), even in the context of TTT. Given the same total number of parameters, applying LoRA to $\boldsymbol{W}_{\text{kvqo}}$ shows little difference in performance compared to applying it to $\boldsymbol{W}_{\text{vq}}$ or $\boldsymbol{W}_{\text{kq}}$.

Table 12: **Top-1 accuracy of zero-shot image classification.**

Method	ImageNet	ImageNet-A	OOD Average	FG Average
TPT	69.02	54.73	60.89	64.91
LoRA-TTT ($r = 16$, $\gamma = 2$)	69.31	55.43	61.21	65.00
LoRA-TTT ($r = 16$, $\gamma = 12$)	69.40	60.52	62.93	64.39



(a) $\lambda_1 = 1$, λ_2 Tuning

(b) Number of views

(c) N_p

(d) Augmentations

Figure 6: **Hyper-parameter tuning.** Results may slightly vary due to trial randomness, even with the same parameters.

F.2. Masking strategy

In masked image modeling, the mask strategy plays a crucial role (Hondru et al., 2024; Gao et al., 2024). We examine the effects of the masking ratio, the confidence selection cutoff, the use of an image decoder, and the impact of reconstruction targets. We use a randomly initialized transformer-based decoder with 8 layers, 16 heads, and a 768 embedding size, without prior fine-tuning to ensure a fair evaluation. This decoder allows us to incorporate the pixel-wise reconstruction loss proposed in TTT methods based on MAE (Gandelsman et al., 2022; Wang et al., 2023).

As shown in Table 16, while the masking ratio does not significantly affect the overall performance, we choose a default masking ratio of 50% as it strikes a good balance between performance and computational efficiency. As proposed in TPT, selecting and masking the top 10% of augmented images with the lowest entropy yields better performance than masking all 64 images (*i.e.*, applying a cutoff of 1), with an improvement of over 1% observed in the OOD average. The 10% cutoff not only improves performance but also enhances the computational efficiency of TTT by calculating the loss on only one-tenth of the images. Furthermore, reconstructing the class token is more effective than reconstructing masked visual tokens or image pixels using the decoder. This supports the hypothesis that improving zero-shot image classification performance in VLMs relies more on aligning high-level semantics than on capturing fine-grained features.

F.3. Initialization of LoRA weights

LoRA demonstrates high effectiveness and efficiency for TTT, even when initialized with random weights. In this section, we explore the performance gains achieved by fine-tuning the LoRA weights before TTT. We prepare a third dataset, CC3M (Sharma et al., 2018), for LoRA initialization and train only the LoRA weights using the same contrastive loss as in CLIP pre-training (Radford et al., 2021) with image-text pairs. We employ Adam with a learning rate of 1e-6 and a weight decay of 0.05 for optimization, performing one epoch of training with a batch size of 64.

Table 13: **Top-1 accuracy of zero-shot image classification on the OOD benchmark with the CLIP-ViT-L/14 baseline**. Performance improvements over the zero-shot CLIP-ViT-L/14 are indicated with an upward blue arrow (↑blue) and a downward red arrow (↓red).

Method	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-Sketch	Average	OOD Avg.
CLIP-ViT-L/14	73.45	68.77	67.75	85.41	57.82	70.64	69.94
CLIP-ViT-L/14 + Ensemble	75.53	70.75	69.70	87.85	59.60	72.69	71.97
LoRA-TTT-M (Ours)	75.20 (↑1.75)	73.73 (↑4.96)	69.74 (↑1.99)	87.69 (↑2.28)	59.76 (↑1.94)	73.22 (↑2.58)	72.73 (↑2.79)
LoRA-TTT-A (Ours)	73.88 (↑0.43)	69.91 (↑1.13)	67.98 (↑0.23)	85.99 (↑0.58)	58.30 (↑0.49)	71.21 (↑0.57)	70.55 (↑0.61)
LoRA-TTT (Ours)	75.07 (↑1.61)	72.56 (↑3.79)	69.24 (↑1.49)	87.27 (↑1.86)	59.48 (↑1.67)	72.72 (↑2.08)	72.14 (↑2.20)
LoRA-TTT + Ensemble (Ours)	77.03 (↑3.57)	75.63 (↑6.85)	71.86 (↑4.11)	89.73 (↑4.32)	61.41 (↑3.59)	75.13 (↑4.49)	74.66 (↑4.72)

Table 14: **Top-1 accuracy of zero-shot image classification on the fine-grained benchmark with the CLIP-ViT-L/14 baseline**. Performance improvements over the zero-shot CLIP-ViT-L/14 are indicated with an upward blue arrow (↑blue) and a downward red arrow (↓red).

Method	Flower102	DTD	Pets	Cars	UCF101	Caltech	Food101	SUN397	Aircraft	EuroSAT	FG Avg.
CLIP-ViT-L/14	76.21	52.42	93.05	76.91	73.72	95.17	88.58	67.68	30.03	55.09	70.89
CLIP-ViT-L/14 + Ensemble	75.92	54.73	93.05	77.78	75.89	95.62	89.20	70.15	31.86	51.70	71.59
LoRA-TTT-M (Ours)	76.45 (↑0.24)	54.14 (↑1.71)	93.81 (↑0.76)	78.34 (↑1.43)	75.23 (↑1.51)	95.05 (↓0.12)	89.32 (↑0.74)	68.97 (↑1.29)	33.30 (↑3.27)	52.32 (↓2.77)	71.69 (↑0.81)
LoRA-TTT-A (Ours)	76.65 (↑0.45)	52.72 (↑0.30)	93.43 (↑0.38)	77.42 (↑0.51)	74.17 (↑0.45)	95.13 (↓0.04)	88.90 (↑0.32)	67.81 (↑0.13)	30.42 (↑0.39)	55.01 (↓0.07)	71.17 (↑0.28)
LoRA-TTT (Ours)	76.57 (↑0.37)	54.14 (↑1.71)	93.87 (↑0.82)	78.31 (↑1.41)	74.83 (↑1.11)	95.54 (↑0.37)	89.34 (↑0.77)	68.72 (↑1.04)	33.12 (↑3.09)	53.74 (↓1.35)	71.82 (↑0.93)
LoRA-TTT + Ensemble (Ours)	75.92 (↓0.28)	55.08 (↑2.66)	93.08 (↑0.03)	79.38 (↑2.47)	76.79 (↑3.07)	95.94 (↑0.77)	89.79 (↑1.21)	71.13 (↑3.45)	35.34 (↑5.32)	52.19 (↓2.90)	72.46 (↑1.58)

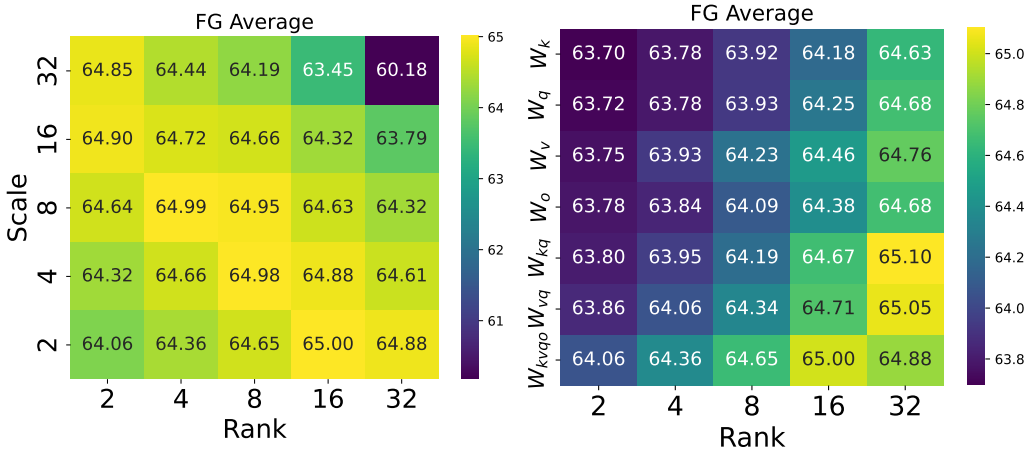
As shown in Figure 8, LoRA initialization using 21k randomly sampled image-text pairs from CC3M (*i.e.*, only 1% of the total CC3M dataset) improves performance by more than 1% on the fine-grained benchmark and by 0.6% on the OOD benchmark. Furthermore, TTT consistently improves performance on both the benchmarks, regardless of the LoRA initialization. Our experiments demonstrate that fine-tuning LoRA with a small amount of data shows the potential to enhance its performance. While adhering to the constraints of not leveraging domain-specific information or a teacher model, LoRA fine-tuning delivers significant performance improvements in TTT, establishing it as an effective approach for future applications of LoRA in TTT.

G. Qualitative Analysis

Table 17 shows the t-SNE visualization of image features after the image encoder for various evaluation datasets, comparing the baseline CLIP-ViT-B/16 and our method. The results show that our approach achieves better class separation than the baseline, indicating improved classification performance on the test data. Additionally, the visualizations highlight that the type of test loss affects how class separation is achieved.

Table 15: **Layers for LoRA application.**

LoRA Layer	ImageNet	OOD Average	FG Average
12	**69.59**	62.65	64.68
11-12	69.40	**62.93**	**65.00**
9-12	68.97	62.86	64.73
5-8	66.88	61.34	64.83
1-4	67.99	60.69	64.56
All	68.12	62.54	64.62



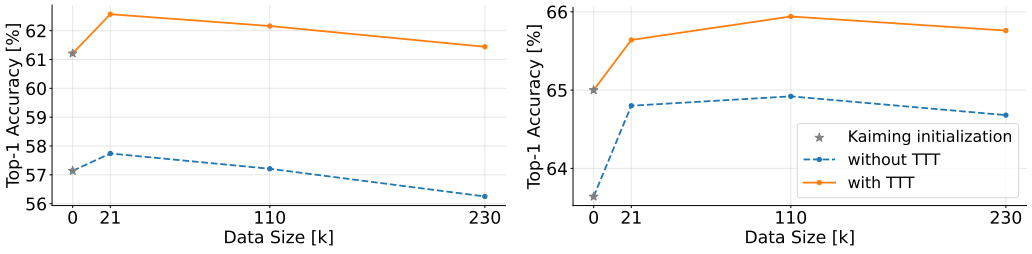
(a) LoRA rank and scale

(b) Attention matrices

Figure 7: **Impact of LoRA application design.** The average top-1 accuracy on the fine-grained benchmark is shown, with LoRA applied to layers 11 and 12 of the image encoder.

Table 16: **Masking strategy.** The LoRA scale γ is set to 2 for both benchmarks. Performance differences from zero-shot CLIP-ViT-B/16 are shown with a blue (↑) or red (↓) arrow.

Reconstruction	Mask Ratio	Cutoff	Decoder	ImageNet	OOD Average	FG Average
Class token	0.25	0.1		67.65 (↑0.94)	**58.94** (↑1.80)	64.57 (↑0.92)
	0.5	0.1		**67.78** (↑1.07)	58.85 (↑1.71)	**64.72** (↑1.08)
	0.75	0.1		67.48 (↑0.76)	57.87 (↑0.72)	64.35 (↑0.71)
	0.5	0.5		67.52 (↑0.80)	58.30 (↑1.16)	64.49 (↑0.84)
	0.5	1		67.20 (↑0.48)	57.79 (↑0.65)	64.34 (↑0.69)
	0.5	0.1	✓	67.27 (↑0.55)	58.28 (↑1.14)	64.14 (↑0.50)
Visual tokens	0.5	0.1		66.89 (↑0.17)	57.46 (↑0.32)	63.79 (↑0.15)
Image pixel	0.5	0.1	✓	66.67 (↓0.05)	57.05 (↓0.10)	63.50 (↓0.14)



(a) OOD Average

(b) Fine-Grained Average

Figure 8: **Impact of LoRA weight initialization** by data size and comparison with TTT.

Table 17: **t-SNE visualizations.** The plot colors indicate the category classes of each dataset.