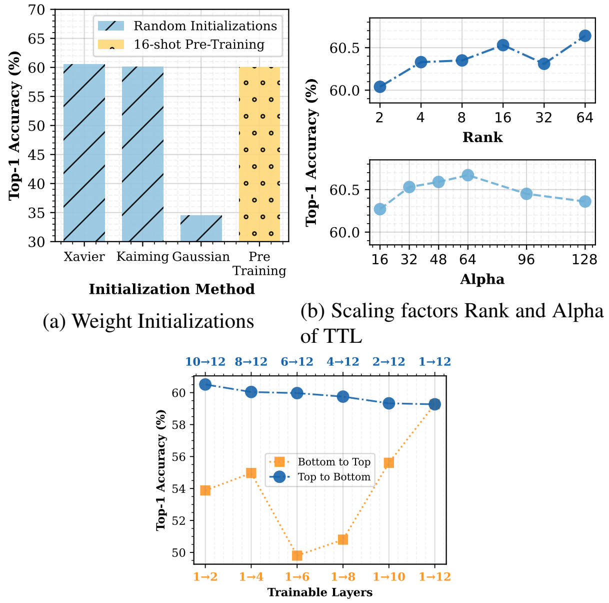
A. Appendix

A.1. Experimental Setup

Hardware and Software details: We implement the CLIP ViT-B/16 backbone architecture for implementing our TTL model. TTL and comparative baseline models are executed on a single NVIDIA A100 40GB GPU, leveraging the PyTorch framework.

Reproducibility: We conducted additional experiments using the settings outlined in Sec. 4.1. To compare baselines across additional model backbones, we specifically implemented CLIP and TPT. Other methods, such as PromptAlign [17], necessitate computation of source data statistics using respective backbones before inference. Since these source data statistics for backbones other than ViT-B/16 are not provided, and recalculating them across more than a million LAION samples would be computationally expensive, we excluded those comparisons.



(a) Weight Initializations

(b) Scaling factors Rank and Alpha of TTL

(c) ViT layers to adapt during Test-time model optimization

Figure 9. Ablating the effects of different TTL components. In (a), Pre-Training refers to 16-shot pre-training of TTL's *LoRA weights* using the ImageNet dataset.

A.2. Effect of TTL Components

Weight Initialization: We assess different methods for *initializing LoRA weights* in TTL, including Xavier [14], Kaiming [18], and Gaussian initializations as shown in Figure 9a. Additionally, we conduct 16-shot pre-training of TTL's adapters using ImageNet dataset following the settings of PromptAlign [17]. Our analysis shows that Xavier, Kaiming, and Pre-trained initialized weights exhibit nearly similar performance, with Xavier showing best performance gain of +0.4 compared to Kaiming. Gaussian initialization results in worse performance. Pre-training TTL's adapters does not enhance generalization as downstream task requires highly task-specific representations, which diverge significantly from the general features learned during pre-training. This suggests that random initialization captures more relevant features for the specific task, rather than relying on broader, less specialized pre-trained knowledge.

Adaptation Layers: In Figure 9c, we investigate the *influence of adapting LoRA adapters in specific layers* of the image encoder on the performance of TTL. Our results reveal that TTL has the most significant impact when adaptation is focused on optimizing attention blocks in the last layers, specifically layers 10 $\rightarrow$ 12. This is evident as the later layers of transformer based models such as CLIP are more discriminative, capturing high-level representations.

Rank and Alpha: We investigate the *impact of change in rank* (r) *and alpha* (α) (Eq. 1 of main paper) on the performance. Our observations reveal that with increasing r, there is a notable enhancement in performance, as depicted in Figure 9b. This phenomenon can be attributed to the availability of more trainable parameters within attention groups at higher ranks. Consequently, this suggests that achieving higher performance gains may necessitate sacrificing parameter efficiency.

A.3. Effect of Entropy Margin

In Eq. 5, entropy margin ε is simply a normalization factor to control the sensitivity of the exponentiation to small changes in entropy. As shown in Figure 10, across different ε values, a negligible change of 0.04 in performance is observed, indicating that TTL's performance is insensitive towards change in ε which signfies that TTL can be used with broad range of ε values.

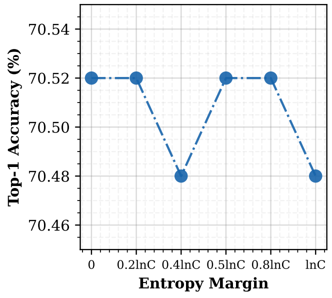


Figure 10. Effect of weighted entropy margin. C=e^x.

A.4. Different Backbone Scales

To verify the scalability of approach, We conduct additional experiments by replacing the ViT-B/16 backbone

Table 4. Analysis on the components of TTL and different methods for test-time optimization of Vision-Language Model. † denotes 16-shot pre-training and * denotes reported accuracy. **Top-1 Accuracy** indicates ImageNet-A accuracy. The arrow ↑ and ↓ indicate **improvements** and **decrements** of our method against the CLIP ($bs.$) method, *i.e.*, CLIP-ViT-B/16.

Method	Tunable Parameter	Entropy-Loss	NO Pre-training	NO Auxiliary-Data	NO External-Model	Top-1 Accuracy
CLIP($bs.$)	–	–	–	–	–	47.14($bs.$)
TPT [41]	Text Prompt	✓	✓	✓	✓	54.59(7.45) ↑
DiffTPT* [12]	Text Prompt	✓	✓	✓	✘	55.68(8.54) ↑
PromptAlign [17]	Multi-Modal Prompt	✓	✓	✘	✓	45.52(1.62) ↓
PromptAlign† [17]	Multi-Modal Prompt	✓	✘	✘	✓	59.03(11.89) ↑
TTL (Ours)	Low-Rank Attentions	✓	✓	✓	✓	**60.51**(13.37) ↑

Table 5. **Top 1 accuracy** % of state-of-the-art baselines under `strict zero-shot settings`, where **ImageNet-Sk.** indicates the ImageNet-Sketch dataset, **OOD Avg.** indicates the OOD average results. $bs.$ indicates the baseline, *i.e.*, CLIP-ViT-B/32.

Method	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-Sk.	Average	OOD Avg.
CLIP-ViT-B/32	59.63($bs.$)	29.57($bs.$)	54.74($bs.$)	66.27($bs.$)	40.77($bs.$)	50.20($bs.$)	47.84($bs.$)
TPT$_{2022}$ [41]	60.93(1.30) ↑	34.61(5.04) ↑	57.15(2.41) ↑	69.60(3.33) ↑	41.62(0.85) ↑	52.78(2.58) ↑	50.75(2.91) ↑
TTL (Ours)	**66.94**(7.31) ↑	**42.43**(12.86) ↑	**58.85**(4.11) ↑	**71.29**(5.02) ↑	**43.54**(2.77) ↑	**56.61**(6.41) ↑	**54.03**(6.19) ↑

with ViT-B/32 variant of CLIP. We use same hyperparameters as in original TTL settings without undergoing any hyperparameter tuning. As shown in Table 5, TTL consistently outperforms CLIP, TPT, and other baselines even with different backbone. In terms of in-domain generalization accuracy, with an accuracy of **56.61**, TTL achieves an average gain of +6.41 and +3.83 compared to CLIP and TPT, respectively. Across out-of-distribution (OOD) shift datasets, TTL demonstrates an average accuracy of **54.03**, with specific gains of +6.19 and +3.28 over CLIP and TPT respectively.

A.5. Qualitative Analysis on Feature Shift

The t-SNE visualizations of various approaches, extracted from visual token embeddings of the last layer of the visual encoder on the ImageNet-A dataset, are presented in Figure 11. It is evident that TPT [41] exhibits a similar arrangement in the t-SNE visualization to that of CLIP [39], as TPT updates prompts solely on the textual side without modifying visual token embeddings. Conversely, PromptAlign [17], which aligns the visual embeddings of test samples to a given source statistic, fails to achieve a optimal class separation boundary, even with pre-trained prompts. In contrast, TTL demonstrates clearly separable boundaries across various hyperparameters like rank r, indicating effective model adaptation for a given test sample.

A.6. Qualitative Analysis on Attention Shift

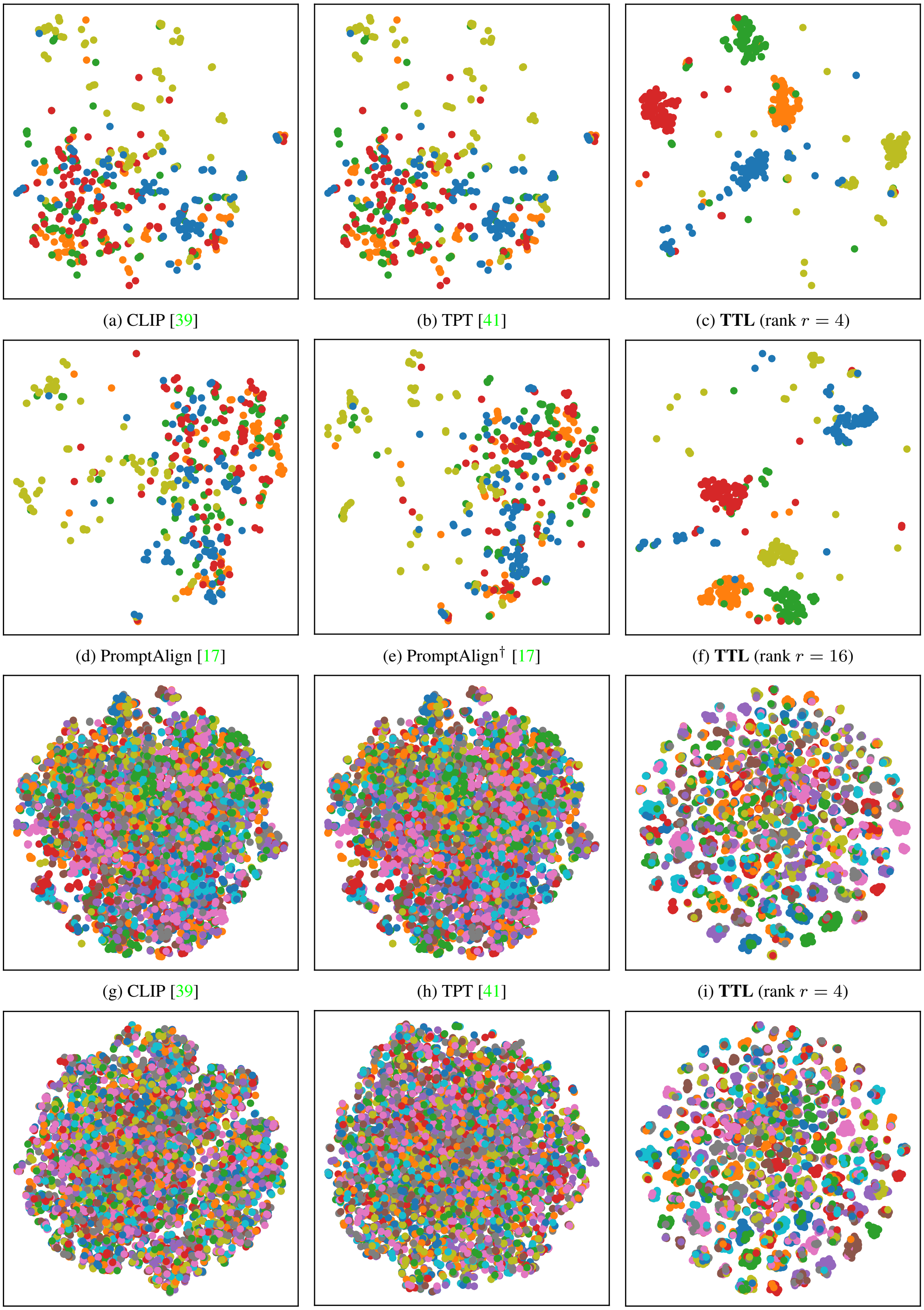
TTL demonstrates enhanced attention towards target discriminative visual areas while simultaneously reducing attention towards background regions compared to CLIP when making correct predictions. However, in cases of incorrect predictions, TTL's attention does not adapt towards the object of interest. This underscores the direct influence of TTL's LoRA parameters in updating the query and value weights of VLM's attention blocks, resulting in improved predictions by focusing the model's attention on object of interest. As shown in Figure 12, TTL amplifies attention towards the object of interest while attenuating attention towards background areas across input samples. By adapting to task-specific data, TTL effectively identifies relevant features in the input sample, thereby improving prediction accuracy.

A.7. Limitations and Future Directions

Limitations: While TTL doesn't necessitate any source data or annotations, our approach does involve a one-step backpropagation process when adapting the low-rank weights during testing. As TTL generates multiple augmented views of a single test sample, it leads to higher memory usage during inference compared to the foundational CLIP model.

Future Research Directions: We present several directions for future works.

- The concept of TTL has the potential to be extended to various downstream tasks like segmentation and detection, thereby enhancing their ability for zero-shot generalization.
- Exploring methods to minimize the memory overhead of TTL and enhance its computational efficiency would be interesting.
- TTL could also be adapted for other domain-specific classification and visual reasoning tasks such as medical imaging and remote sensing applications.
- A promising direction for further research would involve evaluating and devising strategies to enhance the adversarial robustness of vision-language foundational models built upon TTL.



(a) CLIP [39]

(b) TPT [41]

(c) **TTL** (rank $r = 4$)

(d) PromptAlign [17]

(e) PromptAlign† [17]

(f) **TTL** (rank $r = 16$)

(g) CLIP [39]

(h) TPT [41]

(i) **TTL** (rank $r = 4$)

(j) PromptAlign [17]

(k) PromptAlign† [17]

(l) **TTL** (rank $r = 16$)

Figure 11. **t-SNE visualizations** of the final class embedding from the test sets of $\mathcal{C}_1$ dataset: ImageNet-A, following Table 1. TTL could produce linearly separable features for zero-shot generalization than $\mathcal{M}_1$ baselines like CLIP, TPT, and PromptAlign. † indicates 16-shot pre-training. **(a)** to **(f)** represents 5 classes of ImageNet-A with highest number of samples, while **(g)** to **(l)** represents all 200 classes of ImageNet-A.

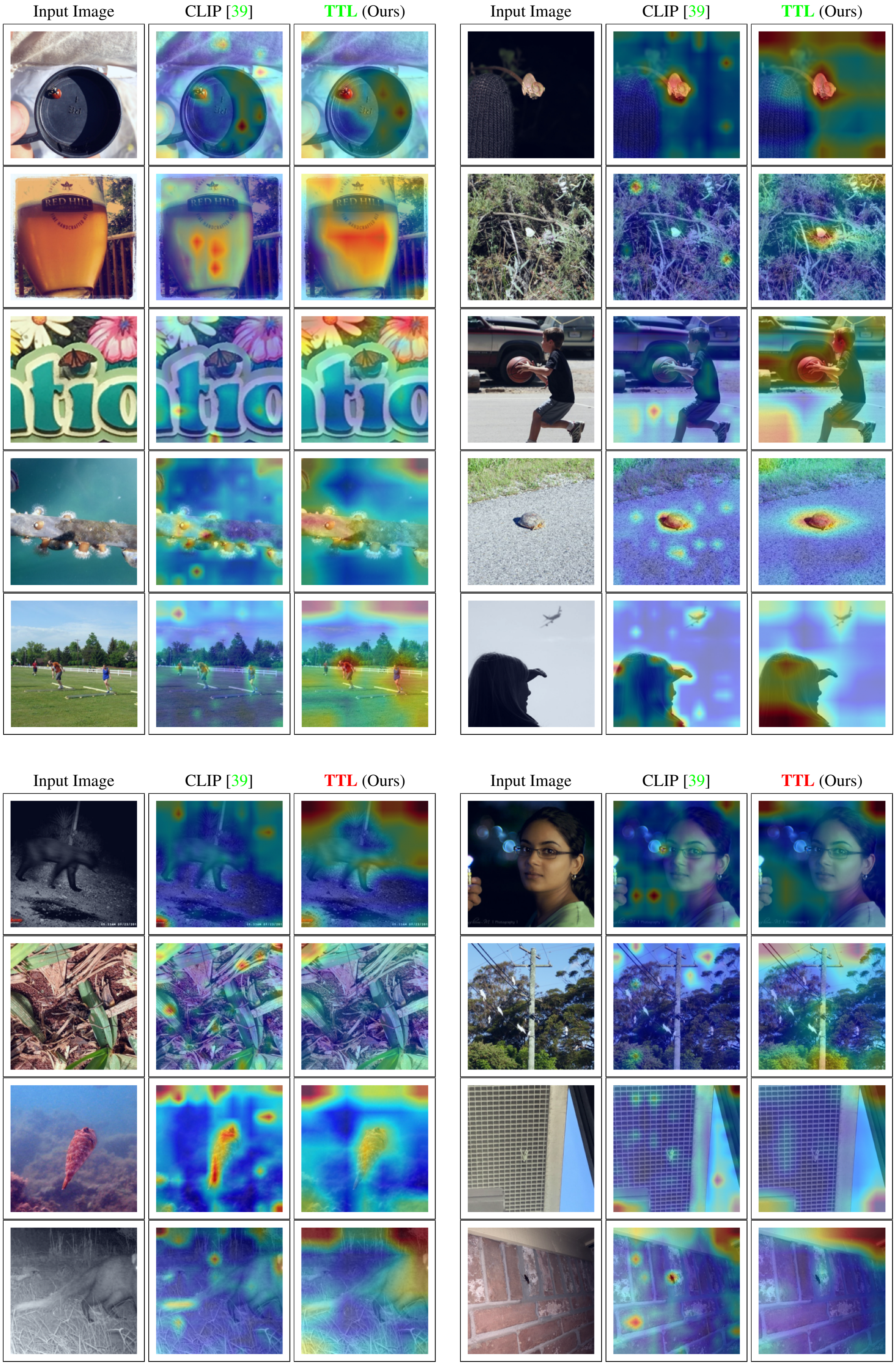


Figure 12. **Attention map visualizations.** TTL updates the attention weights to prioritize features that are more relevant for the target task to better represent domain-specific features, whereas pre-trained CLIP shows inadequacy in capturing such features. Upper row indicates the **correct** prediction, while Lower row indicates **incorrect** prediction.

Table 6. **Weighted Entropy on other baselines.** w = ”with”. PromptAlign† indicates using pre-trained prompts.

Method	**Flower102** [35]	**DTD** [8]	**OxfordPets** [37]	**UCF** [42]	**Caltech101** [11]	**Aircraft** [34]
TPT	69.31	46.23	86.49	66.44	92.49	**24.90**
TPT *w* Wt. Ent.	69.56	46.69	88.58	69.18	93.55	23.14
PromptAlign	51.60	27.60	75.82	57.31	87.18	6.96
PromptAlign *w* Wt. Ent.	52.05	27.66	75.94	58.10	87.61	7.10
PromptAlign†	70.56	45.57	88.96	69.10	92.86	23.70
PromptAlign† *w* Wt. Ent.	71.74	45.04	**89.07**	68.70	93.47	24.15
TTL (Ours)	**70.48**	**46.69**	88.72	**69.20**	**93.63**	23.82

Method	**EuroSAT** [19]	**StanfordCars** [28]	**Food101** [3]	**SUN397** [48]	**Average**
TPT	37.15	66.50	86.93	63.48	63.99
TPT *w* Wt. Ent.	41.96	66.37	84.92	64.96	64.89
PromptAlign	35.57	58.70	82.23	57.84	54.08
PromptAlign *w* Wt. Ent.	37.74	57.99	82.15	57.98	54.43
PromptAlign†	34.91	67.43	86.85	67.73	64.76
PromptAlign† *w* Wt. Ent.	36.56	67.35	**86.91**	**68.03**	65.10
TTL (Ours)	**42.02**	**67.96**	85.05	66.32	**65.39**

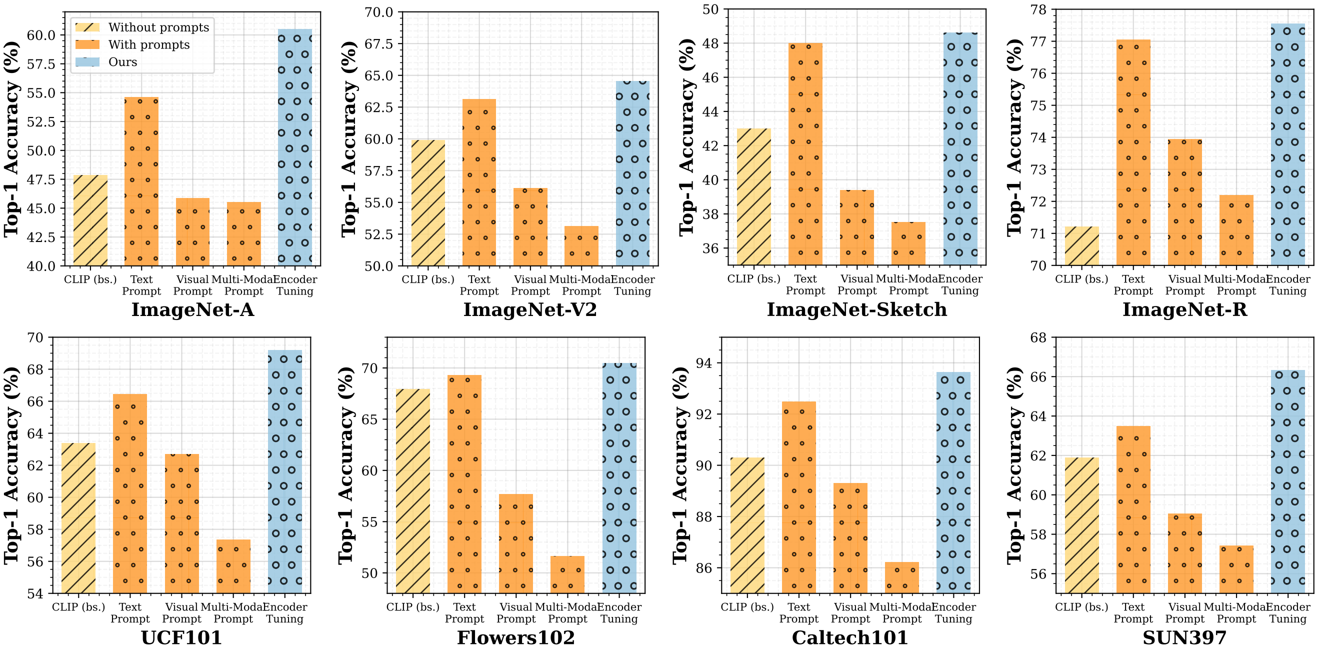


Figure 13. **Test-time performance of zero-shot generalization methods.** CLIP *vs.* Textual Prompt Tuning (TPT) *vs.* Visual Prompt Tuning *vs.* Multi-modal Prompt Tuning *vs.* **TTL (Ours)**. First row denotes the OOD $\mathcal{C}_1$ datasets while Second row denotes 4 Cross-domain $\mathcal{C}_2$ datasets.