

Multi-Cache Enhanced Prototype Learning for Test-Time Generalization of Vision-Language Models

Supplementary Material

In appendix, we provide additional details and experimental results to enhance understanding and insights into our proposed method. This supplementary document is organized as follows:

- **Detailed Dataset Information:** Comprehensive details about the datasets used in our experiments, including their key characteristics and distributions, are provided.
- **Text Templates for Each Dataset:** The text templates used in our experiments for each dataset are listed for reproducibility.
- **Further Discussion on Related Work:** We presented further details of the baselines and highlighted the distinctions between our approach and theirs.
- **Additional Experimental Results:** We provide the accuracy of our method based on the ResNet-50 backbone on natural distribution shifts and cross-dataset generalization, and also provide ablation experimental results of alignment loss, contrast loss, and sensitivity analysis of negative cache size.
- **Theoretical Analysis:** A formal derivation is provided to demonstrate that, compared to using an entropy cache alone, the addition of an align cache results in a lower excess error bound, thereby improving generalization performance.

A. Detailed Dataset Information

In Table A1, we provide comprehensive statistics for each dataset utilized in our experiments, detailing the number of classes, the sizes of the training, validation, and test sets, as well as their associated original tasks. These datasets have emerged as key benchmarks for evaluating the test-time adaptation of vision-language models [9, 17, 33, 36].

B. Text Templates for Each Dataset

In Table B2, we detail the specific hand-crafted prompts utilized for each dataset, following previous work [17]. We also employ CuPL [27] prompts to further enhance performance.

C. Further Discussion on Related Work

We acknowledge that our approach shares some high-level similarities with certain existing methods; however, there are some critical differences. In the following, we discuss

Table A1. Statistics of datasets used in our experiments.

Dataset	Classes	Train	Val	Test	Task
<i>Standard benchmarks</i>					
Caltech101 [8]	100	4 128	1 649	2 465	Object recognition
DTD [5]	47	2 820	1 128	1 692	Texture recognition
EuroSAT [14]	10	13 500	5 400	8 100	Satellite scene classification
FGVCAircraft [22]	100	3 334	3 333	3 333	Aircraft type recognition
Flowers102 [24]	102	4 093	1 633	2 463	Flower species classification
Food101 [3]	101	50 500	20 200	30 300	Food image classification
ImageNet [7]	1 000	1.28 M	—	50 000	Large-scale object recognition
OxfordPets [26]	37	2 944	736	3 669	Pet breed recognition
StanfordCars [18]	196	6 509	1 635	8 041	Car model recognition
SUN397 [37]	397	15 880	3 970	19 850	Scene recognition
UCF101 [35]	101	7 639	1 898	3 783	Action recognition
<i>Robustness benchmarks</i>					
ImageNet-V2 [31]	1 000	—	—	10 000	Collocation robustness
ImageNet-A [15]	200	—	—	7 500	Natural adversarial robustness
ImageNet-R [6]	200	—	—	30 000	Multi-domain robustness
ImageNet-S [13]	1 000	—	—	50 899	Sketch-domain robustness

Table B2. Textual prompts template used in experiments.

Dataset	Prompts
Caltech101 [8]	“a photo of a {CLASS}.”
DTD [5]	“{CLASS} texture.”
EuroSAT [14]	“a centered satellite photo of {CLASS}.”
FGVCAircraft [22]	“a photo of a {CLASS}, a type of aircraft.”
Flowers102 [24]	“a photo of a {CLASS}, a type of flower.”
Food101 [3]	“a photo of {CLASS}, a type of food.”
OxfordPets [26]	“a photo of a {CLASS}, a type of pet.”
StanfordCars [18]	“a photo of a {CLASS}.”
SUN397 [37]	“a photo of a {CLASS}.”
UCF101 [35]	“a photo of a person doing {CLASS}.”
ImageNet [7]	“itap of a {CLASS}.”
ImageNet-S [13]	“a origami {CLASS}.”
ImageNet-R [6]	“a bad photo of the {CLASS}.”
ImageNet-V2 [31]	“a photo of the small {CLASS}.”
ImageNet-A [15]	“a photo of the large {CLASS}.”
	“a {CLASS} in a video game.”
	“art of the {CLASS}.”

the distinctions between our method TDA [17], DPE [39] and PromptAlign [1] individually, and further compare various approaches from several perspectives as summarized in C3. Moreover, following these comparisons, we devote an entire chapter to Prototype Learning to further discuss prototype-based research.

C.1. TDA [17]

While both DMN [45] and TDA [17] enhance test-time generalization by storing historical test samples in a cache or memory module, they share some similarities in this regard. However, these methods rely solely on entropy to assess sample quality, which limits the potential benefits of caching. In contrast, our MCP method fully leverages category distribution information by introducing an alignment

Table C3. Comparison of different methods. **Multimodal**: whether multimodal learning is employed; **HistInfo**: whether historical information is utilized; **NoEncGrad**: no access to encoder gradients; **NoExtraData**: no requirement for extra data or models; **HighEntro**: leverages information from high-entropy samples ; **ExpTestDist**: explicitly considers the distribution of test samples.

Method	Multimodal	HistInfo	NoEncGrad	NoExtraData	HighEntro	ExpTestDist
TPT	✗	✗	✗	✓	✗	✗
DiffTPT	✗	✗	✗	✗	✗	✗
PromptAlign	✓	✗	✗	✗	✗	✓
TDA	✗	✓	✓	✓	✓	✗
DMN	✗	✓	✓	✓	✗	✗
BoostAdapter	✗	✓	✓	✓	✗	✗
DPE	✓	✓	✓	✓	✗	✗
MCP	✗	✓	✓	✓	✓	✓
MCP++	✓	✓	✓	✓	✓	✓

cache to promote intra-class compactness, thereby further enhancing the performance gains from caching. Moreover, MCP++ incorporates learnable residual parameters to refine prototype representations, enabling the model to dynamically adapt based on test samples.

C.2. DPE [39]

Similarly, DPE [39] relies solely on entropy as the criterion for caching samples and does not utilize high-entropy samples within the test stream. Its performance gains primarily come from evolving visual and textual prototypes to enhance the model’s generalization capability during testing. In contrast, our MCP method requires no additional training and achieves superior performance and higher efficiency than DPE by fully exploiting the complementary multi-cache mechanism.

Similar to DPE, MCP++ augments both visual and textual prototypes with learnable residual vectors, and performs multimodal fine-tuning. However, the three-way cache (Entropy, Align, Negative) supplies a diverse, complementary test sample set, and the **Prototype Center** constraint pushes the visual prototypes to high quality before any multimodal residual tuning. As a result, turning off residual tuning (i.e., MCP++ \rightarrow MCP) causes only a minor drop, whereas the same ablation on DPE leads to a much larger degradation showing that that our initial prototypes are intrinsically stronger. Moreover, DPE leverages only low-entropy samples and lacks a negative cache to constrain and calibrate its visual prototypes. Furthermore, our method does not adopt the textual prototype evolution mechanism used in DPE.

C.3. PromptAlign [1]

PromptAlign [1] is the only existing method that explicitly considers the distribution of test samples. However, it still differs significantly from our MCP approach. Specifically, PromptAlign relies on an ImageNet subset as the source domain and computes the mean and variance of the

token representations produced by the CLIP encoder on this subset. At test time, it constrains the prompt-tuning process by aligning the token statistics of test samples to the source-domain statistics, thereby reducing the domain shift between source and test domains. In contrast, MCP requires no ImageNet subset for initialization; instead, it dynamically constructs the Prototype Center at test time by combining textual and visual information and uses this center solely for sample selection rather than as part of the optimization loss. Moreover, MCP does not access internal token-level statistics (such as mean and variance) of the encoder, but only retrieves the final [CLS] embedding and applies a lightweight residual vector for fine-tuning.

C.4. Prototype Learning

Prototype learning was formally introduced to the few-shot classification paradigm by Snell *et al.* [34] and has since rapidly expanded into a wide array of research domains: semantic segmentation [21, 32], object detection [16], OOD detection [19], continual learning [10, 11, 30], action recognition [23], domain adaption [2, 12, 20, 25, 29, 36, 38, 39, 43, 44, 47]. Similar to previous work [12, 36, 38, 39], we construct text prototypes by averaging the embeddings of multiple prompts and derive visual prototypes by averaging the cached image features. Because plain averaging often yields prototypes that are neither discriminative nor robust, much recent research has focused on refining or calibrating prototype quality. Representative approaches include: DPE [39], which co-evolves visual and textual prototypes while learning a multimodal residual online for precise alignment; ClusterAdapter [12], which clusters prototypes and performs fine-grained tuning inside a lightweight adapter; TPS [36], which combines richer textual descriptions with online residual learning to correct text prototypes; BPRE [28], which introduces a multidimensional quality-aware reward and a prototype–reward interactive evolution mechanism; and ProtoMM [47] aligns visual and textual particles via optimal transport and dynamically re-weights them to

Table C4. Results on the Cross-Domain Benchmark. Top-1 accuracy (%) results are presented for all evaluated methods employing the ResNet-50 visual backbone of CLIP. The best results are highlighted in **bold**.

Method	Aircraft	Caltech	Cars	DTD	EuroSAT	Flower	Food101	Pets	SUN397	UCF101	Average
CLIP-ResNet50	15.66	85.88	55.70	40.37	23.69	61.75	73.97	82.57	58.80	58.84	55.82
CoOp	15.12	86.53	55.32	37.29	26.20	61.55	75.59	87.00	58.15	59.05	56.18
TPT	17.58	87.02	58.46	40.84	28.33	62.69	74.88	84.49	61.46	60.82	57.66
DiffTPT	17.60	86.89	60.71	40.72	41.04	63.53	79.21	83.40	62.72	62.67	59.85
TDA	17.61	89.70	57.78	43.74	42.11	68.74	77.75	86.18	62.53	64.18	61.03
DMN	22.77	90.14	60.02	50.41	48.72	67.93	76.70	86.78	64.39	65.34	63.71
DPE	19.80	90.83	59.26	50.18	41.67	67.60	77.83	85.97	64.23	61.98	61.93
MCP	23.04	90.99	61.27	53.19	55.60	68.49	78.37	87.35	65.39	67.14	65.08
MCP++	23.40	91.13	61.76	53.61	55.74	69.96	78.44	87.49	65.55	67.86	65.49

Table C5. Results on the OOD Benchmark. Top-1 accuracy (%) results are presented for all evaluated methods employing the ViT-B/16 visual backbone of ResNet-50. The best results are highlighted in **bold**.

Method	ImageNet	A	V2	R	S	OOD Avg.	Average
CLIP-ResNet50	58.16	21.83	51.41	56.15	33.37	40.69	44.18
CoOp	63.33	23.06	55.40	56.60	34.67	42.43	46.61
TPT	60.74	26.67	54.70	59.11	35.09	43.89	47.26
DiffTPT	60.80	31.06	55.80	58.80	37.10	45.69	48.71
TDA	61.35	30.29	55.54	62.58	38.12	46.63	49.58
DMN	63.87	28.57	56.12	61.44	39.84	46.49	49.97
DPE	63.41	30.15	56.72	63.72	40.03	47.66	50.81
MCP	64.19	29.30	56.93	64.40	41.16	48.02	51.20
MCP++	64.44	29.29	57.12	64.92	41.35	48.17	51.42

refine multimodal prototypes. Our method MCP boosts visual prototype quality **without extra training** by screening test samples on two criteria, entropy and distance to the prototype center. The extended variant, MCP++, performs prototype residual tuning at test time, incorporating cross-modal alignment and negative-sample distance constraints to further improve prototype accuracy and robustness.

D. Additional Experimental Results

D.1. Cross-Dataset Generalization

Table C4 presents the results of our cross-dataset generalization experiments on the ResNet-50 backbone, comparing our method against state-of-the-art approaches across ten diverse datasets. The results demonstrate that our proposed methods, MCP and MCP++, achieve superior performance under significant distributional shifts. On the ResNet-50 backbone, our methods achieve the best performance on 9 out of 10 tasks, with MCP++ obtaining the highest average accuracy of 65.49%. Compared to both prompt-based methods (e.g., CoOp [46], TPT [33], and DiffTPT [4]) and cache-based methods (e.g., TDA [17], DPE [39], DMN [45]), MCP and MCP++ achieve superior results.

D.2. Natural Distribution Shifts

We evaluate the robustness of our proposed methods on in-domain ImageNet and its four out-of-distribution variants using the ResNet-50 backbone, with the results presented in Table C5. Our approach outperforms leading prompt-tuning methods, achieving a 4.16% improvement over TPT with MCP++ and a 2.71% improvement over DiffTPT. When compared to cache-based methods, MCP++ surpasses TDA by 1.84%, DMN by 1.45%, and DPE by 0.61%. This result indicates that our method is generally effective in both domain-specific variations and out-of-distribution robustness scenarios, demonstrating its ability to maintain high performance under natural distribution shifts.

D.3. Ablation Study Results for Losses

As shown in Table D6, introducing either the alignment or contrastive loss improves model performance compared to the baseline. Specifically, applying only the alignment loss yields 30.07% accuracy, while the contrastive loss alone achieves 29.92%. Notably, combining both losses leads to the best result, with 30.18% accuracy on the Aircraft dataset.

Table D6. Ablation Study Results for Align and Contrast Losses

L_{align}	L_{contrast}	Aircraft
		29.74
✓		30.07
	✓	29.92
✓	✓	30.18

D.4. More Sensitivity Analyses of Cache Size

In Fig. 4 (Middle), we have analyzed the sensitivity of model performance to the sizes of the Align Cache and Entropy Cache on cross-domain datasets. We further investigate the effect of the Negative Cache size on the model’s adaptation capability.

As shown in Table D7, the model achieves optimal performance when the negative cache size is set to a moderate value (e.g., 3), while both smaller and larger cache sizes lead to performance degradation. This observation suggests that a moderate number of high-entropy negative samples helps mitigate the impact of noisy pseudo-labels and suppress misclassification near category boundaries. However, an excessive number of negative samples may introduce additional noise, impairing the model’s discriminative ability.

Overall, model performance exhibits a certain degree of sensitivity to cache size. Larger cache capacity does not necessarily yield better results; instead, a balance must be struck between “information coverage” and “representation purity.” In addition, we observe that the optimal cache configuration may vary across datasets. To ensure consistency and fairness throughout all experiments, we adopt a unified configuration of $|M_c^{\text{entropy}}| = 10$, $|M_c^{\text{align}}| = 10$, and $|M_c^{\text{negative}}| = 3$.

Table D7. Effects of different negative cache sizes on cross-domain datasets under ResNet-50.

Negative Cache Size	2	3	5
Accuracy (%)	64.97	65.08	64.73

E. Theoretical Analysis

In this section, we demonstrate that incorporating Align Cache further reduces the error upper bound compared to using only Entropy Cache. First, we present the fundamental theoretical setup for our analysis; then, we introduce several key assumptions. Finally, we derive the error upper bounds for both approaches and prove that the method employing Align Cache achieves a tighter bound. We adopt the proof strategy of BoostAdapter [41]. Unlike their setting, which automatically satisfies the Strong Density Condition in Assumption E.4 (i.e., $c_a > c_t$) due to its construction, our setting does not guarantee this inequality *a priori*.

Therefore, we explicitly prove that $c_a > c_t$ holds within our framework.

E.1. Problem Setting.

We formalize the foundational setup for test-time adaptation. Consider a binary classification task (extensible to multi-class scenarios) with joint data distribution $p_t(x, y)$ over the target domain. Assume we observe n i.i.d. test samples:

$$\{(x_i, y_i)\}_{i=1}^n \stackrel{\text{i.i.d.}}{\sim} p_t(x, y)$$

where $y_i \in \{0, 1\}$ represents one-hot encoded labels in the binary case (and, in the multi-class case, y_i is an one-hot vector of length N).

E.2. Definitions

Classification Error. For a binary classifier $f : \mathcal{X} \rightarrow \{0, 1\}$, its error under distribution $p_t(x, y)$ is defined as:

$$\epsilon(f) = \mathbb{E}_{p_t(x, y)}[\mathbf{1}_{\{f(x) \neq y\}}] = \mathbb{E}_{p_t(x, y)}[|f(x) - y|] \quad (1)$$

The equality holds specifically for binary classification.

Excess Error. Let f^* denote the Bayes-optimal classifier defined as

$$f^*(x) = \mathbb{I}_{\{\eta(x) \geq 1/2\}}$$

where $\eta(x) = \mathbb{E}[y|x]$. The excess error of f is:

$$\mathcal{E}(f) = \epsilon(f) - \epsilon(f^*) = 2\mathbb{E}_{x \sim p_t(x)} \left[\left| \eta(x) - \frac{1}{2} \right| \cdot \mathbb{I}_{\{f(x) \neq f^*(x)\}} \right] \quad (2)$$

Cache Classifier. Given an encoder $g : \mathcal{X} \rightarrow \mathbb{R}^d$ and a cache of K stored samples, Tip-Adapter [40] proposes

$$p_{\text{cache}}(x) = A(g(x) G_{\text{cache}}^\top) Y, \quad (3)$$

where $A(z) = \alpha \exp(-\beta(1 - z))$ is a scaling function, $G_{\text{cache}} \in \mathbb{R}^{K \times d}$ contains features of cached samples, and $Y \in \mathbb{R}^{K \times N}$ holds their labels.

Aligned Distribution $p_a(x)$ In addition to the original target distribution $p_t(x, y)$, we define an aligned distribution $p_a(x)$ that focuses on test samples near a category anchor $\mu_c \in \mathbb{R}^d$. Specifically, we retain only those x satisfying

$$\|g(x) - \mu_c\| \leq d_0,$$

where $d_0 > 0$ is a constraint threshold. Formally,

$$p_a(x) = p_t(x \mid \|g(x) - \mu_c\| \leq d_0).$$

Hence, $p_a(x)$ captures the center-constrained subset of the target domain.

Aligned Cache Classifier. While existing cache-based methods generally keep only historical test samples, we further incorporate *aligned samples* near μ_c . Suppose k_t historical samples and k_a aligned samples reside in the cache. Building on Eq. 3, we define:

$$p_{\text{align}}(x) = A(g(x) \tilde{G}_{\text{cache}}^\top) \tilde{Y}, \quad (4)$$

where $\tilde{G}_{\text{cache}} \in \mathbb{R}^{(k_t+k_a) \times d}$ mixes historical and aligned features, and $\tilde{Y} \in \mathbb{R}^{(k_t+k_a) \times N}$ is the corresponding label matrix.

E.3. Practical Implementation.

In practice, we adopt an entropy-based threshold to pick reliable historical samples, and require $\|g(x) - \mu_c\| \leq d_0$ for the aligned ones. The distance d_0 is computed dynamically from the cache statistics, defined as the distance between the class anchor μ_c and the farthest retained sample in the cache. Each class is capped at k slots, replacing higher-entropy or out-of-reach samples. We omit further details here, as our main focus is on the theoretical guarantees.

E.4. Assumptions

Strong Density Condition. Let x_0 be any test sample from $p_t(x)$ and also from the aligned distribution $p_a(x_0)$. We assume there exist positive constants m and M , scaling factors c_t and c_a , and a radius $R > 0$. Define $\mathcal{B}(x, r) = \{x' \mid \|x' - x\| \leq r\}$. We assume $p_t(x)$ and $p_a(x_0)$ are absolutely continuous w.r.t. the Lebesgue measure λ in \mathbb{R}^d . Then, for any $r \in (0, R]$,

$$\begin{cases} \lambda[p_t(x) \cap \mathcal{B}(x_0, r)] \geq c_t \lambda[\mathcal{B}(x_0, r)], \\ \lambda[p_a(x_0) \cap \mathcal{B}(x_0, r)] \geq c_a \lambda[\mathcal{B}(x_0, r)], \\ m < \frac{dp_t(x)}{d\lambda} < M, \quad m < \frac{dp_a(x)}{d\lambda} < M. \end{cases}$$

Intuitively, in any local region, the target distribution occupies at least c_t fraction of that area's mass, while the aligned distribution p_a has an area fraction c_a that is strictly greater than c_t .

Proof of $c_a > c_t$. Recall that we defined the aligned distribution as

$$p_a(x) = p_t(x \mid \|g(x) - \mu_c\| \leq d_0),$$

which, by the definition of conditional probability, can be written as

$$p_a(x) = \frac{p_t(x)}{P(\|g(x) - \mu_c\| \leq d_0)}, \quad \text{for } \|g(x) - \mu_c\| \leq d_0.$$

Now, consider any test sample x_0 and any ball $\mathcal{B}(x_0, r)$ (with $r \in (0, R]$) that is fully contained in the region

$$\{x : \|g(x) - \mu_c\| \leq d_0\}.$$

For the aligned distribution, the probability mass within $\mathcal{B}(x_0, r)$ is given by

$$\begin{aligned} \lambda[p_a(x) \cap \mathcal{B}(x_0, r)] &= \int_{\mathcal{B}(x_0, r)} p_a(x) dx \\ &= \frac{1}{P(\|g(x) - \mu_c\| \leq d_0)} \int_{\mathcal{B}(x_0, r)} p_t(x) dx. \end{aligned} \quad (5)$$

By the Strong Density Condition for the target distribution, we have

$$\int_{\mathcal{B}(x_0, r)} p_t(x) dx \geq c_t \lambda[\mathcal{B}(x_0, r)].$$

Thus,

$$\lambda[p_a(x) \cap \mathcal{B}(x_0, r)] \geq \frac{c_t}{P(\|g(x) - \mu_c\| \leq d_0)} \lambda[\mathcal{B}(x_0, r)].$$

By definition of the local density constant for the aligned distribution, we require that

$$\lambda[p_a(x) \cap \mathcal{B}(x_0, r)] \geq c_a \lambda[\mathcal{B}(x_0, r)].$$

Comparing the two inequalities, it follows that

$$c_a \geq \frac{c_t}{P(\|g(x) - \mu_c\| \leq d_0)}.$$

Since the filtering condition ensures that not all samples satisfy $\|g(x) - \mu_c\| \leq d_0$, we have

$$P(\|g(x) - \mu_c\| \leq d_0) < 1.$$

Therefore, it must hold that

$$c_a > c_t.$$

L-Lipschitz Condition. We assume there is a positive constant L such that the classification function f is L -Lipschitz:

$$|f(x) - f(x')| \leq L \|x - x'\|.$$

Intuitively, f is smooth, so small changes in input cannot produce large output variations.

Low Noise Condition. Let β and C_β be positive constants. We assume that for any $t > 0$,

$$P_{x \sim p_t(x)}\left(|f(x) - \frac{1}{2}| < t\right) \leq C_\beta t^\beta.$$

Intuitively, the probability mass near the threshold ($f(x) \approx \frac{1}{2}$) is small, ensuring high confidence in that region.

E.5. Proof Sketch

Under these three assumptions, we show how the cache classifier can achieve low empirical risk by retrieving sufficiently representative samples. The key arguments rely on:

- **Strong Density:** guarantees local coverage of the target domain and (separately) of the aligned distribution near μ_c . In particular, $c_a > c_t$ implies the aligned region is “denser” when restricted to center-based samples.
- **Lipschitz Smoothness:** ensures f does not fluctuate excessively around similar points.
- **Low Noise:** limits the mass near the decision boundary, avoiding high uncertainty.

We leverage these properties to analyze two scenarios: one that uses only an entropy cache, and another that combines it with an align cache.

E.6. Propositions

Based on the above assumptions, we propose the following two propositions. These results were originally derived in AdaNPC[42] and further extended in Boostadapter[41]. For more detailed derivations and complete proofs, please refer to the appendices of both AdaNPC[42] and Boostadapter[41].

Proposition 1 (Entropy Cache Reduces Empirical Risk).

Consider a training-free classifier f that uses only historical low-entropy samples, as defined in Eq. 3. Let n_t be the number of confidently predicted low-entropy samples from the target domain, and k_t the subset of those stored in the cache. Under Assumptions 1–3, and for sufficiently large n_t and k_t , it holds with high probability that

$$\mathcal{E}(f) \leq \mathcal{O}\left(\left(\frac{1}{k_t}\right)^{\frac{1}{4}} + \left(\frac{k_t}{c_t n_t}\right)^{\frac{1}{d}}\right)^{1+\beta}. \quad (6)$$

This guarantees that by selecting k_t high-quality historical samples out of n_t , one can achieve low empirical risk if n_t is large enough, since the cache captures representative information from the target domain.

Proposition 2 (Entropy Cache Benefits from Align Cache).

Next, let n_t again be the count of low-entropy historical samples, while n_a is the number of aligned samples obtained via distance-to-center filtering. Suppose the cache stores k_t low-entropy samples and k_a aligned samples. Let w_{ti} and w_{ai} denote the instance weights assigned to low-entropy and aligned samples, respectively, with the explicit normalization constraint $\sum_{i=1}^{k_t} w_{ti} + \sum_{i=1}^{k_a} w_{ai} = 1$. Then, the empirical risk of f (as in Eq. 4) is bounded by

$$\begin{aligned} \mathcal{E}(f) \leq & \mathcal{O}\left(\left(\frac{1}{k_t+k_a}\right)^{\frac{1}{4}} + \sum_{i=1}^{k_t} w_{ti} \left(\frac{k_t}{c_t n_t}\right)^{\frac{1}{d}}\right. \\ & \left. + \sum_{i=1}^{k_a} w_{ai} \left(\frac{k_a}{c_a n_a}\right)^{\frac{1}{d}}\right)^{1+\beta}. \end{aligned} \quad (7)$$

Hence, the historical cache can reduce empirical risk even further by incorporating k_a aligned samples, thereby capturing both general historical information and center-focused aligned information about the target domain.

E.7. Proof of Proposition 2 Improvement

We now show that, incorporating aligned samples leads to a strictly tighter error bound than using only historical samples. For a more rigorous proof, please refer to the appendices of AdaNPC [42] and BoostAdapter [41].

Step 1: Retention Ratio Consistency

Let the filtering rule select samples satisfying event A (Here, entropy is less than $\leq \tau$), which is *statistically independent* of the alignment condition $\mathcal{R} = \{x : \|g(x) - \mu_c\| \leq d_0\}$. Then:

- For historical samples from $p_t(x)$:

$$\frac{k_t}{n_t} = p_t(A)$$

- For aligned samples from $p_a(x) = p_t(x | \mathcal{R})$:

$$\frac{k_a}{n_a} = p_t(A | \mathcal{R}) \stackrel{(\text{indep.})}{=} p_t(A)$$

This implies:

$$\frac{k_t}{n_t} = \frac{k_a}{n_a} \quad (8)$$

Step 2: Error Bound Reformulation

Since we have normalized weights w_{ti}, w_{ai} as:

$$\sum_{i=1}^{k_t} w_{ti} + \sum_{i=1}^{k_a} w_{ai} = 1$$

Let $a = \sum_{i=1}^{k_t} w_{ti}$, $1-a = \sum_{i=1}^{k_a} w_{ai}$. The bounds become:

- Proposition 1 (Historical only):

$$\mathcal{E}(f) \leq \mathcal{O}\left(\underbrace{\left(\frac{1}{k_t}\right)^{\frac{1}{4}}}_{\text{Term 1}} + \underbrace{\left(\frac{k_t}{c_t n_t}\right)^{\frac{1}{d}}}_{\text{Term 2}}\right)^{1+\beta}$$

- Proposition 2 (Historical + Aligned):

$$\begin{aligned} \mathcal{E}(f) \leq & \mathcal{O}\left(\underbrace{\left(\frac{1}{k_t+k_a}\right)^{\frac{1}{4}}}_{\text{Term' 1}}\right. \\ & \left. + \underbrace{\left(a \left(\frac{k_t}{c_t n_t}\right)^{\frac{1}{d}} + (1-a) \left(\frac{k_a}{c_a n_a}\right)^{\frac{1}{d}}\right)}_{\text{Term' 2}}\right)^{1+\beta} \end{aligned}$$

Step 3: Strict Dominance of Proposition 2

- Sample Size Advantage

$$\underbrace{\left(\frac{1}{k_t + k_a}\right)^{\frac{1}{4}}}_{\text{Term' 1}} < \underbrace{\left(\frac{1}{k_t}\right)^{\frac{1}{4}}}_{\text{Term 1}} \quad (\because k_t + k_a > k_t).$$

- Convex Combination Advantage From

$$\frac{k_t}{n_t} = \frac{k_a}{n_a} \quad \text{and} \quad c_a > c_t :$$

$$\frac{k_a}{c_a n_a} = \frac{k_t}{c_t n_t} < \frac{k_t}{c_t n_t} \implies \left(\frac{k_a}{c_a n_a}\right)^{\frac{1}{d}} < \left(\frac{k_t}{c_t n_t}\right)^{\frac{1}{d}}.$$

Thus,

$$a \underbrace{\left(\frac{k_t}{c_t n_t}\right)^{\frac{1}{d}}}_{\text{Term' 2}} + (1-a) \underbrace{\left(\frac{k_a}{c_a n_a}\right)^{\frac{1}{d}}}_{\text{Term 2}} < \underbrace{\left(\frac{k_t}{c_t n_t}\right)^{\frac{1}{d}}}_{\text{Term 2}}.$$

- Combined Effect

$$\underbrace{\text{Term' 1} + \text{Term' 2}}_{\text{Proposition 2}} < \underbrace{\text{Term 1} + \text{Term 2}}_{\text{Proposition 1}}.$$

E.8. Concluding Discussion

Overall, we have theoretically explored the problem of test-time adaptation by developing a cache-based classifier model and defining the relevant concepts. Under the assumptions of strong density, Lipschitz smoothness, and low noise, we analyzed the error bounds for both using only historical samples and combining them with aligned samples. The theoretical proofs demonstrate that the overall error bound is significantly smaller when aligned samples are incorporated, providing solid theoretical support for the introduction of the Align Cache method.

References

- [1] Jameel Abdul Samadh, Mohammad Hanan Gani, Noor Hussein, Muhammad Uzair Khattak, Muhammad Muzammal Naseer, Fahad Shahbaz Khan, and Salman H Khan. Align your prompts: Test-time prompting with distribution alignment for zero-shot generalization. *Advances in Neural Information Processing Systems*, 36:80396–80413, 2023. 1, 2
- [2] Eman Ali, Sathira Silva, and Muhammad Haris Khan. Dpa: Dual prototypes alignment for unsupervised adaptation of vision-language models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6083–6093. IEEE, 2025. 2
- [3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101—mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461. Springer, 2014. 1
- [4] Yong Liu Salman Khan Chun-Mei Feng, Kai Yu and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, page 2704–2714, 2023. 3
- [5] Mircea Cimpoi, Subhansu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2013. 1
- [6] Norman Mu Saurav Kadavath Frank Wang Evan Dorundo Rahul Desai Tyler Zhu Samyak Parajuli Mike Guo Dawn Song Jacob Steinhardt Dan Hendrycks, Steven Basart and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 1
- [8] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178. IEEE, 2004. 1
- [9] Chun-Mei Feng, Kai Yu, Yong Liu, Salman A. Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2704–2714, 2023. 1
- [10] Takuma Fukuda, Hiroshi Kera, and Kazuhiko Kawamoto. Adapter merging with centroid prototype mapping for scalable class-incremental learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4884–4893, 2025. 2
- [11] Jiayang Gou, Luping Ji, Pei Liu, and Mao Ye. Queryable prototype multiple instance learning with vision-language models for incremental whole slide image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3158–3166, 2025. 2
- [12] Meng-Hao Guo, Yi Zhang, Tai-Jiang Mu, Sharon X. Huang, and Shi-Min Hu. Tuning vision-language models with multiple prototypes clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11186–11199, 2024. 2
- [13] Zachary Lipton Haohan Wang, Songwei Ge and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019. 1
- [14] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 1
- [15] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In

- Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 1
- [16] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8419–8428, 2018. 2
- [17] Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El-Saddik, and Eric P. Xing. Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 14162–14171, 2024. 1, 3
- [18] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. 1
- [19] Shawn Li, Huixian Gong, Hao Dong, Tiankai Yang, Zhengzhong Tu, and Yue Zhao. Dpu: Dynamic prototype updating for multimodal out-of-distribution detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10193–10202, 2025. 2
- [20] Yiwen Liang, Hui Chen, Yizhe Xiong, Zihan Zhou, Mengyao Lyu, Zijia Lin, Shuaicheng Niu, Sicheng Zhao, Jungong Han, and Guiguang Ding. Advancing reliable test-time adaptation of vision-language models under visual variations. *arXiv preprint arXiv:2507.09500*, 2025. 2
- [21] Jie Liu, Jiayi Shen, Pan Zhou, Jan-Jakob Sonke, and Efstathios Gavves. Probabilistic prototype calibration of vision-language models for generalized few-shot semantic segmentation. *arXiv preprint arXiv:2506.22979*, 2025. 2
- [22] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1
- [23] Xinzhe Ni, Yong Liu, Hao Wen, Yatai Ji, Jing Xiao, and Yujiu Yang. Multimodal prototype-enhanced network for few-shot action recognition. *arXiv preprint arXiv:2212.04873*, 2022. 2
- [24] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pages 722–729. IEEE, 2008. 1
- [25] Kamalesh Palanisamy, Yu-Wei Chao, Xinya Du, Yu Xiang, et al. Proto-clip: Vision-language prototypical network for few-shot learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2594–2601. IEEE, 2024. 2
- [26] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3498–3505. IEEE, 2012. 1
- [27] Sarah Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15645–15655, 2022. 1
- [28] Xiaozhen Qiao, Peng Huang, Jiakang Yuan, Xianda Guo, Bowen Ye, Zhe Sun, and Xuelong Li. Bidirectional prototype-reward co-evolution for test-time adaptation of vision-language models. *arXiv preprint arXiv:2503.09394*, 2025. 2
- [29] Hongyu Qu, Jianan Wei, Xiangbo Shu, and Wenguan Wang. Learning clustering-based prototypes for compositional zero-shot learning. *arXiv preprint arXiv:2502.06501*, 2025. 2
- [30] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2
- [31] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019. 1
- [32] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 2
- [33] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *arXiv preprint arXiv:2209.07511*, 2022. 1, 3
- [34] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 2
- [35] K Soomro. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1
- [36] Elaine Sui, Xiaohan Wang, and Serena Yeung-Levy. Just shift it: Test-time prototype shifting for zero-shot generalization with vision-language models. *arXiv preprint arXiv:2403.12952*, 2024. 1, 2
- [37] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010. 1
- [38] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10899–10909, 2022. 2
- [39] Ce Zhang, Simon Stepputtis, Katia P. Sycara, and Yaqi Xie. Dual prototype evolving for test-time generalization of vision-language models. *arXiv preprint arXiv:2410.12790*, 2024. 1, 2, 3
- [40] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kun-chang Li, Jifeng Dai, Yu Jiao Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 4
- [41] Taolin Zhang, Jinpeng Wang, Hang Guo, Tao Dai, Bin Chen, and Shu-Tao Xia. Boostadapter: Improving vision-language test-time adaptation via regional bootstrapping. *arXiv preprint arXiv:2410.15430*, 2024. 4, 6

- [42] Yifan Zhang, Xue Wang, Kexin Jin, Kun Yuan, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Adanpc: Exploring non-parametric classifier for test-time adaptation. In *International Conference on Machine Learning*, pages 41647–41676. PMLR, 2023. [6](#)
- [43] Yi Zhang, Ce Zhang, Xueting Hu, and Zhihai He. Unsupervised prototype adapter for vision-language models. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 197–209. Springer, 2023. [2](#)
- [44] Yi Zhang, Chun-Wun Cheng, Ke Yu, Zhihai He, Carola-Bibiane Schönlieb, and Angelica I Aviles-Rivero. Node-adapter: Neural ordinary differential equations for better vision-language reasoning. *arXiv preprint arXiv:2407.08672*, 2024. [2](#)
- [45] Yabin Zhang, Wen-Qing Zhu, Hui Tang, Zhiyuan Ma, Kaiyang Zhou, and Lei Zhang. Dual memory networks: A versatile adaptation approach for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28718–28728, 2024. [1](#), [3](#)
- [46] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130:2337 – 2348, 2021. [3](#)
- [47] Xingyu Zhu, Shuo Wang, Beier Zhu, Miaoge Li, Yunfan Li, Junfeng Fang, Zhicai Wang, Dongsheng Wang, and Hanwang Zhang. Dynamic multimodal prototype learning in vision-language models. *arXiv preprint arXiv:2507.03657*, 2025. [2](#)