

# Prompting without Panic: Attribute-aware, Zero-shot, Test-Time Calibration

Ramya Hebbalaguppe<sup>1,2§</sup>   Tamoghno Kandar<sup>2§</sup>   Abhinav Nagpal<sup>1</sup>  
Chetan Arora<sup>1</sup>

<sup>1</sup>IIT Delhi   <sup>2</sup> TCS Research Labs

**Source code:**

[https://github.com/rhebbalaguppe/TCA\\_PromptWithoutPanic](https://github.com/rhebbalaguppe/TCA_PromptWithoutPanic)

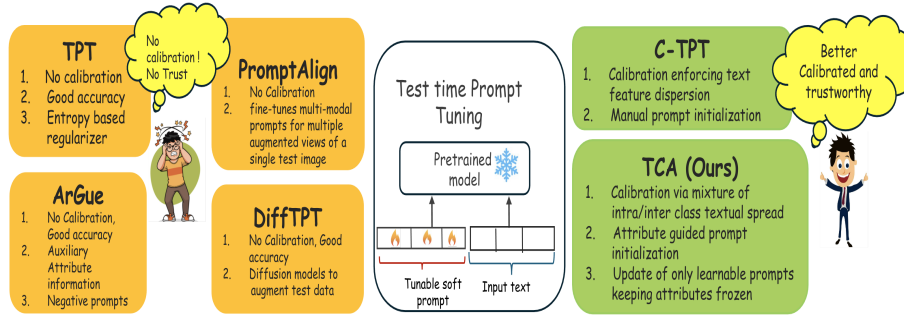
**Abstract.** Vision language models (VLMs) have become effective tools for image recognition, primarily due to their self-supervised training on large datasets. Their performance can be enhanced further through test-time prompt tuning (TPT). However, TPT’s singular focus on accuracy improvement often leads to a decline in confidence calibration, restricting its use in safety-critical applications. In this work, we make two contributions: **(1)** We posit that random or naive initialization of prompts leads to overfitting on a particular test sample, and is one of the reasons for miscalibration of VLMs after TPT. To mitigate the problem, we propose careful initialization of test time prompt using prior knowledge about the target label attributes from a large language model (LLM). **(2)** We propose a novel regularization technique to preserve prompt calibration during test-time prompt tuning (TPT). This method simultaneously minimizes intraclass distances while maximizing interclass distances between learned prompts. Our approach achieves significant calibration improvements across multiple CLIP architectures and 15 diverse datasets, demonstrating its effectiveness for TPT. We report an average expected calibration error (ECE) of 4.11 with our method, TCA, compared to 11.7 for vanilla TPT (24), 6.12 for C-TPT(48) (ICLR’24), 6.78 for DiffTPT(8) (CVPR’23), and 8.43 for PromptAlign(37) (NeurIPS’23).

## 1 Introduction

**VLMs and Confidence calibration.** Vision-Language Models (VLMs) have unlocked transformative applications across a wide range of fields, from healthcare diagnostics (45) to assistive solutions for visually impaired (46). However, recent findings (40) reveal that VLMs suffer from miscalibration, which can hinder model trustworthiness in critical applications. Traditional calibration methods rely on large labeled datasets, posing significant limitations for settings like test-time adaptation, where the labeled data is unavailable or infeasible to obtain. Inspired by the success of VLMs in generalizing to unseen data in a zero-shot setting (48), in this paper we focus on zero-shot setting, and adapt these models using prompt tuning.

---

<sup>§</sup> Equal contribution



**Fig. 1: Conceptual comparison between our proposed TCA vs. the contemporaries.** Test-time prompt tuning methods, such as TPT (24), learn test-time prompts through parameter optimization. However, these methods often face performance disadvantages in calibration, as they struggle to dynamically adapt to varying textual feature distributions, limiting effective prompt calibration. Methods, ArgGue(39), DiffTPT(8), and PromptAlign(37) do not explicitly optimize for calibration. Although C-TPT (48) introduces enhancements in calibration, it still falls short in capturing *nuanced visual attributes* that contribute to precise prompt conditioning leading to suboptimal prompt specificity. Our method termed **Test-time Calibration via Attribute Alignment (TCA)** infuses relevant attribute information providing context via LLMs and captures intra/inter-class textual attribute spread improving prompt calibration. **Note:** TCA works in zero-shot and test-time settings without any labeled data, making it very practical for real-world deployment where data annotation is infeasible. No model finetuning required: Only prompts are updated at test time; base vision and text encoders are kept frozen.

**Prompt Tuning.** Test-time prompt tuning (TPT) has emerged as a promising approach to improve generalization of VLMs, offering a way to adapt prompts to specific contexts without requiring any labeled data from the target domain. Hard prompts (27), often composed of fixed vocabulary tokens from standard templates like “A photo of a {class name}” can simplify prompt creation. However, (48) indicate that more flexible prompt designs, such as soft prompts or learned embeddings, can significantly enhance a model’s adaptability and effectiveness. On the other hand, domain-specific prompt creation for image-text models requires substantial expertise and time, with no guarantee of optimal results despite extensive engineering efforts(36). Shu et al.(24) suggested a TPT technique (hereinafter referred to as Vanilla TPT (VTPT)) which aims to enhance the accuracy of CLIP based models by minimizing the entropy in the prediction distribution as a self-supervision signal during test time. However, a reduction in entropy leads the model to generate overconfident predictions, a characteristic often observed in models trained with cross-entropy loss (9; 48). Fig. 1 illustrates the conceptual distinction between existing prompt tuning approaches and the method proposed in this work.

**Contributions.** This work focuses on TPT strategy to improve model’s calibration. At first, this may seem infeasible since various calibration techniques employed in standard supervised training of neural networks require substantial amounts of labeled training data, which restricts their applicability in test-time prompt tuning scenarios for CLIP based models. Here, we come up with a clever workaround, by extracting label attributes using a LLM, and leveraging them in TPT instead of label supervision.

1. **Attribute-Aware Prompting for Improved Calibration:** Unlike the contemporary methods that directly attach soft prompts before class names, we append the model with precise visual attributes produced by an LLM that provide rich context. The visual attributes are sorted by their **relevance**. It may be noted that a particular attribute may be relevant for more than one labels. Hence, by aligning the visual embeddings with the chosen attributes allows a model to not only demonstrate that it recognizes features that are crucial for distinguishing the correct class from others, but also allows the model to express its prediction uncertainty in terms of the ambiguous attributes. Multiple relevant attributes also enhance the compositional nature of visual data as they serve as semantic anchors. Their incorporation in soft prompt design improves image-text alignment scores as they establish interpretable correspondences between visual and linguistic embeddings.
2. **Regularization Loss:** Proposed visual attributes-based prompt initialization allows the model a much better starting point compared to random initialization and prevents overfitting in the presence of limited variations in the single sample (and its augmentation) based training. However, the gradient-based update of the prompts may still overfit the prompts to the sample. Hence, we propose a loss on text prompt embeddings to minimize intra-class text feature dispersion, while maximizing inter-class dispersion. The idea is inspired from contrastive learning (15) in supervised training where the intra-class distance w.r.t. anchor is minimized and inter-class distance w.r.t. negative sample is maximized. The proposed loss can be combined with other prompt tuning methods for e.g. PromptAlign (37) and DiffTPT (8). In supplementary, we show gains in accuracy and ECE when we incorporate TCA on top of PromptAlign (37) and DiffTPT (8).
3. **Superior Performance:** We perform extensive experiments across various datasets and CLIP based models, incorporating our proposed attributes aware prompt initialization, and proposed loss. We report an average performance on 11 benchmark datasets improving the model calibration by 7.5% over the baseline TPT (24) and 2.01% in terms of ECE over C-TPT (48) respectively.

## 2 Related Works

**Miscalibration in Neural Network.** Accurate estimation of predictive uncertainty, often referred to as model calibration, is a critical aspect of deploying neural networks in safety-sensitive applications. Proper calibration ensures that

the confidence associated with a model’s predictions aligns with its true accuracy, thereby facilitating more reliable decision-making. However, recent studies have highlighted frequent instances of miscalibration in modern neural network architectures, indicating a concerning trend: despite improvements in predictive performance, newer and more accurate models tend to produce poorly calibrated probability estimates (9; 41).

**Calibration Techniques.** Calibration techniques can be broadly classified as train-time methods and post-hoc methods. Train-time techniques typically used additional loss terms along with the NLL (cross-entropy) loss during training. Some representative works include: (31; 29). These techniques are not practical in our setting as it requires retraining the neural network with the regularization terms. Post-hoc calibration are applied after the model has been trained and often require a validation set to fine-tune the output probabilities. Some common post-hoc calibration techniques include: TS (32), DC (18) etc.

**Prompt Tuning for VLMs.** To efficiently adapt the large foundational models, prompting (21) has emerged as a resource-efficient method. Prompt tuning typically uses static or learnt prompts as part of the input text to guide the model in performing specific tasks in a zero-shot, or few-shot manner. Hand-crafted prompts consisting of predefined vocabulary tokens, or hard prompts, may not be optimal in various settings. Hence, there is a growing focus on techniques that regard prompts as learnable vectors which can be optimized through gradient descent (22). For instance, CoOp (51) tunes the prompts in CLIP using labeled training samples to improve its classification accuracy. However, CoCoOp (50) identified that CoOp struggles with generalizing to out-of-distribution data and recommends conditioning the prompt on input images. While effective, these methods require access to annotated training data, which limits the zero-shot adaptation of pre-trained models like ours. To tackle this challenge, recent research has introduced a TPT technique (24), which enables adaptive prompt learning at the inference time, using just one test sample. TPT optimizes the prompt by minimizing the entropy with confidence selection so that the model has consistent predictions for each test sample. DiffTPT (8) innovates test-time prompt tuning by leveraging pre-trained diffusion models to augment the diversity of test data samples used in TPT. PromptAlign (37) fine-tunes multi-modal prompts at test-time by aligning the distribution statistics obtained from multiple augmented views of a single test image with the training data distribution statistics. Although previous studies (51; 50; 2; 24) have primarily concentrated on refining prompt templates to improve accuracy, they have largely neglected calibration (9), except for (48).

Our paper focuses on the critical and under-explored challenge of calibrating VLMs in a **zero-shot, test-time setting**. To maintain the efficiency and practicality, we develop our solution within prompt tuning framework.

### 3 Proposed Method

#### 3.1 Preliminaries

**Confidence Calibration.** Given a data distribution  $\mathcal{D}$  of  $(x, y) \in \mathcal{X} \times \{0, 1\}$ , let  $c$  denote the predictive confidence of a predictor  $f : \mathcal{X} \rightarrow [0, 1]$ . The predictor is said to be calibrated (5), if:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}}[y \mid f(x) = c] = c, \quad \forall c \in [0, 1]. \quad (1)$$

Intuitively, if a network predicts a class ‘‘cancer’’ for an image with a score of 0.9, then a network is calibrated, if the probability that the image actually contains a cancer is 0.9. Expected Calibration Error (ECE) is a common metric used for measuring calibration, and evaluates how well the predicted confidence of a model align with its accuracy. To compute ECE, the confidence interval  $[0, 1]$  is divided into a fixed number of bins. Each bin encompasses a range of predicted confidence. ECE value is then computed as (25):

$$\text{ECE} = \sum_{k=1}^K \frac{|B_k|}{m} |\text{acc}(B_k) - \text{conf}(B_k)|,$$

where  $K$  is the number of bins,  $B_k$  is the set of samples,  $|B_k|$  is the number of samples,  $\text{acc}(B_k)$  is the prediction accuracy, and  $\text{conf}(B_k)$  is the average predictive confidence in bin  $k$ . A lower ECE is preferred.

**Zero-Shot Classification with CLIP.** Let  $\mathcal{X}$  be the image space, and  $\mathcal{Y}$  be the label space. Let  $t \in T$  be the text prompt corresponding to an image sample  $x \in \mathcal{X}$ . CLIP (34) architecture is composed of two distinct encoders: a visual encoder denoted by:  $f$ , and a text encoder  $g$ . In the vanilla zero-shot inference with CLIP, we attach a manually designed prompt prefix,  $\mathbf{p}$  (e.g.,  $\mathbf{p}$  = ‘‘a photo of a’’) to each possible class  $y_i \in \mathcal{Y} = \{y_1, y_2, \dots, y_K\}$ , generating class-specific textual descriptions  $t_i = [\mathbf{p}; y_i]$ . Here,  $K$  denotes the number of classes. Next, we generate text features  $g(t_i)$ , and image features  $f(x)$  by passing the relevant inputs to the respective encoders. This allows to compute the similarity between text feature, and image features as:  $s_i = s(f(x), g(t_i = [\mathbf{p}; y_i]))$ , where  $s(\cdot)$  refers to the cosine similarity. The probability of predicting class  $y_i$  for the test image  $x$  can be computed as:

$$p(y_i|x) = \frac{\exp(s(g(t_i), f(x))/\tau)}{\sum_{j=1}^K \exp(s(g(t_j), f(x))/\tau)},$$

where  $\tau$  is the temperature for the **softmax** function. The predicted class is  $\hat{y} = \arg \max_{y_i} p(y_i | x)$ , with predicted confidence  $\hat{p} = \max_{y_i} p(y_i | x)$ .

**Test-time Prompt Tuning.** Several researchers have demonstrated the efficacy of few shot prompt tuning in general (19; 14; 47; 49; 16), as well as for CLIP based models (51; 50; 2; 10). Test-time prompt tuning (Vanilla TPT (VTPT)) introduced by (24) aims to benefit from the rich knowledge of CLIP to boost

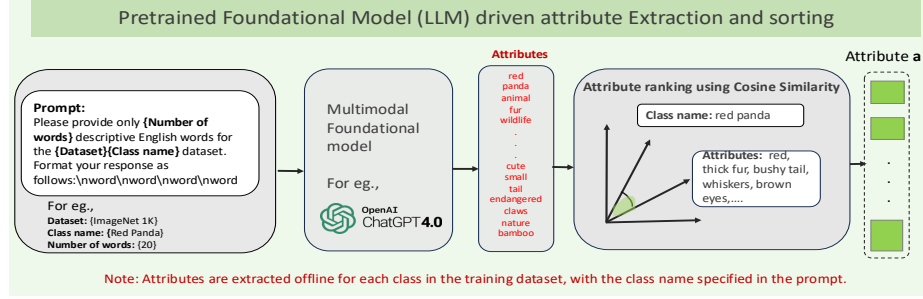


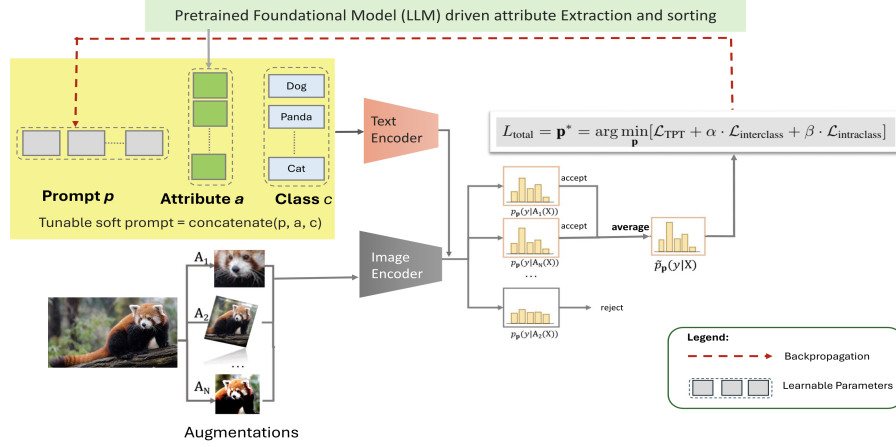
Fig. 2: Visual attributes are extracted by prompting a multimodal foundational model as shown in the leftmost block. The extracted attributes (shown in red) are ranked based on their similarity to the Class name in the Dataset (e.g., the top 20 attributes for "red panda" in ImageNet1K dataset). This offline process aids model calibration by identifying relevant attributes. The relevant attributes  $a \subset \{a_i\}_{i=1}^N$  by identifying the attribute similarity with respect to a class name. Here  $a_i$  is the set of attributes returned for a particular class by pretrained LLM.

its generalization in a zero-shot manner. optimizes prompts without requiring labeled data. During inference,  $N$  augmented views,  $x^j$ , of the test sample  $x$  are generated. Predictions with entropy values below a predefined threshold are retained, while those with higher entropy are discarded through a confidence selection filter. The entropy of the remaining predictions is then averaged, and this value is used to update the prompts in an unsupervised manner using back-propagation from the following the objective function (24).

$$\mathcal{L}_{\text{TPT}} = - \sum_{i=1}^K \bar{p}(y_i) \log \bar{p}(y_i), \quad \text{where } \bar{p}(y_i) = \frac{1}{N} \sum_{j=1}^N p(y_i | x^j). \quad (2)$$

Here,  $\bar{p}(\cdot)$  represents the mean of vector class probabilities produced by the model across different augmented views preserved after the confidence selection filter. Additionally, it has been shown that test-time prompt tuning can be effectively combined with few-shot prompt tuning techniques (during train time), further boosting vanilla VTPT's performance (24).

**Attribute Alignment using an LLM.** In VLMs, attribute alignment in prompt tuning guides the model to generate outputs matching specific visual or textual attributes. Authors in (33) use LLMs to create descriptive sentences highlighting key features of image categories. An attribute extractor identifies relevant domain-specific information like color or context (33; 26; 39), and the prompt is adjusted accordingly. This aligned prompt improves inference accuracy by tailoring the model to the task. Unlike the train-time techniques above, our approach focuses on test-time calibration.



**Fig. 3: Calibration using Test-time Attribute Alignment for zero-shot image classification:** In a typical test time prompt tuning for image classification, a category label is prefixed with a template text, such as “a photo of a” (e.g., “a photo of a red panda”) to generate the prompt for tuning. Our approach differs in the following ways: **(a)** Visual attributes are extracted as shown in Fig. 2. **(b)** Our approach takes an image and its augmentations ( $A_1, A_2, \dots, A_N$ ) as the input. In contrast to TPT (24), we utilize the attribute vector  $\mathbf{a}$  concatenated with template text  $\mathbf{p}$  and class name  $c_i$  to initialize the prompt. We introduce two auxiliary terms in the objective function for test-time calibration via attribute alignment:  $L_{\text{interclass}}$  to maximize mean text features between classes and  $L_{\text{intraclasse}}$  to minimize intra-class variance of textual attributes during prompt tuning to improve alignment between predicted and actual class probabilities, enhancing model calibration. This allows us to tune adaptive prompts on the fly with a single test sample, and without the need for additional training data or annotations. Both visual and text encoders are kept frozen while prompt tuning.

### 3.2 Test Time Calibration via Attribute Alignment

Our proposed attribute-aware prompt tuning procedure comprises of two steps, namely, **(a)** relevant attribute extraction (See Fig 2) ; **(b)** enhancing calibration via test-time loss on textual features separation/contraction (See Fig 3).

Fig. 2 depicts the first step, we obtain visual attributes that provide context by prompting LLMs with inquiries about the visual characteristics of specific classes. The LLM input exclusively consists of class names from a dataset. Formally, given any label  $y_i \in \mathcal{Y}$ , we retrieve its corresponding class name,  $c_i$ , and a list of attributes  $\mathbf{a}_{y_i} = \gamma(y_i)$  where  $\gamma$  is any language model like GPT4. The template for prompting LLM has been pre-defined (see Fig. 2). The attributes are subsequently ranked in descending order of relevance by sorting based on the cosine similarity between the class name and attribute names. We then store  $M$  most relevant attributes in the attribute vector  $\mathbf{a}_c$  (we use top 2 attribute in our

**Algorithm 1** Test-time Calibration via Attribute Alignment (**Inference**)

- 
- 1: Initialize manual prompt,  $\mathbf{p}$  = "a photo of a"
  - 2: Attribute  $a$  and class =  $c$
  - 3: **for** each class  $i \in \{1, \dots, K\}$  **do**
  - 4:   **for** each attribute  $j \in \{1, \dots, M\}$  **do**
  - 5:     Form text embedding  $t_{ij} = \mathbf{p} \oplus_j \oplus c_i$
  - 6:   **end for**
  - 7:   Compute the mean of text embeddings for each class  $\bar{t}_{y_i} = \frac{1}{M} \sum_{j=1}^M g(t_{ij})$ ,  
where  $g(\cdot)$  is the CLIP text encoder.
  - 8:   Calculate mean text attribute spread (MTAS) for class  $y_i$ :  $\text{MTAS}(y_i) = \frac{1}{M} \sum_{j=1}^M \|g(t_{ij}) - \bar{t}_{y_i}\|_2$
  - 9:    $\mathcal{L}_{\text{intra-class}}(y_i) = \text{MTAS}(y_i)$
  - 10: **end for**
  - 11: Compute the mean of text embeddings for all classes,  $\bar{\bar{t}} = \frac{1}{K} \sum_{i=1}^K \bar{t}_{y_i}$
  - 12: Calculate Average Text Feature Dispersion (ATFD) (48) across all classes:  $\text{ATFD} = \frac{1}{K} \sum_{i=1}^K \|\bar{\bar{t}} - \bar{t}_{y_i}\|_2$ .
  - 13:  $\mathcal{L}_{\text{inter-class}} = -\text{ATFD}$
  - 14:  $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{TPT}} + \alpha \cdot \mathcal{L}_{\text{inter-class}} + \beta \cdot \mathcal{L}_{\text{intra-class}}$ .
- 

implementation based on our ablation study). In Fig. 2 we illustrate this with an example of a "red panda" image. The attributes thus generated are appended to the tunable prompt,  $\mathbf{p}^1$ , along with the class names, such that tunable prompt = concatenate( $\mathbf{p}$ ,  $\mathbf{a}$ ,  $c$ ) (also see the block diagram corresponding to yellow box in Fig. 3. The full prompt text including the attributes are shown in the json file for Caltech 101 dataset included in the supplementary material. In step (b), to enforce effective calibration, we employ a contrastive loss at test-time, and a test-time calibration process as specified in Algorithm 1.

We start with the initialized prompts as described earlier, and then for every class  $i$  and attribute  $j$ , we form the text embedding  $\mathbf{p} \oplus a_j \oplus c_i$  and then compute the centroid of these text embeddings. We then minimize the distance between class centroid and textual embeddings corresponding to class (generated using different class attributes). This is referred to as intra-class loss and serves to learn most discriminative features of a class. Similar to C-TPT (48), we also increase the distance between text embeddings of distinct classes and this loss is referred to as inter-class loss. For this, we first take the mean of the embeddings corresponding to different attributes of a specific class. This represents the textual feature corresponding to a class. We then maximise the distance between these representative features of each class so that all classes are well separated. The overall loss used to tune the prompts is the summation of vanilla test time prompt tuning loss  $\mathcal{L}_{\text{TPT}}$  (24), and the above two loss terms. Note that the back-propagated gradients only update tokens corresponding to  $\mathbf{p}$ , whereas  $\mathbf{a}$ , and  $c_i$  tokens remain frozen, to prevent overfitting on the test sample.

---

<sup>1</sup> recall  $\mathbf{p}$  is generated from manual template text, such as "a photo of"



**Difference between TCA and other contemporary techniques.** Although prompt tuning through C-TPT (48) introduces enhancements in calibration, it still falls short in capturing nuanced class specific features which are important to disambiguate between classes, and thus necessary for uncertainty calibration. Though the sample specific labels are absent in the test time setting as ours, however we make a observation, and note that even then class specific information is indeed available. We make use of LLMs to generate class attributes and then use the proposed technique to choose most representative attributes. In another big difference, we choose not to update these attribute features. In C-TPT, firstly the text prompt initialization is same for all the classes, and then all of them get updated by the test-time loss, leading to overfitting on the sample, and less than ideal calibration. In our case, the frozen attribute based features provide adequate grounding and prevent overfitting, whereas other learnable prompts allow to adapt to the particular sample, thus leading to better calibration through proposed TCA over the current state-of-the-art, C-TPT. Our approach also differs from that of TPT (24), as they do not incorporate attribute auxiliary information from LLMs, nor do they explicitly optimize for calibration. As a result, their method exhibits sub-optimal calibration performance.

## 4 Experiments

This section outlines the benchmarks for assessing our method and the experimental results. Consistent with previous research on the prompt tuning of vision-language models (51; 50; 2; 24), our evaluation is centered on two primary aspects: (1) a range of fine-grained classifications and (2) the natural distribution shift. **Note:** In particular, given our objective to enhance calibration in the context of test-time prompt tuning, our experimental framework emphasizes prompt optimization in the absence of labeled training data.

**Datasets.** For fine-grained classification, we utilize a diverse set of datasets, including ImageNet (6), Caltech101 (7), OxfordPets (30), StanfordCars (17), Flowers102 (26), Food101 (1), FGVC Aircraft (23), SUN397 (44), UCF101 (38), DTD (4), and EuroSAT (11). For the out-of-distribution (OOD) generalization task, we define ImageNet (6) as the in-distribution (source) dataset and extend evaluation to four OOD variants: ImageNetV2 (35), ImageNet-Sketch (42), ImageNet-A(13), and ImageNet-R(12).

**Implementation Details.** We report results in following experimental configurations. The initialized prompt is set to a hard prompt ‘a photo of a’ ( $\text{CLIP}_{\text{HardPrompt}}$ ) and the corresponding 4 tokens are optimized based on a single test image using TPT ( $\text{TPT}_{\text{HardPrompt}}$ ) or jointly using TPT and our proposed technique TCA ( $\text{TPT}_{\text{HardPrompt}} + \text{TCA}$ ). We also include an ensemble setting where we average the logits from 4 different hard-prompt initialization using ‘a photo of a’, ‘a photo of the’, ‘a picture of a’, ‘a picture of the’ ( $\text{CLIP}_{\text{Ensemble}}$ ). Similarly, we optimize using TPT as well ( $\text{TPT}_{\text{Ensemble}}$ ), or jointly using TPT and TCA ( $\text{TPT}_{\text{Ensemble}} + \text{TCA}$ ) on each of the hard-prompt initialization and average the resulting logits. We have tried to use 1, 2, and 3 attribute

Method	Metric	ImageNet	Caltech	Pets	Cars	Flower	Food101	Aircraft	SUN397	DTD	EuroSAT	UCF101	Average
CLIP-RN50 <sub>HardPrompt</sub>	Acc.	58.1	85.8	83.8	55.7	61	74	15.6	58.6	40	23.7	58.4	55.9
	ECE	3.83	4.33	5.91	4.7	<b>3.19</b>	3.11	<b>6.45</b>	<b>3.54</b>	<b>9.91</b>	15.4	<b>3.05</b>	<b>5.61</b>
+TPT <sub>HardPrompt</sub>	Acc.	60.7	87	84.5	58	62.5	74.9	17	61.1	41.5	28.3	59.5	57.7
	ECE	11.4	5.04	<b>3.65</b>	3.76	13.4	5.25	16.1	9.24	25.7	22.5	12.4	11.7
+TPT <sub>HardPrompt</sub> +C-TPT	Acc.	60.2	86.9	84.1	56.5	65.2	74.7	17	61	42.2	27.8	59.7	57.8
	ECE	<u>3.01</u>	<u>2.07</u>	<b>2.77</b>	<b>1.94</b>	4.14	<b>1.86</b>	10.7	<b>2.93</b>	19.8	<u>15.1</u>	<b>3.83</b>	6.2
+TPT <sub>HardPrompt</sub> +TCA (2 Attribute)	Acc.	58.72	86.69	86.21	55.95	64.47	75.38	17.04	60.02	39.59	31.32	61.04	57.85
	ECE	<b>1.76</b>	<b>1.79</b>	5.43	<b>3.35</b>	<b>3.7</b>	<b>2.45</b>	<b>4.48</b>	4.32	<b>8.16</b>	<b>5.5</b>	4.33	<b>04.11</b>
+TPT <sub>Ensemble</sub>	Acc.	61.1	87.4	83.2	59.2	61.4	76.2	17.9	62	42.8	28.4	60.2	58.2
	ECE	11.2	4.29	4.79	3.08	14.1	5.27	14.6	7.68	22.2	18.9	11.1	10.7
+TPT <sub>Ensemble</sub> +C-TPT	Acc.	61.2	87.4	84	57.3	65.3	76	17.5	62.1	43.1	29.4	60.7	58.5
	ECE	<b>4.13</b>	<b>2.15</b>	<b>2.71</b>	<b>1.68</b>	<b>3.6</b>	<b>1.47</b>	<b>10.9</b>	<b>2.96</b>	<b>15.7</b>	<b>8.7</b>	<b>3.27</b>	5.2
+TPT <sub>Ensemble</sub> +TCA (2 Attributes)	Acc.	68.1	93.26	90.13	65.94	68.9	84.23	25.38	65.84	43.91	47.17	67.72	<b>65.50</b>
	ECE	<b>1.88</b>	<b>3.09</b>	<b>4.38</b>	<b>3.93</b>	<b>3.57</b>	<b>1.91</b>	<b>3.36</b>	<b>6.02</b>	<b>4.36</b>	<b>9.36</b>	<b>2.71</b>	<b>4.05</b>
CLIP-ViT-B/16 <sub>HardPrompt</sub>	Acc.	66.7	92.9	88	65.3	67.3	83.6	23.9	62.5	44.3	41.3	65	63.7
	ECE	2.12	5.5	<b>4.37</b>	<b>4.25</b>	3	2.39	5.11	2.53	8.5	7.4	3.59	4.43
+TPT <sub>HardPrompt</sub>	Acc.	69	93.8	87.1	66.3	69	84.7	23.4	65.5	46.7	42.4	67.3	65
	ECE	10.6	4.51	5.77	5.16	13.5	3.98	16.8	11.3	21.2	21.5	13	11.6
+TPT <sub>HardPrompt</sub> +C-TPT	Acc.	68.5	93.6	88.2	65.8	69.8	83.7	24	64.8	46	43.2	65.7	<b>64.8</b>
	ECE	3.15	<b>4.24</b>	<b>1.9</b>	<b>1.59</b>	5.04	<b>3.43</b>	<b>4.36</b>	<b>5.04</b>	11.9	13.2	<b>2.54</b>	5.13
+TPT <sub>HardPrompt</sub> +TCA (2 Attribute)	Acc.	67.37	92.86	90.51	65.92	69.18	69.18	25.32	65.5	44.73	45.58	66.9	<b>65.91</b>
	ECE	<b>2.27</b>	<b>3.01</b>	6.3	<b>7.85</b>	<b>3.67</b>	5.28	<b>3.6</b>	7.17	<b>5.48</b>	<b>8.37</b>	<b>2.82</b>	<b>5.07</b>
CLIP-ViT-B/16 <sub>Ensemble</sub>	Acc.	68.2	93.4	86.3	65.4	65.7	85.2	V23.5	64	45.6	43	66.1	64.2
	ECE	3.7	6.16	4.88	7.09	6.01	3.78	4.56	4.01	13.8	6.01	4.05	5.82
+TPT <sub>Ensemble</sub>	Acc.	69.6	94.1	86.1	67.1	67.6	85.1	24.4	66.5	47.2	44	68.5	65.5
	ECE	9.82	4.48	5.72	4	13.9	4.27	14.6	9.01	18.6	14.1	10.5	9.91
+TPT <sub>Ensemble</sub> +C-TPT	Acc.	69.3	94.1	87.4	66.7	69.9	84.5	23.9	66	46.8	48.7	66.7	65.8
	ECE	4.48	3.14	<b>1.54</b>	<b>1.84</b>	5.77	<b>2.38</b>	6.4	<b>3.09</b>	13.7	<b>5.49</b>	<b>3.04</b>	<b>4.62</b>
+TPT <sub>Ensemble</sub> +TCA 2 attributes	Acc.	68.1	93.26	90.13	65.94	68.9	84.23	25.38	65.84	43.91	47.17	67.72	<b>65.5</b>
	ECE	<b>1.88</b>	<b>3.09</b>	4.38	3.93	<b>3.57</b>	<b>1.91</b>	<b>3.36</b>	6.02	<b>4.36</b>	9.36	<b>2.71</b>	<b>4.05</b>

Table 1: **Fine-Grained Classification.** Results for CLIP-RN50 and CLIP-ViT-B/16 are reported, providing the **Accuracy** ( $\uparrow$ ) and **ECE** ( $\downarrow$ ) metrics for different experimental configuration (please see main test for configuration details). The values highlighted in **bold** indicate the lowest ECE achieved following test-time prompt tuning and **underline** is the second best. **Note:** The full table, which includes comparisons with other contemporary methods, can be found in the supplementary material due to space limitations in the main paper - we ablate TCA loss with promptAlign(37) and DiffTPT(8) to show gains on top of contemporary methods PromptAlign (NeurIPS’24) and DiffTPT (ICCV’23)

initialization. **Hyperparameters  $\alpha$  and  $\beta$ :** We employ a test-time prompt tuning strategy, which does not allow access to data for hyperparameter tuning. We perform a grid search over  $\alpha$  and  $\beta$  to balance the calibration loss for the least ECE using Caltech 101 dataset and apply the same values for 11 datasets following a setup similar to C-TPT(48). We obtain  $(\alpha, \beta)$  as  $(10, 35)$ , respectively. Using 2 attributes gave the best ECE values on majority of the datasets for finegrained classification. For Natural distribution shifts, we obtained,  $(\alpha, \beta)$  as  $(45, 15)$ . For TPT (24), we optimize the prompt in one step using the AdamW optimizer with a learning rate of 0.005. Our method runs on a single NVIDIA Tesla V100 GPU with 32GB of memory, except for the ImageNet, ImageNet-A, and ImageNet V2 datasets, which use two GPUs for evaluation.

#### 4.1 Comparison on Fine Grained Classification

For the fine-grained classification task, we compare contemporary methods against hard prompt and benchmark approaches, such as TPT (24) and C-TPT (48). Tab. 1 summarizes the results: accuracy and ECE values. Our evaluation includes multiple CLIP architectures, specifically CLIP RN-50 and ViT-B/16. The results show that our method significantly outperforms the hard prompt configuration. When comparing the average performance of C-TPT across all 11 datasets, our method

Methods	Metric	IN-A	IN-V2	IN-R	IN-S	Avg.
CLIP-RN50 <sub>HardPrompt</sub>	Acc.	21.7	51.4	56	33.3	40.6
	ECE	21.3	3.33	2.07	3.15	7.46
+TPT <sub>HardPrompt</sub>	Acc.	25.2	54.6	58.9	35.1	43.5
	ECE	31.0	13.1	9.18	13.7	16.7
+TPT <sub>HardPrompt</sub> +C-TPT	Acc.	23.4	54.7	58	35.1	42.8
	ECE	25.4	8.58	4.57	9.7	12.1
+TPT <sub>HardPrompt</sub> +TCA (2 Attributes)	Acc.	20.77	51.74	54.83	32.83	40.04
	ECE	<b>22.53</b>	<b>4.39</b>	<b>1.25</b>	<b>6.22</b>	<b>8.59</b>
CLIP-RN50_Ensemble	Acc.	22.7	52.5	57.9	34.7	42
	ECE	17	2.68	5.64	10.9	9.06
+TPT <sub>Ensemble</sub>	Acc.	26.9	55	60.4	35.6	44.5
	ECE	29.1	12.7	7.5	14	15.8
+TPT <sub>Ensemble</sub> +C-TPT	Acc.	25.6	54.8	59.7	35.7	44
	ECE	27	9.84	5.17	12.2	13.6
+TPT <sub>Ensemble</sub> +TCA (2 Attributes)	Acc.	21.12	51.8	55.57	33.11	40.4
	ECE	<b>22.99</b>	<b>3.69</b>	<b>0.94</b>	<b>5.37</b>	<b>8.24</b>
CLIP-ViT-B/16 <sub>HardPrompt</sub>	Acc.	47.8	60.8	74	46.1	57.2
	ECE	8.61	3.01	3.58	4.95	5.04
+TPT <sub>HardPrompt</sub>	Acc.	52.6	63	76.7	47.5	59.9
	ECE	16.4	11.1	4.36	16.1	12
+TPT <sub>HardPrompt</sub> +C-TPT	Acc.	51.6	62.7	76	47.9	59.6
	ECE	<b>8.16</b>	6.23	<b>1.54</b>	<b>7.35</b>	<b>5.82</b>
+TPT <sub>HardPrompt</sub> +TCA (2 Attributes)	Acc.	46.95	59.94	72.78	45.1	56.19
	ECE	8.59	<b>4.95</b>	5.1	8.62	6.81
CLIP-ViT-B/16 <sub>Ensemble</sub>	Acc.	50.9	62	74.5	46	58.4
	ECE	8.85	3.01	2.85	9.7	6.1
+TPT <sub>Ensemble</sub>	Acc.	54.2	63.9	78.2	48.5	61.2
	ECE	13.5	11.2	3.64	15.3	10.9
+TPT <sub>Ensemble</sub> +C-TPT	Acc.	52.9	63.4	78	48.5	60.7
	ECE	10.9	8.38	<b>1.4</b>	12.6	8.32
+TPT <sub>Ensemble</sub> +TCA (2 attributes)	Acc.	47.36	60.85	72.74	45.72	56.66
	ECE	<b>5.21</b>	<b>1.81</b>	3.42	<b>4.81</b>	<b>3.81</b>

Table 2: **Natural Distribution Shifts.** Results for CLIP-RN50 and CLIP-ViT-B/16 are reported, providing the **Acc.** ( $\uparrow$ ) and **ECE** ( $\downarrow$ ) metrics for different experimental configurations (please refer to the main text for details of configurations). Dataset abbreviations: ImageNet-V2 (IN-V2), ImageNet-A (IN-A), ImageNet-R (IN-R), and ImageNet-Sketch (IN-S). Values highlighted in **bold** indicate the lowest ECE achieved after test-time prompt tuning.

achieves a similar average predictive accuracy while notably reducing the average ECE. For CLIP RN-50, the ECE decreases from 5.6 to 4.11. Similarly, for ViT-B/16, the ECE is reduced from 5.82 to 4.05.

## 4.2 Robustness to Natural Distribution Shifts

We follow the setting in Radford et al.(34) and evaluate model’s robustness to natural distribution shifts on 4 ImageNet Variants which have been considered as OOD for ImageNet in previous works. We report the results in Table 2. The table shows that we outperform contemporary methods (TPT, and C-TPT) in terms of ECE on 3 out of 4 datasets.

## 4.3 Ablation Study

We investigate the factors contributing to calibration— whether it is driven by the inclusion of attributes or by the choice of loss function. To examine this,

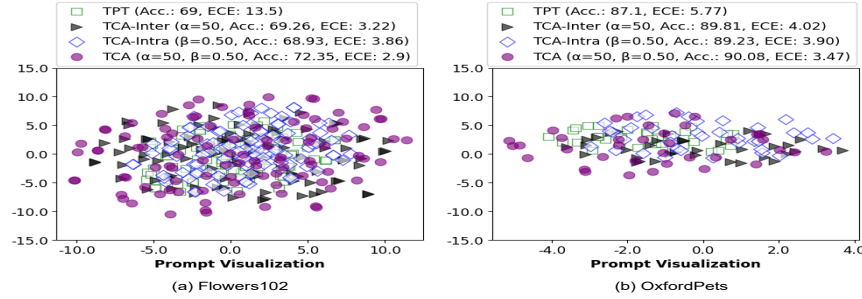


Fig. 4: The t-SNE plot shows Class-specific Text Embeddings on tuned prompts. We conduct ablation on each term of  $L_{\text{total}} = \mathbf{p}^* = \arg \min_{\mathbf{p}} [\mathcal{L}_{\text{TPT}} + \alpha \cdot \mathcal{L}_{\text{inter-class}} + \beta \cdot \mathcal{L}_{\text{intra-class}}]$  to understand its relative contribution empirically. In (a) and (b), notice that incorporating all three terms in  $L_{\text{total}}$  results in the lowest ECE and highest feature dispersion or spread.

we conducted an experiment under two conditions. In the first condition, we incorporate attributes into the prompts and evaluate the method using the TPT loss function. In the second, we again incorporate attributes into the prompts but evaluate using the combined TPT + TCA loss function on 3 datasets.

#### Relative Contribution of Attribute Initialization and Proposed Loss.

To better understand the contribution we conduct the ablation experiments on DTD dataset using ResNet50 feature extractor and report (Acc  $\uparrow$ , ECE  $\downarrow$ ). We have 3 variants: (a) +TPT<sub>HardPrompt</sub> (41.5, 25.7), (b) +TPT<sub>HardPrompt</sub> + initialization with 2 attributes (40.96, 20.45), (c) +TPT<sub>HardPrompt</sub> + initialization with 2 attributes + proposed TCA loss (42.79, 5.59). The key observations with ablation are as follows: **(1.) Attribute Initialization:** When initialized with 2 attributes, there was a 20.6% reduction in ECE compared to the hard prompt model; **(2.) TCA Loss:** Introduction of the TCA loss resulted in a  $3.65\times$  reduction in ECE, bringing ECE down from 25.7 to 5.59, significantly improving the model’s calibration. **(3.) Combined Effect of Both:** When both attribute initialization and TCA loss were used together, the ECE reduction was even more pronounced, with an overall  $4.59\times$  reduction in ECE, yielding the lowest ECE value of 5.59 and maximum accuracy of 42.79. Thus, both proposed contributions, attribute initialization strategy, as well as the proposed loss play significant roles in improving model calibration. The proposed loss is particularly effective in reducing ECE, and combining it with attribute initialization leads to the most significant improvement in both accuracy and calibration. Refer to Fig. 4, which illustrates the comparison of feature dispersion, found to be inversely correlated with ECE. When both inter- and intra-loss terms are utilized, we observe the maximum Class-specific Text Embedding dispersion and the lowest ECE, consistent with the findings of (48). See suppl. for details on how the plot was obtained.

#### 4.4 Discussion

**Confidence Calibration and TCA:** Here, we provide an intuitive understanding of our proposed loss function, formulated as:  $L_{\text{total}} = \mathbf{p}^* = \arg \min_{\mathbf{p}} [\mathcal{L}_{\text{TPT}} + \alpha \cdot \mathcal{L}_{\text{inter-class}} + \beta \cdot \mathcal{L}_{\text{intra-class}}]$ . To assess the significance of each component within this formulation, we conduct a systematic ablation study. This includes t-SNE visualizations, which facilitate the analysis of the impact of individual loss terms on feature separability and clustering. Additionally, we compare our approach against state-of-the-art test-time calibration methods in the zero-shot setting, thereby demonstrating its effectiveness and robustness.

**Need for intra-inter class losses:** TCA improves representation quality by leveraging contrastive principles thus enabling the generation of high-quality, discriminative embeddings that effectively capture semantic similarity/dissimilarity. TCA addresses calibration by aligning similar classes, and the use of the dispersion term explicitly penalizes the embedding overlap for dissimilar classes. This discourages the model from assigning overly confident probabilities to incorrect predictions, ensuring that extreme predictive probabilities (near 0 or 1) are only assigned when the different classes are well-separated. Fig. 4 shows an ablation over individual loss terms’ impact on calibration: Using both  $\mathcal{L}_{\text{intra}}$  and  $\mathcal{L}_{\text{inter}}$  in  $\mathcal{L}_{\text{total}}$  leads to the lowest ECE and greatest text feature dispersion.

**Conceptual differences between TCA Loss and Contemporaries:** The recent contemporary method, DAPT(3) targets improved accuracy in few-shot settings, whereas we focus on zero-shot calibration. DAPT uses exponential inter- and intra-dispersion on both vision and text embeddings, while our method relies on  $L_2$  norm distance between the test sample and mean text embeddings.  $L_2$  norm is easier to interpret as it measures the Euclidean distance between embeddings, making it more intuitive and transparent, especially when comparing distances in high-dimensional spaces, but less sensitive to outliers and computationally efficient. (20) facilitates calibration using temperature scaling on the ImageNet validation set. However, when applying TS with TCA loss on the Caltech 101 dataset (ViT B-16), we observe a degradation in (Accuracy,ECE) from 93.02, 12.92 with TS vs. 92.45, 3.89 without TS, suggesting a decrease in performance with TS. (43) uses Distribution aware calibration for fine-tuned VLM calibration, while our focus is on zero-shot settings like C-TPT(48). Finally, (28) involves few-shot finetuning, making it not directly comparable to our approach.

**Vizualisation of Class-specific Text Embeddings on tuned prompts.** Please refer to the supplemental materials for t-SNE plots across multiple datasets, which illustrate the lower ECE and the highest dispersion indicating better class separability of TCA relative to contemporaneous methods.

**Supplementary Material** details the factors behind TCA’s superior performance, datasets, metrics, feature extractor, experimental setup, hyperparameters, and t-SNE comparisons with PromptAlign (37), DiffTPT (8).

## 5 Conclusions and Future directions

In this work, we introduced two key insights to enhance the effectiveness of test time prompt tuning. First, we demonstrated that attribute-aware prompting, wherein relevant visual attributes are appended to the prompts. This allows the model to better align its visual embeddings with discriminative features, resulting in improved predictive uncertainty handling and class-separation. Second, we proposed a regularization loss that encourages the model to minimize intra-class text feature dispersion while maximizing inter-class dispersion, inspired by contrastive learning principles. This ensures that the learned prompts do not overfit to individual samples, even when limited data is available.

This work opens up new possibilities for leveraging unsupervised attribute information to improve model performance in low-data or test-time settings, paving the way for more robust and adaptable models in real-world applications.

## Bibliography

- [1] Bossard, L., Guillaumin, M., Gool, L.: Food-101: mining discriminative components with random forests. In: ECCV (2014) [9](#)
- [2] Chen, G., Yao, W., Song, X., Li, X., Rao, Y., Zhang, K.: PLOT: Prompt learning with optimal transport for vision-language models. In: ICLR (2023) [4](#), [5](#), [9](#)
- [3] Cho, E., Kim, J., Kim, H.J.: Distribution-aware prompt tuning for vision-language models. In: Proceedings of the IEEE/CVF ICCV. pp. 22004–22013 (2023) [13](#)
- [4] Cimpoi, M., Maji, S., Kokkinos, I., S., Vedaldi, A.: Describing textures in the wild. In: CVPR. pp. 3606–3613 (2014) [9](#)
- [5] Dawid, A.P.: The well-calibrated bayesian. Journal of the American Statistical Association (1982) [5](#)
- [6] Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009) [9](#)
- [7] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: CVPR Workshops. p. 178 (2004) [9](#)
- [8] Feng, C.M., Yu, K., Liu, Y., Khan, S., Zuo, W.: Diverse data augmentation with diffusions for effective test-time prompt tuning. ICCV (2023) [1](#), [2](#), [3](#), [4](#), [10](#), [13](#)
- [9] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: ICML. vol. 70, pp. 1321–1330 (2017) [2](#), [4](#)
- [10] Hantao Yao, Rui Zhang, C.X.: Visual-language prompt tuning with knowledge-guided context optimization. In: CVPR (2023) [5](#)
- [11] Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. **12**(7), 2217–2226 (2019) [9](#)
- [12] Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. ICCV (2021) [9](#)
- [13] Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. CVPR (2021) [9](#)
- [14] Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: ECCV. pp. 709–727 (2022) [5](#)
- [15] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. NeurIPS **33**, 18661–18673 (2020) [3](#)
- [16] Koo, G., Yoon, S., Yoo, C.D.: Wavelet-guided acceleration of text inversion in diffusion-based image editing. arXiv preprint arXiv:2401.09794 (2024) [5](#)

- [17] Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: IEEE Workshop on 3D Representation and Recognition (3dRR-13) (2013) [9](#)
- [18] Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., Flach, P.: Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *NeurIPS* **32** (2019) [4](#)
- [19] Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: EMNLP. pp. 3045–3059 (Nov 2021) [5](#)
- [20] LeVine, W., Pikus, B., Raja, P., Gil, F.A.: Enabling calibration in the zero-shot inference of large vision-language models. *arXiv preprint arXiv:2303.12748* (2023) [13](#)
- [21] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys* **55**(9), 1–35 (2023) [4](#)
- [22] Liu, X., Ji, K., Fu, Y., Tam, W.L., Du, Z., Yang, Z., Tang, J.: P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks (2022) [4](#)
- [23] Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. *Tech. rep.* (2013) [9](#)
- [24] Manli, S., Weili, N., De-An, H., Zhiding, Y., Tom, G., Anima, A., Chaowei, X.: Test-time prompt tuning for zero-shot generalization in vision-language models. In: *NeurIPS* (2022) [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#)
- [25] Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: *AAAI*. p. 2901–2907 (2015) [5](#)
- [26] Nilsback, M., Zisserman, A.: Automated flower classification over a large number of classes. In: *ICVGIP*. pp. 722–729 (2008) [6](#), [9](#)
- [27] Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., Tan, M.: Efficient test-time model adaptation without forgetting. In: *ICLR*. vol. 162, pp. 16888–16905 (17–23 Jul 2022) [2](#)
- [28] Oh, C., Lim, H., Kim, M., Han, D., Yun, S., Choo, J., Hauptmann, A., Cheng, Z.Q., Song, K.: Towards calibrated robust fine-tuning of vision-language models. *Advances in Neural Information Processing Systems* **37**, 12677–12707 (2024) [13](#)
- [29] Park, H., Noh, J., Oh, Y., Baek, D., Ham, B.: Acls: Adaptive and conditional label smoothing for network calibration. In: *ICCV*. pp. 3936–3945 (2023) [4](#)
- [30] Parkhi, O., Vedaldi, A., A.Zisserman, Jawahar, C.V.: Cats and dogs. In: *CVPR* (2012) [9](#)
- [31] Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548* (2017) [4](#)
- [32] Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: *ADVANCES IN LARGE MARGIN CLASSIFIERS*. pp. 61–74. MIT Press (1999) [4](#)
- [33] Pratt, S., Covert, I., Liu, R., Farhadi, A.: What does a platypus look like? generating customized prompts for zero-shot image classification. In: *ICCV*. pp. 15691–15701 (2023) [6](#)



- [34] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Aspell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763 (2021) [5](#), [11](#)
- [35] Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: ICML. pp. 5389–5400 (2019) [9](#)
- [36] S., G., Basu, S., Feizi, S., Manocha, D.: Intcoop: Interpretability-aware vision-language prompt tuning. In: EMNLP (2024) [2](#)
- [37] Samadh, A.: Align your prompts: Test-time prompting with distribution alignment for zero-shot generalization. *NeurIPS* **36** (2024) [1](#), [2](#), [3](#), [4](#), [10](#), [13](#)
- [38] Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR* **abs/1212.0402** (2012) [9](#)
- [39] Tian, X., Zou, S., Yang, Z., Zhang, J.: Argue: Attribute-guided prompt tuning for vision-language models. In: CVPR. pp. 28578–28587 (2024) [2](#), [6](#)
- [40] Tu, W., Deng, W., Campbell, D., Gould, S., Gedeon, T.: An empirical study into what matters for calibrating vision-language models (2024) [1](#)
- [41] Wang, D.B., Feng, L., Zhang, M.L.: Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *NeurIPS* **34**, 11809–11820 (2021) [4](#)
- [42] Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. In: *NeurIPS* (2019) [9](#)
- [43] Wang, S., Wang, J., Wang, G., Zhang, B., Zhou, K., Wei, H.: Open-vocabulary calibration for fine-tuned clip. *arXiv preprint arXiv:2402.04655* (2024) [13](#)
- [44] Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: CVPR. pp. 3485–3492 (2010) [9](#)
- [45] Y., N.t.: Multimodal healthcare ai: Identifying and designing clinically relevant vision-language applications for radiology. In: CHI Conference on Human Factors in Computing Systems (2024) [1](#)
- [46] Yi, Z., Yilin, Z., Rong, X., Jing, L., Hillming, L.: Vialm: A survey and benchmark of visually impaired assistance with large models. *arXiv preprint arXiv:2402.01735* (2024) [1](#)
- [47] Yoon, E., Yoon, H.S., Harvill, J., Hasegawa-Johnson, M., Yoo, C.D.: IN-Tapt: Information-theoretic adversarial prompt tuning for enhanced non-native speech recognition (2023) [5](#)
- [48] Yoon, H.S., Yoon, E., Tee, J.T.J., Hasegawa-Johnson, M.A., Li, Y., Yoo, C.D.: C-TPT: Calibrated test-time prompt tuning for vision-language models via text feature dispersion. In: ICLR (2024) [1](#), [2](#), [3](#), [4](#), [8](#), [9](#), [10](#), [12](#), [13](#)
- [49] Yoon, S., Koo, G., Hong, J.W., Yoo, C.D.: Neutral editing framework for diffusion-based video editing. *arXiv preprint arXiv:2312.06708* (2023) [5](#)
- [50] Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: CVPR (2022) [4](#), [5](#), [9](#)
- [51] Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. *IJCV* (2022) [4](#), [5](#), [9](#)