

---

# Enhancing Visual Prompting through Expanded Transformation Space and Overfitting Mitigation

---

**Shohei Enomoto**

NTT

Tokyo, Japan

[shohei.enomoto@ntt.com](mailto:shohei.enomoto@ntt.com)

## Abstract

Visual prompting (VP) has emerged as a promising parameter-efficient fine-tuning approach for adapting pre-trained vision models to downstream tasks without modifying model parameters. Despite offering advantages like negligible computational overhead and compatibility with black-box models, conventional VP methods typically achieve lower accuracy than other adaptation approaches. Our analysis reveals two critical limitations: the restricted expressivity of simple additive transformation and a tendency toward overfitting when the parameter count increases. To address these challenges, we propose ACAVP (Affine, Color, and Additive Visual Prompting), which enhances VP’s expressive power by introducing complementary transformation operations: affine transformation for creating task-specific prompt regions while preserving original image information, and color transformation for emphasizing task-relevant visual features. Additionally, we identify that overfitting is a critical issue in VP training and introduce TrivialAugment as an effective data augmentation, which not only benefits our approach but also significantly improves existing VP methods, with performance gains of up to 12 percentage points on certain datasets. This demonstrates that appropriate data augmentation is universally beneficial for VP training. Extensive experiments across twelve diverse image classification datasets with two different model architectures demonstrate that ACAVP achieves state-of-the-art accuracy among VP methods, surpasses linear probing in average accuracy, and exhibits superior robustness to distribution shifts, all while maintaining minimal computational overhead during inference. Our code is available at <https://github.com/s-enmt/ACAVP>.

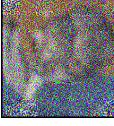
## 1 Introduction

Visual prompting (VP) [Bahng et al., 2022, Elsayed et al., 2019] has emerged as a promising parameter-efficient fine-tuning technique for adapting pre-trained vision models to downstream tasks. By learning task-specific noise applied to input images while keeping the model parameters frozen, VP offers significant advantages: it introduces negligible computational overhead and can be applied to black-box models where internal parameters are inaccessible Tsai et al. [2020], Oh et al. [2023].

Despite these advantages, VP methods typically achieve lower accuracy than full fine-tuning [Bahng et al., 2022]. A straightforward approach to improving VP accuracy is to increase the size of the prompt region, thereby enhancing the expressive power through increased parameterization. Theoretically, since the transformation space with increased parameters becomes a superset of the original transformation space, the approximation error should decrease with more parameters [Cai et al., 2024b]. However, as our preliminary experiments show, this approach is ineffective. As illustrated in Table 1, increasing the number of VP parameters does not improve training accuracy,

and more importantly, leads to lower test accuracy, indicating overfitting issues rather than enhanced expressive power.

Table 1: Results of preliminary experiments comparing performance between padding-type noise addition and image-sized noise addition.

Image		
Params	23,280	50,176
Train Acc	$97.53 \pm 0.14$	$97.35 \pm 0.18$
Test Acc	$93.65 \pm 0.04$	$89.52 \pm 0.12$

The affine transformation component generates and rotation while preserving the original image information. In contrast, the color transformation component enhances the original image information through adjustments to brightness and contrast. Importantly, both components retain the advantages of VP technology because of their negligible computational cost compared to model inference. Furthermore, to address the overfitting problem, we comprehensively evaluated various regularization techniques for VP training. After experimenting with several strategies, including dropout, MSE loss regularization, and weight decay, we found that data augmentation methods, particularly TrivialAugment [Müller and Hutter, 2021], are most effective for mitigating overfitting in VP training. TrivialAugment provides a low-cost yet powerful approach to increase input diversity during training, significantly stabilizing the learning process and improving generalization performance.

We evaluated ACAVP’s effectiveness across twelve image classification datasets using two different architectures. Our method outperforms existing VP approaches and achieves higher average accuracy than linear probing. Additionally, ACAVP demonstrates superior out-of-distribution (OOD) performance due to its improved generalization capabilities. The computational overhead of ACAVP is negligible compared to the inference time of the recognition model.

Our main contributions are as follows:

- We propose ACAVP, a novel VP method that enhances the expressive power of visual prompting by introducing complementary transformation operations: affine transformation for creating task-specific prompt regions while preserving original image information, and color transformation for emphasizing task-relevant visual features. This approach achieves state-of-the-art accuracy among VP methods while maintaining minimal computational overhead.
- We identify and address the critical overfitting problem in VP by empirically evaluating regularization techniques. We demonstrate that TrivialAugment effectively mitigates overfitting in ACAVP and existing VP methods, providing a general technique for improving VP performance.
- We evaluated our approach through extensive experiments across twelve diverse image classification datasets and two different model architectures. Our experiments demonstrate that ACAVP achieves the highest average accuracy among VP methods, surpasses linear probing in average accuracy, and exhibits superior robustness to distribution shifts in out-of-distribution scenarios.

## 2 Related Work

Visual prompting (VP) [Bahng et al., 2022] has emerged as a promising parameter-efficient fine-tuning paradigm. First introduced as adversarial reprogramming (AR) by Elsayed et al. [2019], VP adapts pre-trained models to downstream tasks by applying learnable prompts to input images while keeping the model parameters frozen. This approach enables efficient adaptation without modifying the underlying model architecture.

Our analysis reveals two critical limitations in VP: (1) simply adding more parameters with the traditional additive transformation has inherent limitations in its expressive power, and (2) increasing parameter count leads to overfitting issues. This leads us to a key question. How can we enhance the expressive power of VP while effectively preventing overfitting?

To address this challenge, we propose ACAVP (Affine, Color, and Additive Visual Prompting), a novel VP transformation design that enhances accuracy while maintaining minimal computational overhead. ACAVP incorporates two parameterized image transformation techniques into existing VP methods: affine and color transformations.

VP regions through operations such as resizing

and rotation while preserving the original image information. In contrast, the color transformation component enhances the original image information through adjustments to brightness and contrast. Importantly, both components retain the advantages of VP technology because of their negligible computational cost compared to model inference. Furthermore, to address the overfitting problem, we comprehensively evaluated various regularization techniques for VP training. After experimenting with several strategies, including dropout, MSE loss regularization, and weight decay, we found that data augmentation methods, particularly TrivialAugment [Müller and Hutter, 2021], are most effective for mitigating overfitting in VP training. TrivialAugment provides a low-cost yet powerful approach to increase input diversity during training, significantly stabilizing the learning process and improving generalization performance.

We evaluated ACAVP’s effectiveness across twelve image classification datasets using two different architectures. Our method outperforms existing VP approaches and achieves higher average accuracy than linear probing. Additionally, ACAVP demonstrates superior out-of-distribution (OOD) performance due to its improved generalization capabilities. The computational overhead of ACAVP is negligible compared to the inference time of the recognition model.

Our main contributions are as follows:

- We propose ACAVP, a novel VP method that enhances the expressive power of visual prompting by introducing complementary transformation operations: affine transformation for creating task-specific prompt regions while preserving original image information, and color transformation for emphasizing task-relevant visual features. This approach achieves state-of-the-art accuracy among VP methods while maintaining minimal computational overhead.
- We identify and address the critical overfitting problem in VP by empirically evaluating regularization techniques. We demonstrate that TrivialAugment effectively mitigates overfitting in ACAVP and existing VP methods, providing a general technique for improving VP performance.
- We evaluated our approach through extensive experiments across twelve diverse image classification datasets and two different model architectures. Our experiments demonstrate that ACAVP achieves the highest average accuracy among VP methods, surpasses linear probing in average accuracy, and exhibits superior robustness to distribution shifts in out-of-distribution scenarios.

**Transformation design for VP.** Although the original VP adds padding-type noise to the image, several studies have focused on improving its transformation design. Wu et al. [2024] introduced enhanced visual prompting (EVP), which preserves original image information by resizing the image and adding prompts around it. Building on this concept, Tsao et al. [2024] proposed AutoVP, an end-to-end framework that automates VP design choices and introduces learnable input scaling parameters to optimize the resize intensity dynamically. In a different direction, Jin et al. [2025] developed Low-Rank matrix multiplication for visual prompting (LoR-VP), which uses low-rank matrix decomposition to ensure that visual prompts influence all image patches while considering induced biases between patches. Rather than using a single prompt for an entire task, several approaches generate customized prompts for individual samples. Oh et al. [2023] proposed Coordinator, which uses an encoder-decoder network to generate sample-specific prompts. Similarly, Cai et al. [2024b] proposed sample-specific multi-channel masks (SMM) that generate appropriate prompt regions for each individual sample. Cho et al. [2024] expanded this concept by generating prompts in both spatial and frequency domains using a network for sample-specific adaptation. While sample-specific VP approaches achieve high expressive power, they significantly increase inference-time computational overhead, undermining one of VP’s primary advantages. In contrast, our work focuses on task-level visual prompting while addressing its limitations.

**Learning strategies for VP.** Researchers have also explored improved learning strategies for VP. Wu et al. [2024] incorporated techniques from adversarial attacks, specifically input diversity [Xie et al., 2019] and gradient normalization [Goodfellow et al., 2015], to improve training stability. Cai et al. [2025] proposed attribute-based visual reprogramming (AttrVR), leveraging descriptive and distinctive attributes to mitigate ambiguities caused by fixed template prompts and promote more context-aware alignment between images and attributes. Zhang et al. [2024] introduced transferable visual prompting (TVP), which learns a single VP to improve performance across diverse multimodal large language models. In the non-CLIP model, a random class mapping is used to align the pre-trained model classes with the downstream tasks’ classes. Wu et al. [2024] proposed a method that first infers the pre-trained model on a downstream dataset and then performs label mapping according to its output distribution. Chen et al. [2023] introduced iterative label mapping (ILM), which automatically remaps source labels to target labels, progressively improving VP performance on target tasks. Taking a probabilistic approach, Cai et al. [2024a] proposed bayesian-guided label mapping (BLM), which constructs a sequentially updated stochastic label mapping matrix quantifying pairwise relationships between pre-trained labels and downstream labels. In this study, we show that VP training is less accurate due to overfitting, and that countermeasures, especially data augmentation, are the most effective learning strategy.

We propose a novel VP design that enhances expressive power while minimizing the computational cost increase in inference time. Additionally, we introduce data augmentation techniques to mitigate overfitting, a common issue in VPs. Our approach balances expressive power and computational efficiency, maintaining VP’s key advantages while improving performance.

### 3 Proposed Method: ACAVP

We propose a novel VP approach, ACAVP (Affine, Color, and Additive Visual Prompting), to enhance VP by introducing additional transformation operations that address its limitations. Our investigation begins with identifying the primary constraints of existing VP techniques and then proposing solutions that maintain the low computational overhead characteristic of VP methods. Figure 1 shows an overview of ACAVP. The theoretical justification for our approach is provided in Appendix B, where we formally analyze the hypothesis spaces of existing VP methods and ACAVP, demonstrating that ACAVP has a smaller approximation error, which explains its superior performance. The implementation details are provided in Appendix C.

#### 3.1 Preliminaries

VP is a parameter-efficient fine-tuning technique that adapts pre-trained recognition models to downstream tasks by applying learnable prompts to input images while keeping the model parameters frozen [Bahng et al., 2022]. During training, VP learns a prompt that minimizes the task-specific loss of the recognition model. At inference time, the learned prompt is applied to input images before model prediction. Formally, given an input image  $x \in \mathbb{R}^{3 \times H \times W}$ , traditional VP transforms

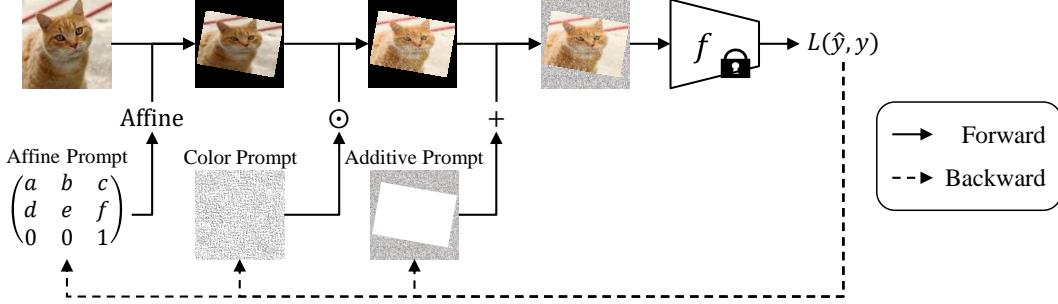


Figure 1: Overview of ACAVP. The input image undergoes three transformations: first, an affine transformation using the affine prompt, followed by a color transformation using the color prompt, and finally an additive transformation using the additive prompt. The resulting VP image is fed into a frozen recognition model, and the loss is backpropagated to update all three prompt parameters.

it into a prompted image  $\tilde{x} \in \mathbb{R}^{3 \times H \times W}$  using the following equation:  $\tilde{x} = x + M \odot \delta$ , where  $M \in \{0, 1\}^{H \times W}$  is a binary mask indicating the prompt region,  $\delta \in \mathbb{R}^{3 \times H \times W}$  is the learnable prompt parameter, and  $\odot$  denotes element-wise multiplication.

VP has advantages such as almost zero computational overhead during inference and applicability to black-box recognition models (including cloud APIs) [Oh et al., 2023, Tsai et al., 2020] in order to maintain the original model architecture and parameters. However, as our preliminary experiments show, VP suffers from two significant limitations: limited expressive power due to its reliance on simple additive transformations and a tendency to overfit when the parameter count increases.

### 3.2 Problem Analysis

Our preliminary experiments revealed two significant limitations of conventional VP approaches. First, increasing the number of VP parameters by expanding the prompt size theoretically reduces the approximation error, but does not improve the test or training accuracy. This suggests that the additive transformation alone has inherent limitations in its expressive power. Second, as the number of parameters increases, the VP model tends to overfit, reducing test accuracy. Based on these observations, we hypothesized that: (1) introducing different types of transformations beyond addition could enhance the expressive power of VP, and (2) introducing effective regularization strategies could mitigate overfitting issues.

### 3.3 ACAVP Design

To address these limitations, we propose ACAVP (Affine, Color, and Additive Visual Prompting), an enhanced VP method incorporating affine transformation and color transformation into the VP design, accompanied by effective regularization through data augmentation techniques. Our proposed method transforms the input image according to:

$$\tilde{x} = \text{Affine}(x, A) \odot \hat{\sigma} + M \odot \delta \quad (1)$$

where  $\text{Affine}(\cdot, A)$  represents the affine transformation with learnable affine prompt  $A \in \mathbb{R}^{3 \times 3}$ ,  $\hat{\sigma} \in \mathbb{R}^{3 \times H \times W}$  is the learnable color prompt, and  $\delta \in \mathbb{R}^{3 \times H \times W}$  is the learnable additive prompt. Unlike conventional VP methods,  $M \in \{0, 1\}^{H \times W}$  is dynamically generated after the affine transformation is applied, where  $M$  has value 1 for pixels where all three channels are zero, and 0 elsewhere. This ensures the additive prompt is only applied to the empty regions that the affine transformation creates. This formulation extends the transformation space beyond simple addition, enabling more expressive prompts while maintaining the computational efficiency of VP.

**The affine transformation prompt** generates task-specific feature regions while preserving the original image information. The affine transformation enables geometric modifications, including rotation, translation, shearing, and scaling, providing a richer set of transformations than simple addition. For an input point  $(u, v)$ , the affine transformation with transformation matrix  $A$  computes its new coordinates  $(\hat{u}, \hat{v})$  as follows:

$$\begin{bmatrix} \hat{u} \\ \hat{v} \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{s}_x \cos \hat{\theta} + \hat{s}h_x \sin \hat{\theta} & -\hat{s}_x \sin \hat{\theta} + \hat{s}h_x \cos \hat{\theta} & \hat{t}_x \\ \hat{s}_y \sin \hat{\theta} + \hat{s}h_y \cos \hat{\theta} & \hat{s}_y \cos \hat{\theta} + \hat{s}h_y \sin \hat{\theta} & \hat{t}_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2)$$

To ensure stable training and prevent extreme transformations that might destroy the original image information, we constrain the transformation parameters using activation functions:  $\hat{t} = \tanh(t) \cdot R_t$ ,  $\hat{\theta} = \tanh(\theta) \cdot R_\theta$ ,  $\hat{s} = \text{sigmoid}(s)$ , and  $\hat{sh} = \tanh(sh) \cdot R_{sh}$ . Here,  $t$  represents learnable translation parameters,  $\theta$  is a learnable rotation parameter,  $s$  refers to learnable scaling parameters, and  $sh$  denotes learnable shearing parameters. The hyperparameters  $R_t$ ,  $R_\theta$ , and  $R_{sh}$  control the maximum range of each transformation. The default values for these hyperparameters are set to  $R_t = 0.05$ ,  $R_\theta = R_{sh} = 0.1$ .

**The color transformation prompt** further enhances the expressive power of our VP by adjusting image properties such as brightness and contrast. This is achieved through a learnable parameter  $\sigma \in \mathbb{R}^{3 \times H \times W}$  applied as a multiplicative factor to the transformed image. The color transformation allows the prompt to emphasize or attenuate specific features in the original image, potentially highlighting task-relevant information.

To prevent excessive color transformation that might corrupt the original image information, we constrain the values using:  $\hat{\sigma} = \text{sigmoid}(\sigma) \cdot R_\sigma$ , where  $R_\sigma$  is a hyperparameter controlling the maximum magnitude of color transformation, with a default value of  $R_\sigma = 6.0$ . This ensures that the color transformations remain within a reasonable range while still providing sufficient flexibility to enhance task-relevant features.

### 3.4 Overfitting Mitigation

As shown in our preliminary experiments, increasing the number of VP parameters theoretically reduces the approximation error, but causes overfitting. This observation led us to hypothesize that effective regularization strategies would be crucial for our enhanced ACAVP approach with its additional transformation parameters. To identify the most effective overfitting mitigation technique, we comprehensively evaluated various regularization approaches applied to VP training. Specifically, we investigated four categories of techniques: data augmentation, dropout applied after feature extraction layer, MSE loss between original and prompted images as an additional regularization loss term, and weight decay. Our experimental results, detailed in Section 4.4, revealed that data augmentation methods consistently provided the most substantial improvements in generalization performance. Among the various augmentation techniques evaluated, TrivialAugment emerged as particularly effective for VP training. TrivialAugment applies a randomly selected augmentation operation with random magnitude to each input image, providing sufficient diversity to mitigate overfitting while maintaining implementation simplicity. While previous VP approaches have employed basic augmentations like random crop and horizontal flip, we found that more sophisticated augmentation strategies could significantly improve the performance of VP.

## 4 Experiments

To evaluate the effectiveness of ACAVP, we conducted extensive experiments across 12 diverse image classification datasets using two different model architectures. We compared ACAVP against zero-shot inference (ZS) with CLIP [Radford et al., 2021] and state-of-the-art VP methods (VP [Bahng et al., 2022], EVP [Wu et al., 2024], and AutoVP [Tsao et al., 2024]). All experiments were repeated three times, and we report the average accuracy and standard error. Each dataset was split into training, validation, and test sets, and we used the checkpoint with the highest accuracy on the validation set for final testing. Detailed experimental settings are provided in Appendix D.

### 4.1 Main Results

We evaluated ACAVP using a ViT-B32-based CLIP model [Radford et al., 2021, Dosovitskiy et al., 2021] across 12 diverse image classification datasets. Table 2 shows the comparative performance results. ACAVP demonstrates superior accuracy on most datasets compared to existing VP methods, achieving an average accuracy of 83.18%, which significantly outperforms VP, EVP, and AutoVP. The

Table 2: Classification accuracy comparison on 12 image classification datasets using CLIP ViT-B/32. **Bold** and underlined values denote the first and second highest accuracy, respectively. The LP and FT results refer to Bahng et al. [2022], Tsao et al. [2024], Tang et al. [2024].

Method	ZS	VP	EVP	AutoVP	ACAVP	LP	FT
CIFAR10	88.93	<u>93.65</u> $\pm$ 0.04	<u>95.87</u> $\pm$ 0.06	95.39 $\pm$ 0.06	<b>96.27</b> $\pm$ 0.07	95.00	95.80
CIFAR100	63.09	74.71 $\pm$ 0.09	78.33 $\pm$ 0.09	<u>79.89</u> $\pm$ 0.04	<b>80.06</b> $\pm$ 0.25	80.00	82.10
CLEVR	20.70	76.40 $\pm$ 0.56	76.82 $\pm$ 0.27	<b>79.19</b> $\pm$ 0.29	77.19 $\pm$ 0.17	66.00	94.40
DTD	42.91	57.19 $\pm$ 0.14	60.89 $\pm$ 0.57	<u>62.81</u> $\pm$ 0.12	<b>67.43</b> $\pm$ 0.83	74.60	72.30
EuroSAT	39.02	<u>96.36</u> $\pm$ 0.07	97.27 $\pm$ 0.08	<u>97.32</u> $\pm$ 0.11	<b>97.51</b> $\pm$ 0.09	95.30	97.90
Flowers	61.63	67.83 $\pm$ 0.47	77.76 $\pm$ 0.59	<u>82.69</u> $\pm$ 0.80	<b>89.36</b> $\pm$ 0.19	96.90	97.40
Food	<b>79.67</b>	<u>78.92</u> $\pm$ 0.03	77.48 $\pm$ 0.00	76.53 $\pm$ 0.03	77.86 $\pm$ 0.09	84.60	80.50
GTSRB	18.18	91.38 $\pm$ 0.28	76.17 $\pm$ 2.40	<b>92.50</b> $\pm$ 0.14	<u>92.04</u> $\pm$ 0.38	85.80	98.90
Pets	85.96	86.51 $\pm$ 0.30	<u>87.85</u> $\pm$ 0.06	86.71 $\pm$ 0.26	<b>87.55</b> $\pm$ 0.02	89.20	88.50
SUN	60.55	60.39 $\pm$ 0.12	<u>63.51</u> $\pm$ 0.03	61.66 $\pm$ 0.17	<b>64.95</b> $\pm$ 0.10	75.00	64.00
SVHN	15.30	91.61 $\pm$ 0.12	<u>86.69</u> $\pm$ 0.09	<b>92.94</b> $\pm$ 0.18	<u>91.81</u> $\pm$ 0.18	65.40	95.70
UCF	61.46	65.68 $\pm$ 0.28	67.69 $\pm$ 0.52	<u>70.88</u> $\pm$ 0.11	<b>76.11</b> $\pm$ 0.38	83.30	80.90
Average	53.12	78.39	78.86	81.54	<b>83.18</b>	82.59	87.37

consistent improvements across diverse datasets highlight the enhanced generalization capability of ACAVP. Notably, ACAVP shows remarkable improvement on the Flowers dataset, with absolute gains of 6.67 percentage points over AutoVP and 21.53 percentage points over the VP. This substantial improvement suggests that our affine transformation and color transformation components are particularly effective for datasets with fine-grained visual features, such as the intricate textures and shapes found in flower images. The ability to capture such nuanced visual characteristics demonstrates the enhanced expressive power of ACAVP. Interestingly, ACAVP surpasses the average accuracy of Linear Probing (LP), which modifies the model parameters. VP methods, including ours, demonstrate particularly strong performance on datasets that differ significantly from the model’s pretraining distribution. For instance, on specialized datasets such as GTSRB (traffic signs) and SVHN (street view house numbers), VP approaches substantially outperform LP. This pattern aligns with previous findings that VP is especially effective for adapting to domains that deviate from the source distribution used during pretraining [Bahng et al., 2022]. In summary, our experimental results demonstrate that incorporating affine transformation and color transformation alongside overfitting mitigation techniques effectively addresses the limitations of conventional VP approaches. These results validate our hypothesis that diversifying the transformation space and mitigating overfitting are crucial factors for enhancing the effectiveness of VP techniques.

We further evaluated ACAVP using ResNet50 [He et al., 2016] pre-trained on the ImageNet-1k dataset [Russakovsky et al., 2015] across the same 12 image classification datasets. This evaluation aims to demonstrate the generalizability of our approach across different model architectures and pretraining strategies. Table 3 shows the results. Similar to our findings with the CLIP model, ACAVP outperforms baseline approaches on many datasets and achieves the highest average accuracy among all VP methods. Specifically, without label mapping optimization (left columns), ACAVP achieves the highest average accuracy, significantly outperforming the second-best method by 15.01 percentage points and 21.50 percentage points on the Flowers and GTSRB datasets, respectively. Similarly, with label mapping (denoted by  $\dagger$  in the right columns), ACAVP achieves the highest average accuracy. While all methods benefit from label mapping, our approach maintains its superior performance, suggesting that the benefits of our enhanced transformation space are complementary to those provided by label mapping. The consistent relative performance across both settings indicates that ACAVP’s improvements stem from fundamental enhancements to the VP framework rather than interactions with specific implementation details. These results demonstrate that ACAVP generalizes effectively across different model architectures and pretraining strategies. The consistent performance improvements observed with both CLIP and ResNet50 backbones suggest that the limitations of conventional VP approaches are fundamental rather than model-specific and that our enhancements to the transformation space and learning methodology principledly address these limitations.

Table 3: Classification accuracy comparison on 12 image classification datasets using ResNet50 pre-trained on ImageNet-1k.  $\dagger$  indicates results with label mapping proposed by Wu et al. [2024].

Method	VP	EVP	AutoVP	ACAVP	VP $\dagger$	EVP $\dagger$	AutoVP $\dagger$	ACAVP $\dagger$
CIFAR10	55.09 $\pm$ 0.25	54.28 $\pm$ 0.20	45.52 $\pm$ 0.34	53.48 $\pm$ 0.61	64.09 $\pm$ 0.08	67.93 $\pm$ 0.26	62.77 $\pm$ 0.35	72.92 $\pm$ 0.91
CIFAR100	10.95 $\pm$ 0.15	7.40 $\pm$ 0.26	6.80 $\pm$ 0.36	14.84 $\pm$ 0.08	27.69 $\pm$ 0.11	32.30 $\pm$ 0.29	29.44 $\pm$ 0.39	33.71 $\pm$ 0.56
CLEVR	42.19 $\pm$ 0.19	40.62 $\pm$ 0.47	28.90 $\pm$ 0.49	32.23 $\pm$ 0.06	40.23 $\pm$ 0.34	37.88 $\pm$ 0.24	29.16 $\pm$ 0.41	38.95 $\pm$ 0.49
DTD	13.46 $\pm$ 0.36	15.98 $\pm$ 0.61	12.59 $\pm$ 0.24	9.34 $\pm$ 2.85	30.65 $\pm$ 0.15	35.62 $\pm$ 0.52	30.04 $\pm$ 0.26	34.16 $\pm$ 0.77
EuroSAT	79.06 $\pm$ 0.08	78.44 $\pm$ 0.11	68.79 $\pm$ 0.64	76.70 $\pm$ 1.16	82.18 $\pm$ 0.20	79.94 $\pm$ 0.14	71.82 $\pm$ 0.99	83.04 $\pm$ 0.23
Flowers	8.40 $\pm$ 0.35	10.37 $\pm$ 0.43	10.22 $\pm$ 0.20	25.38 $\pm$ 0.63	12.13 $\pm$ 0.06	16.59 $\pm$ 0.65	14.60 $\pm$ 0.24	19.77 $\pm$ 0.62
Food	4.22 $\pm$ 0.17	3.24 $\pm$ 0.01	2.82 $\pm$ 0.04	6.69 $\pm$ 0.05	7.66 $\pm$ 0.09	7.14 $\pm$ 0.19	7.18 $\pm$ 0.09	8.55 $\pm$ 0.55
GTSRB	38.87 $\pm$ 0.43	30.26 $\pm$ 0.60	28.17 $\pm$ 0.64	60.37 $\pm$ 0.73	45.89 $\pm$ 0.10	34.93 $\pm$ 0.39	37.66 $\pm$ 0.52	45.77 $\pm$ 0.05
Pets	5.76 $\pm$ 0.11	7.47 $\pm$ 0.18	6.92 $\pm$ 0.14	12.23 $\pm$ 0.25	64.80 $\pm$ 0.32	68.66 $\pm$ 0.08	66.68 $\pm$ 0.07	68.23 $\pm$ 0.27
SUN	0.87 $\pm$ 0.02	1.10 $\pm$ 0.08	0.60 $\pm$ 0.07	1.50 $\pm$ 0.03	7.93 $\pm$ 0.03	8.73 $\pm$ 0.16	7.74 $\pm$ 0.08	7.33 $\pm$ 0.16
SVHN	66.54 $\pm$ 0.32	48.21 $\pm$ 0.30	46.60 $\pm$ 1.51	62.21 $\pm$ 0.48	67.84 $\pm$ 0.21	49.17 $\pm$ 0.47	45.33 $\pm$ 1.24	68.82 $\pm$ 0.44
UCF	8.19 $\pm$ 0.33	7.28 $\pm$ 0.66	6.53 $\pm$ 0.10	13.11 $\pm$ 0.19	24.50 $\pm$ 0.21	27.56 $\pm$ 0.07	26.14 $\pm$ 0.23	28.14 $\pm$ 0.75
Average	27.80	25.39	22.04	30.67	39.63	38.87	35.71	42.45

Table 4: Classification accuracy on the corrupted datasets using CLIP ViT-B/32. Classification accuracy means the average accuracy at 15 different types and 5 levels of severity.

Method	CIFAR10-C	CIFAR100-C
ZS	70.9	42.59
VP	76.65 $\pm$ 0.04	50.52 $\pm$ 0.12
EVP	80.97 $\pm$ 0.14	55.17 $\pm$ 0.16
AutoVP	79.72 $\pm$ 0.28	56.34 $\pm$ 0.15
ACAVP	83.98 $\pm$ 0.20	58.68 $\pm$ 0.29

Table 5: An ablation study shows the contribution of different transformation components. Affine refers to affine transformation, Additive to additive transformation, and Color to color transformation. All methods use TrivialAugment for fair comparison.

Transformation	CIFAR10	Flowers
–	88.93	61.63
Affine	90.17 $\pm$ 0.01	57.42 $\pm$ 0.03
Color	92.14 $\pm$ 0.06	62.92 $\pm$ 0.04
Affine + Color	92.46 $\pm$ 0.04	60.58 $\pm$ 0.08
Resize + Additive	96.00 $\pm$ 0.05	79.24 $\pm$ 0.18
Resize + Additive + Color	96.05 $\pm$ 0.04	80.74 $\pm$ 0.28
Affine + Additive	96.31 $\pm$ 0.02	88.70 $\pm$ 0.18
Affine + Color + Additive	96.27 $\pm$ 0.07	89.36 $\pm$ 0.19

## 4.2 Robustness to Distribution Shift

We evaluated the robustness of ACAVP to distribution shifts by comparing its performance against baseline approaches on artificially corrupted datasets. Distribution robustness is a critical aspect of adaptation methods in real-world scenarios where test data often differs from training data due to various forms of noise and corruption. For this evaluation, we tested various VP approaches on the CIFAR10-C and CIFAR100-C datasets, which apply 15 common types of image corruptions (including noise, blur, weather, and digital distortions) at different severity levels to the original CIFAR10 and CIFAR100 test sets, respectively [Hendrycks and Dietterich, 2019]. These standardized corruption benchmarks provide a systematic way to assess model robustness under various distribution shifts. The results are shown in Table 4. ACAVP consistently outperforms all baseline approaches by significant margins on both corrupted datasets. Notably, the performance improvement of ACAVP is more pronounced on out-of-distribution data compared to in-distribution performance (refer to Table 2). For example, on the CIFAR10-C dataset, ACAVP achieves a 7.33 percentage points improvement over VP, which is higher than the 2.62 percentage points improvement observed on the original CIFAR10 dataset. This enhanced robustness to distribution shifts suggests that the diverse transformation space introduced by ACAVP, combined with overfitting mitigation techniques, enables learning more generalizable transformations. These results highlight an important practical advantage of ACAVP: improved robustness to real-world distribution shifts, which is often crucial for the deployment of adaptation methods in dynamic and unpredictable environments.

## 4.3 Ablation Study

We conducted an ablation study to analyze the contribution of individual components in ACAVP on the CIFAR10 and Flowers datasets. This analysis helps identify which transformations are most effective and how they interact with each other. Using ZS (no transformation) and EVP (which

Table 6: Comparison of overfitting mitigation techniques on image-sized VP with CLIP ViT-B/32. Values in parentheses show accuracy gains from using overfitting mitigation techniques.

Method	CIFAR10	Flowers
Dropout	$88.50 \pm 0.18$ (-1.02)	$53.57 \pm 0.41$ (-0.28)
MSE Loss	$90.26 \pm 0.02$ (+0.74)	$53.74 \pm 0.34$ (-0.11)
Weight decay	$91.25 \pm 0.10$ (+1.73)	$55.37 \pm 0.58$ (+1.52)
TrivialAugment	$91.56 \pm 0.06$ (+2.04)	$70.11 \pm 0.61$ (+16.26)
IPMix	$90.85 \pm 0.07$ (+1.33)	$66.03 \pm 0.80$ (+12.18)
RandAugment	$91.52 \pm 0.05$ (+2.00)	$72.88 \pm 0.70$ (+19.03)

applies an additive transformation after resizing) as the baselines, we evaluated six variants: affine transformation alone (Affine), color transformation alone (Color), affine and color transformations (Affine + Color), EVP with color transformation (Resize + Additive + Color), EVP with affine transformation replacing the resize operation (Affine + Additive), and ACAVP that combines affine transformation, color transformation, and additive transformation (Affine + Color + Additive). To ensure a fair comparison, we incorporated TrivialAugment in the training process for all methods, including the EVP baseline. The results are shown in Table 5. Individual transformations show varying effectiveness across datasets: the affine transformation improves performance on the CIFAR10 dataset but shows reduced performance on the Flowers dataset compared to ZS, while the color transformation consistently outperforms ZS accuracy on both datasets. Combining affine and color transformations yields further improvements. On the Flowers dataset, which contains rich textural and color information, the performance gains are particularly illuminating. The variant with color transformation (Resize + Additive + Color) outperforms EVP by 1.50 percentage points, while the variant with affine transformation (Affine + Additive) exceeds EVP by 9.46 percentage points. These results demonstrate the significant contributions of both color transformation and affine transformation to accuracy, with affine transformation having a more substantial impact on the Flowers dataset. Notably, our full method, which incorporates both affine and color transformations, achieves the highest accuracy, surpassing EVP by 10.12 percentage points on the Flowers dataset. This suggests that these transformations are complementary rather than competing, and their combination leads to more effective adaptation. The affine transformation likely helps in focusing on the most discriminative regions of flowers, while the color transformation enhances the representation of subtle color variations that are critical for flower classification. On the CIFAR10 dataset, all variants show improvements over ZS, though the differences are less pronounced than on the Flowers dataset. This smaller gap is expected since the CIFAR10 features less complex visual patterns and color variations than the Flowers. Nevertheless, the improvements across both datasets confirm that each component contributes positively to the overall performance of our method. This ablation study validates our design choices and demonstrates that both the affine and color transformation components contribute meaningfully to improving VP performance.

#### 4.4 Effectiveness of Overfitting Mitigation Strategies

We demonstrate experimentally that mitigating overfitting plays a crucial role in VP training. Our preliminary experiments showed that VP approaches, particularly those using image-sized noise additions, exhibit strong overfitting tendencies that limit performance. Here, we investigate various strategies to address this limitation. We first experimented applying various overfitting mitigation techniques to the image-sized VP approach, which showed overfitting tendencies in our preliminary studies. The techniques we evaluated include: dropout layer after the feature extraction layer, MSE loss minimization between the original and prompted images, weight decay for parameter regularization, and three data augmentation methods, TrivialAugment, RandAugment [Cubuk et al., 2020], and IPMix [Huang et al., 2023], which increase input diversity. The results are shown in Table 6. Dropout and MSE show no significant improvement over the baseline. Weight decay demonstrates a modest accuracy improvement. Most notably, data augmentation techniques yield substantial performance gains, with TrivialAugment and RandAugment showing powerful improvements on both datasets. These results strongly suggest that overfitting mitigation is critical for VP training, with data augmentations being especially effective.

Next, we investigated whether TrivialAugment, the effective and computationally efficient regularization technique from our first experiment, could improve the performance of existing VP baselines.

Table 7: Effect of TrivialAugment on existing VP methods with CLIP ViT-B/32. Values in parentheses show accuracy gains from adding TrivialAugment.

Method	CIFAR10	Flowers
VP	$94.97 \pm 0.06$ (+1.32)	$79.84 \pm 0.07$ (+12.01)
EVP	$96.00 \pm 0.05$ (+0.13)	$79.24 \pm 0.18$ (+1.48)
AutoVP	$96.16 \pm 0.02$ (+0.77)	$85.83 \pm 0.58$ (+3.14)

The results, shown in Table 7, demonstrate consistent accuracy improvements across all VP baselines when TrivialAugment is applied. The improvements are particularly pronounced for the standard VP approach on the Flowers dataset, with a substantial 12.01 percentage point accuracy gain. Interestingly, with TrivialAugment, the standard VP achieves accuracy comparable to EVP, despite EVP’s additional resize operation. This suggests that the resize operation in EVP may have been functioning primarily as an implicit regularizer by preserving more of the original image information and limiting overfitting. However, this advantage diminishes with explicit input diversity through TrivialAugment, indicating that VP and EVP may have similar expressive power when properly regularized. These findings strongly support our approach of diversifying the transformation space while addressing overfitting. They suggest that the performance limitations of conventional VP methods stem not only from restricted transformation operations but also from their susceptibility to overfitting. By combining a more expressive transformation space with effective regularization techniques, ACAVP addresses both limitations simultaneously, leading to substantial performance improvements across diverse datasets.

#### 4.5 Computational Efficiency Analysis

Table 8: Inference time comparison across different methods using H100 GPU with batch size 500. Relative Time is normalized to the CLIP ViT-B/32’s inference time.

Method	Time (s)	Relative Time
VP	$2.88 \times 10^{-5}$	0.00
EVP	$1.38 \times 10^{-4}$	0.00
AutoVP	$1.77 \times 10^{-2}$	0.60
ACAVP	$1.63 \times 10^{-3}$	0.06
Coordinator	$2.01 \times 10^{-1}$	6.85
Model Inference	$2.93 \times 10^{-2}$	1.00

Table 8. As expected, ACAVP requires more computation time than VP and EVP. However, when compared to the model’s inference time, ACAVP adds only 6% of the model’s computational cost, which is negligible in practical applications. Notably, the Coordinator has a longer inference time than the recognition model inference. This substantial overhead undermines one of the key benefits of VP approaches: their minimal impact on inference efficiency. Our approach strikes an optimal balance between computational efficiency and transformation expressivity. Using linear transformations (affine and color) rather than complex neural networks, we achieve significant improvements in accuracy with only a modest increase in computational cost. This makes ACAVP particularly attractive for real-world applications where performance and efficiency are important considerations. These results highlight an important design principle for VP approaches: while enhancing the transformation space is crucial for improving accuracy, the choice of transformation operations should consider their computational implications. Linear transformations like those used in ACAVP offer a favorable trade-off: They provide substantial expressive power with minimal computational overhead.

### 5 Conclusion

In this paper, we addressed the limitations of conventional visual prompting (VP) techniques by introducing ACAVP, a novel framework that enhances VP’s expressive power and mitigates over-

fitting issues. Our approach builds upon the existing VP paradigm by incorporating two additional transformation operations: affine transformation, which creates task-specific prompt regions while preserving original image information, and color transformation, which emphasizes task-relevant features through brightness and contrast adjustments. Furthermore, we demonstrated that data augmentation techniques, particularly TrivialAugment, significantly improve generalization performance for VP methods by mitigating overfitting tendencies. Our extensive experimental evaluation across 12 diverse image classification datasets using both CLIP ViT and ResNet50 architectures validates the effectiveness of ACAVP. ACAVP consistently outperforms existing VP methods and achieves higher average accuracy than linear probing, which modifies model parameters. Notably, our method exhibits superior robustness to distribution shifts, as demonstrated by its performance on corrupted datasets. These improvements are achieved while maintaining VP’s key advantage of minimal computational overhead during inference, with ACAVP adding only 6% to the model’s inference time. The success of ACAVP highlights the importance of transformation expressivity and regularization in VP training.

## References

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, 2014.

Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Bayesian-guided label mapping for visual reprogramming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Sample-specific masks for visual reprogramming-based prompting. In *International Conference on Machine Learning (ICML)*, 2024b.

Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Attribute-based visual reprogramming for image classification with clip. In *International Conference on Learning Representations (ICLR)*, 2025.

Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and improving visual prompting: A label-mapping perspective. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

Wonwoo Cho, Kangyeol Kim, Saemee Choi, and Jaegul Choo. Training spatial-frequency visual prompts and probabilistic clusters for accurate black-box transfer learning. In *ACM International Conference on Multimedia (ACMMM)*, 2024.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 2015.

Yulu Gan, Xianzheng Ma, Yihang Lou, Yan Bai, Renrui Zhang, Nian Shi, and Lin Luo. Decorate the newcomers: Visual domain prompt for continual test time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Cheng Han, Qifan Wang, Yiming Cui, Zhiwen Cao, Wenguan Wang, Siyuan Qi, and Dongfang Liu. E 2 vpt: An effective and efficient approach for visual prompt tuning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Cheng Han, Qifan Wang, Yiming Cui, Wenguan Wang, Lifu Huang, Siyuan Qi, and Dongfang Liu. Facing the elephant in the room: Visual prompt tuning or full finetuning? In *International Conference on Learning Representations (ICLR)*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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*, 2019.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks (IJCNN)*, 2013.

Zhenglin Huang, Xiaoan Bao, Na Zhang, Qingqi Zhang, Xiao Tu, Biao Wu, and Xi Yang. Ipmix: Label-preserving data augmentation method for training robust classifiers. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, 2022.

Can Jin, Ying Li, Mingyu Zhao, Shiyu Zhao, Zhenting Wang, Xiaoxiao He, Ligong Han, Tong Che, and Dimitris N. Metaxas. Lor-VP: Low-rank visual prompting for efficient vision model adaptation. In *International Conference on Learning Representations (ICLR)*, 2025.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Michael J Kearns and Umesh Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

Dongxu Li, Junnan Li, and Steven Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016.

Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, 2011.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, 2008.

Changdae Oh, Hyeji Hwang, Hee-young Lee, YongTaek Lim, Geunyoung Jung, Jiyoung Jung, Hosik Choi, and Kyungwoo Song. Blackvip: Black-box visual prompting for robust transfer learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Maxime Oquab, Timothée Darzet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)*, 2015.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision (IJCV)*, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Anke Tang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. FusionBench: A Comprehensive Benchmark of Deep Model Fusion, 2024.

Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *International Conference on Machine Learning (ICML)*, 2020.

Yun-Yun Tsai, Chengzhi Mao, and Junfeng Yang. Convolutional visual prompt for robust visual perception. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Hsi-Ai Tsao, Lei Hsiung, Pin-Yu Chen, Sijia Liu, and Tsung-Yi Ho. AutoVP: An Automated Visual Prompting Framework and Benchmark. In *International Conference on Learning Representations (ICLR)*, 2024.

Junyang Wu, Xianhang Li, Chen Wei, Huiyu Wang, Alan Yuille, Yuyin Zhou, and Cihang Xie. Unleashing the power of visual prompting at the pixel level. *Transactions on Machine Learning Research (TMLR)*, 2024.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Runjia Zeng, Cheng Han, Qifan Wang, Chunshu Wu, Tong Geng, Lifu Huangg, Ying Nian Wu, and Dongfang Liu. Visual fourier prompt tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Yichi Zhang, Yinpeng Dong, Siyuan Zhang, Tianzan Min, Hang Su, and Jun Zhu. Exploring the transferability of visual prompting for multimodal large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022b.

## A Related Work

### A.1 Applicability of VP.

VP has been successfully applied to black-box scenarios where model internals are inaccessible. Tsai et al. [2020] proposed black-box adversarial reprogramming (BAR), which leverages zeroth-order optimization, using only numerical evaluations of the loss function instead of gradients. Oh et al. [2023] further advanced this direction with BlackVIP, enabling adaptation without knowledge of the recognition model’s architecture or parameters. VP has shown promise in adapting models at test time to handle distribution shifts. Gan et al. [2023] applied VP to test-time adaptation, improving inference accuracy on out-of-distribution data by updating prompts during test time without labels. Tsai et al. [2023] introduced convolutional visual prompt (CVP), which are pre-trained through self-supervised learning and adapted at test time to enhance out-of-distribution performance.

### A.2 Visual Prompt Tuning.

Visual prompt tuning (VPT) [Jia et al., 2022, Han et al., 2024, 2023, Zeng et al., 2024] is conceptually related to visual prompting. VPT adapts to downstream tasks by optimizing input embeddings as well as the classification head of the model. In contrast, VP methods attach learnable parameters directly to the input image, and typically adjust the classification head using heuristic methods [Wu et al., 2024, Chen et al., 2023, Cai et al., 2024a], rather than end-to-end optimization.

Importantly, VP methods are often designed for settings where the architecture of the recognition model is unknown or the gradients are inaccessible, as in black-box scenarios [Tsai et al., 2020, Oh et al., 2023].

## B Theoretical Analysis

In this section, we provide a theoretical analysis of ACAVP within the framework of probably approximately correct (PAC) learning [Kearns and Vazirani, 1994]. We establish that our proposed method has a smaller approximation error compared to existing visual prompting approaches, which explains its superior empirical performance.

Our analysis draws inspiration from the theoretical framework established by Cai et al. [2024b] for analyzing Sample-specific Masks for Visual Reprogramming (SMM). Cai et al. [2024b] demonstrated that using sample-specific masks reduces the approximation error compared to shared masks in visual reprogramming. We extend this insight to analyze the relationship between different visual prompting methods.

According to Cai et al. [2024b], the approximation error measures how close the best achievable error by a hypothesis space is to the theoretical minimum error on a distribution. When hypothesis spaces exhibit a subset relation, the larger hypothesis space has a smaller approximation error.

**Theorem B.1** (Cai et al. [2024b]). *Given an input space  $X$ , a discrete label space  $Y$ , and a distribution  $D$  over  $X \times Y$ , if there are two hypothesis spaces  $F_1 \subseteq \{f : X \rightarrow Y\}$  and  $F_2 \subseteq \{f : X \rightarrow Y\}$  satisfying  $F_1 \subseteq F_2$ , then we have*

$$Err_D^{apx}(F_1) \geq Err_D^{apx}(F_2). \quad (3)$$

### B.1 Hypothesis Spaces of Visual Prompting Methods

We now formalize the hypothesis spaces of the three visual prompting methods under consideration. For a fixed pre-trained model  $f_P$  and fixed output mapping function  $f_{out}$  (where  $f'_P = f_{out} \circ f_P$ ), we define:

**EVP Hypothesis Space.** Enhanced Visual Prompting (EVP) resizes the input image by a fixed factor  $s$  and adds a learned pattern  $\delta$ . Since the resize factor is predetermined, a fixed mask  $M$  can be prepared in advance. The hypothesis space of EVP is defined as:

$$F^{evp}(f'_P) = \{f | f(x) = f'_P(r_s(x) + M \odot \delta), \forall x \in X\}, \quad (4)$$

where  $r_s(\cdot)$  is a resizing operation with fixed scale factor  $s$ ,  $M$  is the predetermined mask, and  $\delta$  is the learnable pattern.

**AutoVP Hypothesis Space.** AutoVP extends EVP by making the resizing factor learnable. Since the image size changes during training, the mask is dynamically created according to the resize parameter. Its hypothesis space is defined as:

$$F^{\text{autovp}}(f'_P) = \{f \mid f(x) = f'_P(r_{\hat{s}}(x) + M \odot \delta), \forall x \in X\}, \quad (5)$$

where  $\hat{s}$  is a learnable scaling parameter,  $M$  is the mask dynamically generated based on  $\hat{s}$ , and  $\delta$  is the learnable pattern.

**ACAVP Hypothesis Space.** Our proposed ACAVP incorporates affine and color transformation. Its hypothesis space is defined as:

$$F^{\text{acavp}}(f'_P) = \{f \mid f(x) = f'_P(\text{Affine}(x, A) \odot \hat{\sigma} + M \odot \delta), \forall x \in X\}, \quad (6)$$

where  $\text{Affine}(\cdot, A)$  represents the affine transformation with learnable affine prompt  $A \in \mathbb{R}^{3 \times 3}$ ,  $\hat{\sigma} \in \mathbb{R}^{3 \times H \times W}$  is the learnable color prompt, and  $\delta$  is the learnable additive prompt.

## B.2 Relationship Between Hypothesis Spaces

Following the approach in Cai et al. [2024b], we can establish the following relationship between these hypothesis spaces:

**Proposition B.2.** *For any fixed pre-trained model  $f_P$  and fixed output mapping function  $f_{\text{out}}$ , we have*

$$F^{\text{evp}}(f'_P) \subseteq F^{\text{autovp}}(f'_P) \subseteq F^{\text{acavp}}(f'_P). \quad (7)$$

*Proof.* First, we show that  $F^{\text{evp}}(f'_P) \subseteq F^{\text{autovp}}(f'_P)$ . For any function  $f \in F^{\text{evp}}(f'_P)$ , we have

$$f(x) = f'_P(r_s(x) + M \odot \delta) \quad (8)$$

for some fixed scale factor  $s$  and learnable pattern  $\delta$ . We can find a corresponding function in  $F^{\text{autovp}}(f'_P)$  by setting the learnable parameter  $\hat{s} = s$ , resulting in

$$f(x) = f'_P(r_{\hat{s}}(x) + M \odot \delta) \quad (9)$$

with the same  $\delta$ . Therefore,  $f \in F^{\text{autovp}}(f'_P)$ , which means  $F^{\text{evp}}(f'_P) \subseteq F^{\text{autovp}}(f'_P)$ .

Next, we show that  $F^{\text{autovp}}(f'_P) \subseteq F^{\text{acavp}}(f'_P)$ . For any function  $f \in F^{\text{autovp}}(f'_P)$ , we have

$$f(x) = f'_P(r_{\hat{s}}(x) + M \odot \delta) \quad (10)$$

for some learnable parameter  $\hat{s}$  and pattern  $\delta$ . We can construct a corresponding function in  $F^{\text{acavp}}(f'_P)$  by:

- Setting the affine transformation matrix  $A = \begin{bmatrix} \hat{s} & 0 & 0 \\ 0 & \hat{s} & 0 \\ 0 & 0 & 1 \end{bmatrix}$ , which performs only scaling by  $\hat{s}$
- Setting the color transformation prompt  $\hat{\sigma} = \mathbf{1}$  (all ones)
- Using the same additive prompt  $\delta$

With these settings, we have:

$$\text{Affine}(x, A) = r_{\hat{s}}(x) \quad (11)$$

$$\text{Affine}(x, A) \odot \hat{\sigma} = r_{\hat{s}}(x) \odot \mathbf{1} = r_{\hat{s}}(x) \quad (12)$$

Therefore,

$$f(x) = f'_P(\text{Affine}(x, A) \odot \hat{\sigma} + M \odot \delta) = f'_P(r_{\hat{s}}(x) + M \odot \delta) \quad (13)$$

This shows that  $f \in F^{\text{acavp}}(f'_P)$ , which means  $F^{\text{autovp}}(f'_P) \subseteq F^{\text{acavp}}(f'_P)$ .  $\square$

Based on Theorem B.1 and Proposition B.2, we can establish our main theoretical result:

**Proposition B.3.** For any fixed pre-trained model  $f_P$ , fixed output mapping function  $f_{out}$ , and distribution  $D_T$  of the target task, we have

$$Err_{D_T}^{apx}(F^{evp}(f'_P)) \geq Err_{D_T}^{apx}(F^{autovp}(f'_P)) \geq Err_{D_T}^{apx}(F^{acavp}(f'_P)). \quad (14)$$

This theoretical result, following the framework proposed by Cai et al. [2024b], explains why ACAVP achieves superior performance compared to EVP and AutoVP. By expanding the hypothesis space through the introduction of affine and color transformation, ACAVP can more effectively approximate the optimal function for the target task.

Similar to the observation in Cai et al. [2024b] regarding estimation error, we note that while a larger hypothesis space typically reduces approximation error, it may increase estimation error due to the added complexity. However, our experimental results show that this potential increase in estimation error is well-managed through our training approach, particularly through the use of TrivialAugment for overfitting mitigation.

## C Implementation Details

In practice, our ACAVP method applies the transformations in the following order: first, the affine transformation is applied to the input image; then, the color transformation is applied through element-wise multiplication; finally, the additive noise prompt is applied in the designated regions specified by the mask  $M$ .

For initialization, we set the affine transformation parameters to perform a resizing operation, inspired by the success of a previous study [Wu et al., 2024]. Specifically, we initialize the parameters to resize a  $224 \times 224$  image to  $164 \times 164$ , setting  $\hat{s} = 0.73$  (which corresponds to  $s = \text{sigmoid}^{-1}(0.73)$ ) and  $t = \theta = sh = 0$ . This initialization leverages the effectiveness of resizing operations demonstrated in previous research while providing a good starting point for optimization. The color transformation prompt is initialized to  $\hat{\sigma} = 1$ , corresponding to no color transformation of the original image. The additive prompt  $\delta$  is initialized to zero, following the initialization strategy used in [Wu et al., 2024]. This initialization strategy ensures that the training begins with a controlled transformation (modest resizing) while maintaining the essential visual information, allowing the model to gradually learn more complex and task-specific transformations.

During the training process, all three components of ACAVP—affine, color, and additive transformations—are jointly optimized using stochastic gradient descent to minimize the task-specific loss function. The TrivialAugment data augmentation is applied to each mini-batch before the prompting transformations, providing the necessary input diversity to prevent overfitting.

## D Experimental Setup

### D.1 Datasets

We evaluated our method on 12 downstream classification tasks: CIFAR100, CIFAR10 [Krizhevsky et al., 2009], Flowers102 [Nilsback and Zisserman, 2008], Food101 [Bossard et al., 2014], EuroSAT [Helber et al., 2019], SUN397 [Xiao et al., 2010], DTD [Cimpoi et al., 2014], UCF101 [Soomro et al., 2012], SVHN [Netzer et al., 2011], OxfordPets [Parkhi et al., 2012], GTSRB [Houben et al., 2013], and CLEVR [Johnson et al., 2017]. These datasets encompass diverse visual domains, including natural scenes, human actions, textures, and fine-grained object details, providing a comprehensive evaluation of our method’s generalization capability.

For dataset splits, we follow the CoOp protocol [Zhou et al., 2022b,a]. For datasets with predefined validation sets, we use them directly. For others, we randomly allocate 10% of the training data for validation and use the remaining 90% for training. We select the checkpoint with the highest validation accuracy for final evaluation. Detailed dataset information is provided in Table 9.

### D.2 Model Architectures

We evaluate ACAVP on two representative architectures: CLIP ViT-B/32 [Radford et al., 2021] and ImageNet-1k [Russakovsky et al., 2015] pre-trained ResNet50 [He et al., 2016]. For CLIP

Table 9: Detailed dataset information.  $\dagger$  indicates that data splits provided by CoOp were used.

Dataset	Train Size	Validation Size	Test Size	Classes	Image Size
CIFAR100	45,000	5,000	10,000	100	32×32
CIFAR10	45,000	5,000	10,000	10	32×32
Flowers102 $\dagger$	4,093	1,633	2,463	102	128×128
Food101 $\dagger$	50,500	20,200	30,300	101	128×128
EuroSAT $\dagger$	13,500	5,400	8,100	10	128×128
SUN397 $\dagger$	15,880	3,970	19,850	397	128×128
UCF101 $\dagger$	7,639	1,898	3,783	101	128×128
SVHN	65,931	7,326	26,032	10	32×32
OxfordPets $\dagger$	2,944	736	3,669	37	128×128
DTD $\dagger$	2,820	1,128	1,692	47	128×128
GTSRB	35,288	3,921	12,630	43	32×32
CLEVR	63,000	7,000	15,000	8	480×320

experiments, we use the prompt template "This is a photo of a [LABEL]" for all datasets except CLEVR, where we use "This is a photo of [LABEL] objects" to better match its task-specific nature. For the ImageNet pre-trained model, we experiment with two label mapping strategies: random label mapping, which randomly assigns ImageNet classes to target dataset classes, and the label mapping method proposed by [Wu et al., 2024].

### D.3 Baselines

We compare ACAVP against several baseline methods. Zero-shot inference (ZS) using CLIP serves as our basic baseline without any adaptation. We also compare against standard visual prompting (VP) [Bahng et al., 2022] that adds learnable prompts around the input image, Enhanced VP (EVP) [Wu et al., 2024] that applies fixed resizing before adding prompts, and AutoVP [Tsao et al., 2024] that learns the resize ratio parameter along with the prompt. These baselines represent the current state-of-the-art visual prompting techniques, comprehensively evaluating our method's advantages.

### D.4 Training details

In our training setup, we used an initial learning rate of 40 with a Cosine Annealing Learning Rate Scheduler to gradually decrease the learning rate throughout training. The total number of epochs was set to 1000 for all experiments. We employed SGD as our optimizer with a momentum of 0.9. For most datasets, we used a batch size of 256, with the exception of DTD dataset where we used a batch size of 64 due to its smaller size. For input preprocessing, we first resized all training images to 256×256 pixels, then performed center cropping to 224×224 pixels, followed by normalization. For EVP training, we additionally applied horizontal flip as a data augmentation technique and implemented gradient normalization to stabilize training. When training AutoVP with CLIP models, we used gradient clipping via `torch.nn.utils.clip_grad_value_` to limit the maximum gradient value to 0.001. Similarly, for ACAVP training, we applied gradient clipping with a maximum value of 0.001, and when using CLIP models, we also employed gradient normalization for improved stability.

## E Additional Experiments

### E.1 Impact of Data Augmentation on ACAVP

We analyze the effectiveness of different data augmentation techniques when applied to ACAVP training. The results of our comparison are presented in Table 10. As shown in the table, all augmentation techniques improved the performance of ACAVP compared to training without augmentation.

### E.2 Visualization Analysis

We show visualizations of the transformed images generated by different VP methods and the Grad-CAM [Selvaraju et al., 2020] activation maps in Figure 2. For the CLIP ViT model, we observe that

Table 10: Comparison of data augmentation techniques on ACAVP with CLIP ViT-B/32.

Data Augmentation	CIFAR10	Flowers
–	$95.74 \pm 0.03$	$87.55 \pm 0.64$
IPIMix	$95.96 \pm 0.03$	$90.65 \pm 0.27$
RandAugment	$96.01 \pm 0.10$	$88.24 \pm 0.18$
TrivialAugment	$96.27 \pm 0.07$	$89.36 \pm 0.19$

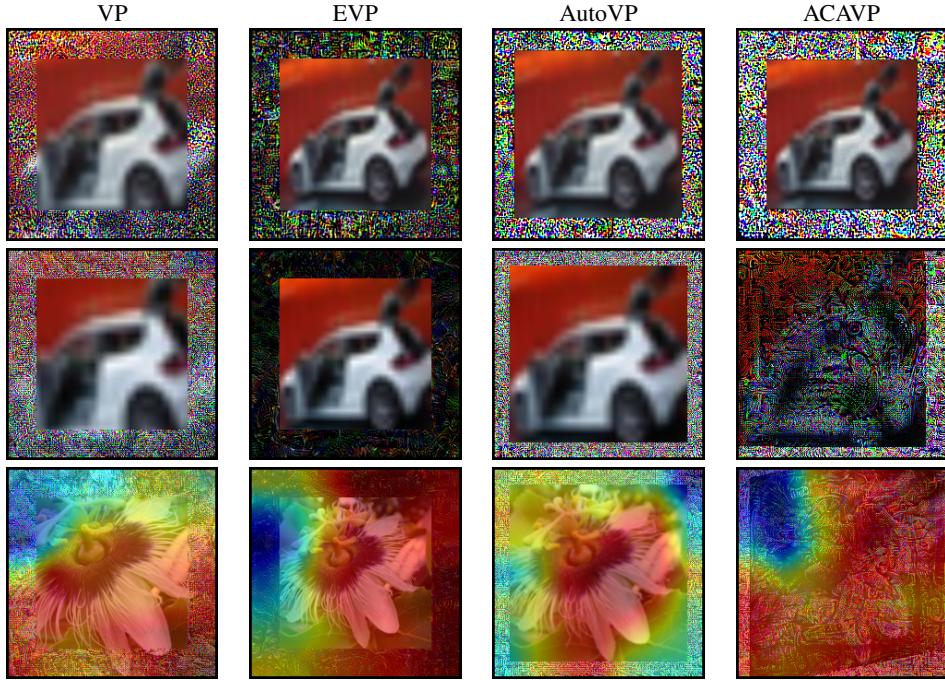


Figure 2: Visualization analysis of different VP methods. The first row shows transformed images on the CIFAR10 dataset using CLIP ViT-B/32. The second row presents transformed images on the CIFAR10 dataset using ResNet50. The third row displays Grad-CAM activation maps on the Flowers dataset using ResNet50.

ACAVP applies a minimal affine transformation, appearing similar to the initial resize operation. This conservative transformation is logical considering that CLIP ViT already achieves high zero-shot classification accuracy on the CIFAR10 dataset (88.93% as shown in Table 2). When a model has strong prior knowledge about the target domain, extensive transformations may be unnecessary or even detrimental, and our method adaptively learns this characteristic. In contrast, when applied to ResNet50 on the CIFAR10 dataset, ACAVP performs more substantial transformations compared to other VP approaches. This is consistent with ResNet50’s lower zero-shot classification capabilities on this dataset, necessitating more significant input modifications to improve accuracy. The ability of ACAVP to adapt its transformation magnitude based on the backbone model inherent capabilities demonstrates its flexibility across different model architectures. The capacity of ACAVP for more expressive transformations stems from the inclusion of both affine transformation and color transformation, which provide a richer transformation space compared to conventional VP approaches that rely solely on additive noise. This enhanced flexibility allows our method to apply appropriate transformations based on the specific requirements of the dataset and backbone model combination. The Grad-CAM visualization results reveal another important advantage of our approach. ACAVP produces higher activation values across broader regions of the image compared to other VP methods, indicating more comprehensive utilization of the image information. This suggests that our approach enables the model to consider a wider range of visual features during classification, rather than focusing on limited regions as observed with other VP methods.

Table 11: Classification accuracy comparison on the CIFAR10 and Flowers datasets using various models. The parentheses indicate model architecture specifications.  $\dagger$  indicates results with label mapping proposed by Wu et al. [2024].

Model	ResNet101 $\dagger$		ViT-L/16 $\dagger$		DINOv2(ViT-S/14) $\dagger$		SigLIP(ViT-SO400M/14)		Average
	CIFAR10	Flowers	CIFAR10	Flowers	CIFAR10	Flowers	CIFAR10	Flowers	
ZS	57.32	5.16	69.49	4.95	72.35	8.89	95.30	88.96	50.30
VP	67.44	10.72	90.69	13.60	81.63	24.69	96.41	97.00	60.27
EVP	72.86	13.52	94.96	27.49	91.63	10.23	98.76	98.66	63.51
AutoVP	68.82	13.20	94.75	25.74	94.25	43.85	98.72	97.48	67.10
ACAVP	<b>75.99</b>	<b>17.66</b>	<b>95.38</b>	<b>39.71</b>	93.62	38.49	<b>98.83</b>	<b>98.74</b>	<b>69.80</b>

### E.3 Hyperparameters sensitivity

In this section, we investigate the sensitivity of ACAVP to its key hyperparameters. The ACAVP introduces several hyperparameters that control the range of transformations applied to input images. Specifically, we analyze the impact of hyperparameters  $R_\theta$ ,  $R_{sh}$ , and  $R_t$  that restrict the range of affine transformation, and  $R_\sigma$  that limits the range of color transformation.

We conducted experiments using the CLIP ViT-B/32 model on the CIFAR10 and Flowers datasets. For the affine transformation parameters, we varied the scaling factor  $X$  applied to the default values ( $R_t = 0.05$ ,  $R_\theta = R_{sh} = 0.1$ ). For the color transformation parameter  $R_\sigma$ , we tested values ranging from 2.5 to 20.0, with the default value being 6.0. All other training settings remained consistent with our main experimental setup.

The results of our hyperparameter sensitivity analysis are presented in Figure 3. As shown in the figure, the performance of ACAVP exhibits some variation across different hyperparameter values, but the impact is relatively modest. For the CIFAR10 dataset, the accuracy ranges from 95.9% to 96.3% across different affine transformation hyperparameter values, and from 96.0% to 96.3% for different color transformation hyperparameter values. Similarly, for the Flowers dataset, the accuracy ranges from 88.0% to 89.5% for affine transformation parameters and from 88.5% to 89.5% for color transformation parameters.

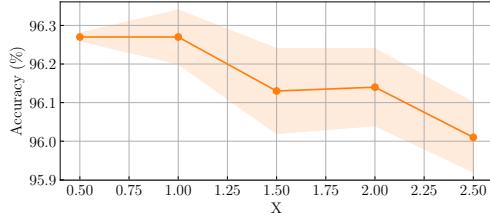
Importantly, ACAVP consistently outperforms the strongest baselines across all hyperparameter configurations tested. For the CIFAR10 dataset, the second-best baseline (EVP) achieves 95.87% accuracy, while ACAVP exceeds this performance regardless of the hyperparameter values selected. Similarly, for the Flowers dataset, the second-best baseline (AutoVP) achieves 82.69% accuracy, which is substantially lower than even the worst-performing ACAVP configuration.

These results demonstrate that while ACAVP’s performance can be further optimized through hyperparameter tuning, the method is relatively robust to hyperparameter choices. This robustness is particularly valuable in practical applications, as it suggests that ACAVP can achieve strong performance even without extensive hyperparameter optimization. The consistent performance advantage over baseline methods across all hyperparameter configurations further validates the effectiveness of the proposed approach.

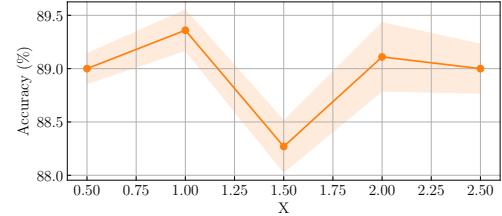
### E.4 Experiments with other models

We conducted additional experiments using the other models (SigLIP [Zhai et al., 2023], DINOv2 [Oquab et al., 2023], ViT-L/16, and ResNet101). All experiments were conducted with a single run. We set the number of training epochs to 200. For learning rate optimization, we evaluated two different learning rates: 40 and 0.1, and reported the results from the higher-performing configuration. The results are presented in Table 11. ACAVP achieves the highest accuracy across all architectures except DINOv2, where it achieves the second-highest performance. All VP methods, including ACAVP, can improve the performance of models with high zero-shot capabilities such as SigLIP. Notably, ACAVP demonstrates strong performance even with state-of-the-art vision encoders.

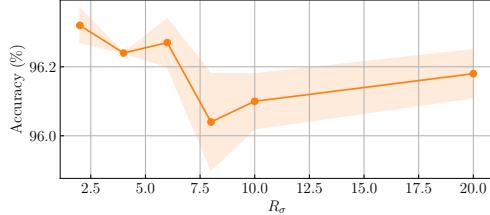
Next, Table 12 shows the computational overhead (the relative time compared to the inference time of each model) for each model. Since these models have shorter inference times than the CLIP ViT-B/32



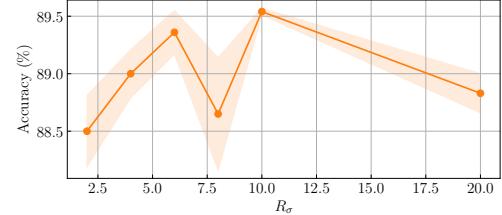
(a) AVCAVP accuracy when changing the hyperparameters of the affine transformation on the CIFAR10 dataset.



(b) AVCAVP accuracy when changing the hyperparameters of the affine transformation on the Flowers dataset.



(c) AVCAVP accuracy when changing the hyperparameter of the color transformation on the CIFAR10 dataset.



(d) AVCAVP accuracy when changing the hyperparameter of the color transformation on the Flowers dataset.

Figure 3: Performance of ACAVP with different hyperparameter values using CLIP ViT-B/32. (a) Accuracy on the CIFAR10 dataset when scaling affine transformation hyperparameters ( $R_t$ ,  $R_\theta$ ,  $R_{sh}$ ) by a factor of  $X$  from their default values, with the x-axis showing the value of  $X$ . (b) Accuracy on the Flowers dataset when scaling affine transformation hyperparameters by a factor of  $X$ , with the x-axis showing the value of  $X$ . (c) Accuracy on the CIFAR10 dataset with different values of the color transformation hyperparameter  $R_\sigma$ , with the x-axis showing the value of  $R_\sigma$ . (d) Accuracy on the Flowers dataset with different values of the color transformation hyperparameter  $R_\sigma$ , with the x-axis showing the value of  $R_\sigma$ .

Table 12: Inference time comparison across different methods using H100 GPU with batch size 500. Relative Time is normalized to the model’s inference time. The parentheses indicate model architecture specifications.

Method	ResNet101	ViT-L/16	DINOv2(ViT-S/14)	SigLIP(ViT-SO400M/14)
VP	0.00	0.01	0.01	0.00
EVP	0.02	0.03	0.04	0.01
AutoVP	1.94	3.38	5.33	1.53
ACAVP	0.18	0.31	0.49	0.14
Coordinator	21.92	38.24	60.33	17.32

used in the main experiments, the relative times for ACAVP are slightly higher than those reported in Table 8. However, across all models, ACAVP requires less than half the inference time of the model itself. Notably, for the high-performing SigLIP, the relative time is only 0.14, indicating minimal overhead. While we have not yet pursued this direction, further optimization of the implementation could potentially reduce the computational time of VP methods even more.

## E.5 Experiments on the ImageNet dataset

The VP method primarily targets users with limited computational resources and is generally evaluated on small to medium-sized datasets, but demonstrating performance on standard benchmarks is crucial. For this purpose, we conducted additional experiments using the ImageNet dataset [Russakovsky et al., 2015]. We used CLIP ViT-B/16 and trained ACAVP for 10 epochs. The results are presented in Table 13. Although the performance gain is modest in this preliminary experiment, it demonstrates

Table 13: Classification accuracy comparison on the ImageNet dataset using CLIP ViT-B/16.

Method	ImageNet
ZS	67.48
ACAVP	<b>67.64</b>

Table 14: Comparison of classification accuracy with generative models. We used InstructBLIP for VP training and did not use BLIP2 during training, only during testing.

Model	InstructBLIP		BLIP2		
	Method	CIFAR10	CIFAR100	CIFAR10	CIFAR100
ZS		88.11	82.41	58.41	60.65
EVP		98.40	85.05	83.20	58.39
TVP		98.78	83.10	85.15	62.80
ACAVP		<b>98.85</b>	<b>88.72</b>	<b>85.32</b>	<b>66.67</b>

that ACAVP can be extended to large-scale datasets like ImageNet. This result highlights the potential flexibility and scalability of our method beyond its original low-resource target scenario.

## E.6 Experiments on generative models and transferability to unseen models

Generative models are increasingly prevalent in current state-of-the-art technology, making it important to demonstrate that our method is effective against such models. We used InstructBLIP [Dai et al., 2023] as the recognition model for VP training and evaluated both InstructBLIP and BLIP2 [Li et al., 2023] on the CIFAR10 and CIFAR100 datasets. BLIP2 was not used during VP training but only during evaluation. During this experiment, we did not access BLIP2’s architecture or gradient information; we used only its input-output interface. This setting replicates the typical usage scenario of cloud APIs and demonstrates the applicability of our method in actual black-box environments. The results are presented in Table 14. The results demonstrate that all VP methods are indeed effective on generative models like InstructBLIP. Notably, ACAVP achieves the highest accuracy, outperforming existing methods across both datasets. Interestingly, ACAVP exhibits superior transferability compared to TVP, even though TVP incorporates additional loss functions specifically designed to enhance transferability to models not used during training (BLIP2 in this case). This suggests that the expanded transformation space and overfitting mitigation introduced by ACAVP effectively improve VP’s generalization performance and transferability.

## E.7 Experiments on the object detection and semantic segmentation tasks

We conducted experiments on the object detection and semantic segmentation tasks to evaluate the broader applicability of our approach. For the object detection task, we trained and evaluated an SSD [Liu et al., 2016] model (with ImageNet pre-trained VGG16 [Simonyan and Zisserman, 2014] backbone) on PASCAL VOC2007 [Everingham et al., 2015]. For the semantic segmentation task, we trained and evaluated a DeepLabv3 [Chen et al., 2017] model (with ImageNet pre-trained ResNet50 backbone) on PASCAL VOC2012. We jointly optimized VP and the classification heads of these models during training. The results are presented in Table 15. ACAVP demonstrates superior performance compared to the baseline VP method across both tasks. These results indicate that ACAVP’s enhanced transformation space is effective beyond image classification tasks. While the absolute performance on the object detection task is relatively modest, this can be attributed to experimental constraints imposed by limited time resources, specifically the reduced number of training epochs and the use of only PASCAL VOC2007 for training (whereas standard practice typically involves training on both PASCAL VOC2007 and PASCAL VOC2012 datasets).

## E.8 Comparisons with Visual Prompt Tuning

We compare ACAVP and VPT [Jia et al., 2022] using the accuracy reported in [Wu et al., 2024]. The results are presented in Table 16. While VPT performs better on some datasets, ACAVP outperforms VPT on the majority of them, demonstrating its competitive performance.

Table 15: Performance comparison on the object detection and semantic segmentation tasks.

Method	PASCAL VOC2007 mAP	PASCAL VOC2012 mIoU
ZS	32.8	62.3
ACAVP	<b>33.0</b>	<b>64.1</b>

Table 16: Classification accuracy comparison of ACAVP and VPT using CLIP ViT-B/32.

Method	CIFAR10	CIFAR100	CLEVR	DTD	EuroSAT	Flowers	Food	Pets	SUN	SVHN
VPT	95.0	76.6	58.6	61.6	94.6	76.2	<b>84.7</b>	<b>92.1</b>	<b>69.3</b>	86.1
ACAVP	<b>96.27</b> $\pm$ 0.07	<b>80.06</b> $\pm$ 0.25	<b>77.19</b> $\pm$ 0.17	<b>67.43</b> $\pm$ 0.83	<b>97.51</b> $\pm$ 0.09	<b>89.36</b> $\pm$ 0.19	77.86 $\pm$ 0.09	87.55 $\pm$ 0.02	64.95 $\pm$ 0.10	<b>91.81</b> $\pm$ 0.18

## F Limitations

While ACAVP demonstrates superior performance across diverse datasets and model architectures, it has certain limitations that should be acknowledged. First, although the computational overhead of ACAVP is relatively minimal compared to model inference (only 6% of the model’s computational cost), it still requires more computation than simpler VP methods such as VP and EVP. This increased computational requirement might become relevant in extremely resource-constrained environments where even small additional costs are significant.

ACAVP’s transformation space, while significantly more expressive than conventional VP approaches, is still limited to affine, color, and additive transformations. More complex transformations, such as non-linear geometric transformations or advanced image processing operations, could potentially further enhance the expressive power of VP. However, incorporating such transformations would likely increase the computational overhead and potentially introduce additional training challenges.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The claims made in the abstract are detailed in the introduction part, and further discussed in each section.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations are discussed in Appendix F.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: All assumptions and proof are included in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: **[Yes]**

Justification: Implementation details and experimental settings are provided in Appendix C and Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

#### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: **[No]**

Justification: Due to the policies of the affiliated organization, the code cannot be made public until it has been accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: **[Yes]**

Justification: Detailed dataset information is included in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: **[Yes]**

Justification: Experiments are run on three seeds, with the standard error being reported in all results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: **[Yes]**

Justification: The computational resources (GPU used in the experiment and execution time) are described in Table 8.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: **[Yes]**

Justification: The research conducted in the paper conform with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: **[NA]**

Justification: Since the proposed method in this paper are used to improve the performance of VP in downstream classification tasks, there is no potential societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not release data or models that have a high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Creators or original owners of code or data have been cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

### 16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorosity, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLM is used only for writing, editing, or formatting purposes.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.