Exploring the Transferability of Visual Prompting for Multimodal Large Language Models

Yichi Zhang[1,2] Yinpeng Dong[1,2]✉ Siyuan Zhang[1] Tianzan Min[1] Hang Su[1,3]✉ Jun Zhu[1,2,3]
[1]Dept. of Comp. Sci. and Tech., Institute for AI, Tsinghua-Bosch Joint ML Center,
THBI Lab, BNRist Center, Tsinghua University, Beijing 100084, China
[2]RealAI [3] Pazhou Laboratory (Huangpu), Guangzhou, Guangdong
{zyc22@mails., dongyinpeng@, siyuan-z20@mails., suhangss@, dcszj@}tsinghua.edu.cn

Abstract

Although Multimodal Large Language Models (MLLMs) have demonstrated promising versatile capabilities, their performance is still inferior to specialized models on downstream tasks, which makes adaptation necessary to enhance their utility. However, fine-tuning methods require independent training for every model, leading to huge computation and memory overheads. In this paper, we propose a novel setting where we aim to improve the performance of diverse MLLMs with a group of shared parameters optimized for a downstream task. To achieve this, we propose Transferable Visual Prompting (TVP), a simple and effective approach to generate visual prompts that can transfer to different models and improve their performance on downstream tasks after trained on only one model. We introduce two strategies to address the issue of cross-model feature corruption of existing visual prompting methods and enhance the transferability of the learned prompts, including 1) Feature Consistency Alignment: which imposes constraints to the prompted feature changes to maintain task-agnostic knowledge; 2) Task Semantics Enrichment: which encourages the prompted images to contain richer task-specific semantics with language guidance. We validate the effectiveness of TVP through extensive experiments with 6 modern MLLMs on a wide variety of tasks ranging from object recognition and counting to multimodal reasoning and hallucination correction.

1. Introduction

The recent success of Large Language Models (LLMs) [45, 49, 50] has motivated researchers to explore their capabilities in solving multimodal tasks. Tremendous efforts have been made to develop Multimodal Large Language Models (MLLMs) [2, 9, 34, 37, 69], which seamlessly integrate visual input into LLMs by aligning image features with text embeddings. These models have achieved remarkable performance in image understanding and reasoning [15, 32, 59] and serve as "foundation models" [5] for a variety of tasks. Despite their excellent generalization performance, existing MLLMs usually lag behind the specialized state-of-the-art models on downstream tasks (*e.g.*, image classification), especially when evaluated in zero-shot manner [59, 62]. This is because MLLMs are primarily pre-trained on massive data and fine-tuned on a small amount of modality alignment and instruction data [9, 37, 69], while lacking specialized training on certain tasks. Consequently, when users aim to employ MLLMs for downstream tasks, their performance is far from satisfactory, making it necessary to develop effective and efficient strategies to bridge this gap and enhance the utility of MLLMs in task-specific applications.

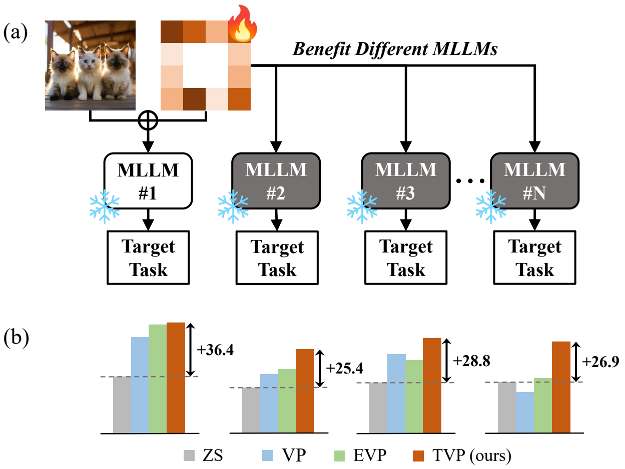


Figure 1. **(a) Illustration of problem setting:** We aim to improve the performance of different MLLMs on a specific task with a set of shared parameters. This is achieved by exploiting the transferability of the visual prompts trained on one model and using them on other models. **(b) Demonstration of the effect:** We show the partial results on SVHN [42] with the visual prompt trained on MiniGPT-4 [69] and tested on InstructBLIP [9], BLIP2 [34] and BLIVA [21]. Compared with the existing visual prompting methods [3, 56], the proposed Transferable Visual Prompting (TVP) improves different models with larger margins. Detailed results are in Sec. 4.2. ZS is for zero-shot inference when non-prompted.

✉ Corresponding authors. Code available at https://github.com/zycheiheihei/Transferable-Visual-Prompting

Adapting MLLMs for downstream tasks conventionally requires fine-tuning on task-specific data. Though effective in various fields, such as science [37] and biomedicine [33], full-parameter fine-tuning (FFT) is computationally demanding and storage-intensive, particularly for models with billions of parameters. To address these problems, various parameter-efficient fine-tuning (PEFT) techniques have been proposed, including Adapters [18], LoRA [20], and prompt tuning [24, 38]. These methods perform gradient-based optimization of additional model-specific parameters for a downstream task. Nevertheless, they require a prohibitive amount of memory for optimization and the resultant parameters lack generalizability across different models. In a practical scenario, users with no prior knowledge of PEFT and limited computation resources would prefer auxiliary parameters (*e.g*., prompts) that can improve their own models on downstream tasks without further fine-tuning. This leads to a novel and challenging setting that we aim to develop a set of *shared parameters* that can benefit numerous MLLMs on the same task, while only optimized on one or few of them, as shown in Fig. 1. This pathway is hopeful to be resource-friendly and flexible, making it easier to adapt different models for a given task simultaneously, even when model weights are not accessible. Moreover, it aligns with the "Prompt as a Service" (PaaS) paradigm [58], where users can request a prompt for a downstream task from the PaaS provider while keeping the local models confidential.

As MLLMs take images as input, the image pixel space is a promising shared space for parameter learning. Previous methods have explored visual prompting (VP) [3, 56] to adapt pre-trained models for downstream tasks. VP learns parameters in pixel space around clean images as a frame, known as visual prompts. Inspired by the transferability of adversarial examples [11, 68], we take transferring visual prompts to boost the performance of other models as a feasible solution for our problem. However, the transferability of the learned visual prompts for other models is yet to be studied. We find that, though VP can effectively elevate the performance of models used for prompt training, it can lead to limited performance improvement or significant degradation for other models. We attribute this to the fact that the trained prompts lead to notable changes in visual features across different models, defined as *cross-model feature corruption*. This indicates that the visual prompts overfit the model for their training and invalidate the plenty knowledge acquired from large-scale pre-training when transferred to other models, thus impacting their performance.

In this paper, we propose **Transferable Visual Prompting (TVP)** to enhance the transferability of visual prompts across MLLMs and improve these models simultaneously. To achieve this, we formulate a unified framework of VP on different tasks for MLLMs. We propose two key strategies to fortify both general knowledge and task-specific representations. First, we propose **Feature Consistency Alignment (FCA)** to mitigate the issue of feature corruption that highly suppresses the transferability. FCA facilitates model adaptation to downstream tasks by imposing constraints on visual features after applying prompts, preserving essential inner knowledge. Consequently, it helps models better retain and leverage task-agnostic representations for improvement. Second, we introduce **Task Semantics Enrichment (TSE)** to further embed task information explicitly into visual prompts. By leveraging CLIP [44], TSE encourages the prompted images to exhibit semantic similarity with text features tailored for specific tasks, rather than simply utilizing task-specific objectives for end-to-end prompt learning. This enables models to extract better shareable task-specific semantics and get improved on the target tasks.

We examine the performance of TVP through substantial experiments. The visual prompts trained with one single model can facilitate the overall performance of 6 modern MLLMs on 10 datasets ranging from visual tasks like recognition and counting to multimodal reasoning and hallucination correction, which significantly surpasses the existing visual prompting baselines. The performance is further improved with model ensembling. We demonstrate that TVP can enhance different models with diverse data scales, generalize to different datasets, and resist image corruptions, emphasizing the practicality of our method in real scenarios. Comparisons with existing fine-tuning methods suggest the feasibility of improving different models with shared parameters and the effectiveness of our TVP.

2. Related Works

In this section, we briefly review the related works in the context of Multimodal Large Language Models and adaptation methods for large-scale pre-trained models.

2.1. Multimodal Large Language Models

The significant advancements in LLMs for language-centric tasks have spurred investigations into their potential applications in diverse multimodal contexts [60]. This exploitation is primarily manifested in works focusing on modality alignment and instruction tuning [2, 9, 34, 37, 69]. These proposed models lay the foundation for MLLMs and many subsequent works have been proposed to improve performance concerning the issues of in-context learning [64], efficient training [63], richer modalities [48], *etc*.

Several benchmarks [15, 32, 59] have demonstrated that MLLMs show versatile capabilities in visual perception and comprehension. However, their performance falls short of specialized models on specific tasks, limiting their applicability in certain scenarios [59]. Moreover, MLLMs face challenges related to safety and reliability, including issues of value alignment [39] and hallucination [16, 35]. MLLMs need further tuning to address these challenges.

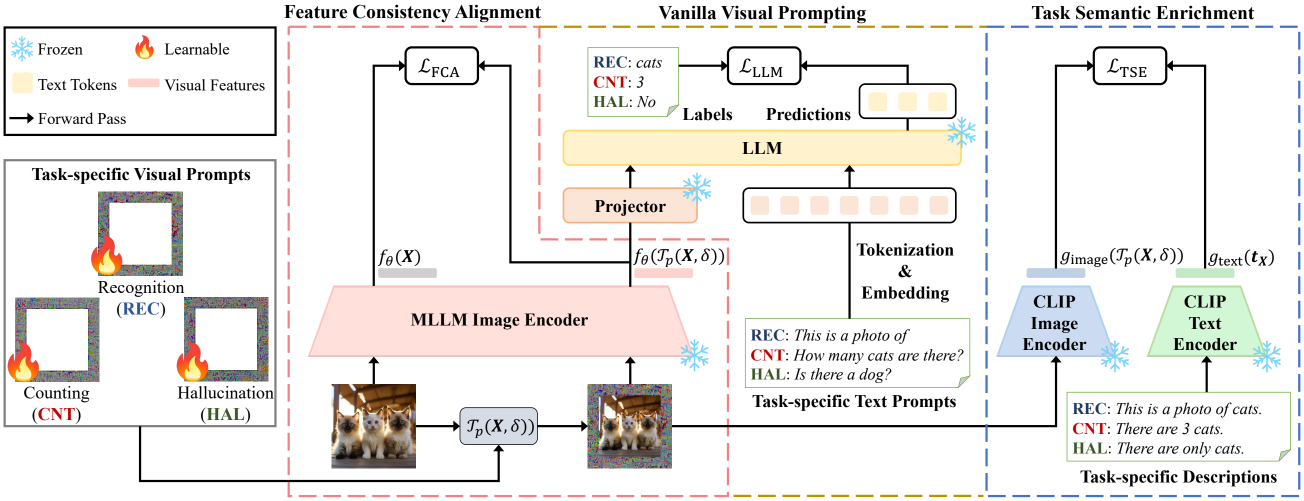

Feature Consistency Alignment
Vanilla Visual Prompting
Task Semantic Enrichment
Frozen
Learnable
Text Tokens
Visual Features
Forward Pass
Task-specific Visual Prompts
Recognition (REC)
Counting (CNT)
Hallucination (HAL)
$\mathcal{L}_{\text{FCA}}$
$\mathcal{L}_{\text{LLM}}$
$\mathcal{L}_{\text{TSE}}$
REC: cats
CNT: 3
HAL: No
Labels
Predictions
LLM
Projector
$f_\theta(\boldsymbol{X})$
$f_\theta(\mathcal{T}_p(\boldsymbol{X},\delta))$
MLLM Image Encoder
$\mathcal{T}_p(\boldsymbol{X},\delta)$
Tokenization & Embedding
REC: This is a photo of
CNT: How many cats are there?
HAL: Is there a dog?
Task-specific Text Prompts
$g_{\text{image}}(\mathcal{T}_p(\boldsymbol{X},\delta))$
$g_{\text{text}}(\boldsymbol{t_X})$
CLIP Image Encoder
CLIP Text Encoder
REC: This is a photo of cats.
CNT: There are 3 cats.
HAL: There are only cats.
Task-specific Descriptions

Figure 2. Overview of our proposed Transferable Visual Prompting (TVP) method for adapting MLLMs. TVP optimizes a visual prompt on a single MLLM towards a downstream task. Feature Consistency Alignment (FCA) and Task Semantic Enrichment (TSE) are proposed to make learned visual prompts more transferable and benefit more unseen MLLMs to improve on the same task.

2.2. Adaptation for Large-Scale Pre-trained Models

Adapting MLLMs mainly follows methods for large models (*e.g.*, LLMs [49], CLIP [44]). Fine-tuning for a downstream task is straightforward but costly in computation and storage. Parameter-efficient fine-tuning (PEFT) methods, such as Adapters [18], LoRA [20], and prompt tuning [24, 38], have emerged to ease these challenges. Some recent advanced works also focus on efficient modality bridging and adaptation via routing and skipping with adapters [40, 57]. However, they are inherently model-specific and require access to the inner structure of models, diverging from our goal of optimizing a single set of parameters to adapt multiple models in a resource-friendly and flexible manner.

Recent developments in visual prompting [3], inspired by adversarial reprogramming [14, 51], offer a promising solution for model adaptation by introducing learnable perturbations in the pixel space of images. As the pixel space is a shared domain for different models, it becomes a natural choice for parameter tuning. Many follow-up works have explored topics like performance refinement [56] and data generalization [22, 28], but none has studied the generalization of visual prompts across models, or their transferability as defined in adversarial attacks [10, 68]. While popular works of prompt tuning like CoOp [65, 66], VPT [24] and MaPLe [27] operate soft prompts for both modalities at the early layers of the model, even at the embedding space, they are invalid under complete black-box conditions where only discrete texts and images are accessible for input.

In this paper, we investigate the direct transfer of trained visual prompts to other MLLMs for adaptation. This reduces the computation and storage overloads, and also offers a more convenient and flexible solution in diverse application scenarios like "Prompt as a Service" (PaaS) [58], where users can directly request a visual prompt towards a certain task for their local models from the PaaS provider with a guarantee of the model confidentiality.

3. Methods

Visual prompting offers an effective means to adapt vision-language models, such as CLIP [44], to downstream visual tasks without resorting to fine-tuning. In this study, we extend the application of VP to MLLMs and investigate its potential for enhancing performance across a range of models. Although existing methods can enhance model performance through prompt training, these trained prompts often fall short when applied to other models due to issues related to feature corruption. To this end, we introduce the method of Transferable Visual Prompting (TVP), aiming to enhance the transferability of visual prompts across diverse MLLMs.

In this section, we will first briefly present some preliminaries about MLLMs and VP, then formulate our problem of transferring visual prompts across MLLMs, and finally introduce our proposed TVP approach. The overview of our method is depicted in Fig. 2.

3.1. Preliminaries

Multimodal Large Language Models. MLLMs primarily use an architecture that projects visual features to the text embedding space to integrate images with LLMs [9, 21, 69].

To be specific, assume that we have a visual encoder f_θ, an LLM P_ϕ and a projector h_ψ. The textual response $\mathbf{r}$ of an MLLM given image input $\mathbf{X}$ and text input $\mathbf{t}$ is decided autoregressively according to the likelihood

$$\mathbf{r}_i \sim P_\phi(\mathbf{r}_i | h_\psi(f_\theta(\mathbf{X})), \mathbf{t}, \mathbf{r}_{<i}), \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{3\times H\times W}$ is an RGB image and $\mathbf{t} \in \mathbb{V}^N$ is a text with N tokens from vocabulary $\mathbb{V}$. Image features $f_\theta(\mathbf{X})$ are mapped by a projector h_ψ (*e.g.*, MLP [9, 37, 69]) to align with the text and further concatenated with text tokens as unified input for the downstream LLM.

Visual Prompting. As proposed in [14] and [3], a trainable visual prompt $\boldsymbol{\delta} \in \mathbb{R}^{3\times H\times W}$ is learned in the pixel space and imposed to the clean images with different transformations $\mathcal{T}$ (*e.g.*, global perturbations [43], padding [3]) to adapt models to a certain downstream task.

In this paper, we follow the common practice of VP [3] by adding universal pixel-level prompt around resized input images. Mathematically, we describe the process of visual prompting as

$$\mathcal{T}_p(\mathbf{X}, \boldsymbol{\delta}) = \text{Resize}_{H\times W\to H'\times W'}(\mathbf{X}) + \underbrace{M_p \odot \boldsymbol{\delta}}_{\text{visual prompt}}, \tag{2}$$

where p is the width of the visual prompt and M_p is a binary mask with a border of width p taking values of 1. The original image $\mathbf{X}$ of size $H \times W$ is resized to $H' \times W' = (H-2p)\times(W-2p)$, so that the prompted image is of the same size as the original without overlapping with $\boldsymbol{\delta}$. We take $H = W = 224$ and $p = 30$ by default.

3.2. Problem Formulation

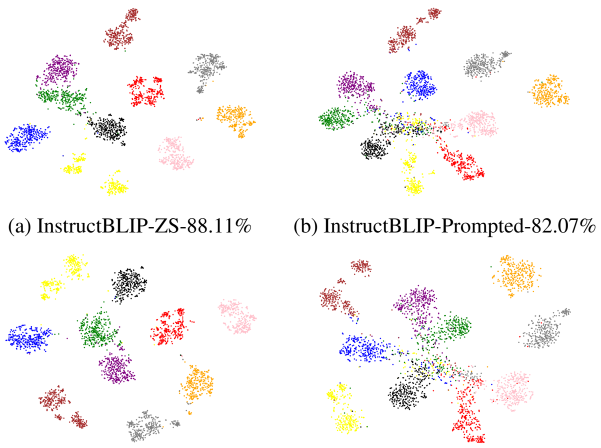
We extend VP [3, 56] to adapt MLLMs for downstream tasks while avoiding heavy computations in massive parameter fine-tuning. To make it more general, we unify different visual tasks into the form of text completion and take the autoregressive loss (*i.e.*, cross-entropy loss over the vocabulary) as the training objective for VP. For a task with dataset $\mathcal{D}$, this is formulated as minimizing $\mathcal{L}_{\text{LLM}}(\boldsymbol{\delta}) =$

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{t},\mathbf{r})\sim\mathcal{D}}\left[\sum_{i=1}^{N_r} -\log P_\phi(\mathbf{r}_i | h_\psi(f_\theta(\mathcal{T}_p(\mathbf{X},\boldsymbol{\delta}))), \mathbf{t}, \mathbf{r}_{<i})\right]. \tag{3}$$

Here, $(\mathbf{t}, \mathbf{r})$ is the prompt-target text pair for a task, with N_r denoting the length of $\mathbf{r}$. Prompts and targets for different tasks are introduced in Appendix A.1.

In this work, we exploit the transferability of visual prompts, inspired by transfer attacks in the field of adversarial robustness [68]. We transfer the one-time trained prompts to other models to improve their performance. Specifically, we optimize a prompt $\boldsymbol{\delta}$ on an MLLM minimizing its loss $\mathcal{L}_{\text{LLM}}$ and expect it can lower the loss $\mathcal{L}'_{\text{LLM}}$ of an arbitrary different MLLM, *i.e.*, to improve its performance when we apply this trained prompt on it without any further fine-tuning on the target task.

After examining existing VP methods [3, 56] on different models, we find that the transferability of the generated visual prompts is poor, resulting in modest improvement or even remarkable performance decline, as shown in Fig. 1. We identify the reason for this phenomenon as "*cross-model feature corruption*" given visual prompts trained on one model. We can observe significant changes in visual features triggered on different models by the visual prompts. As shown in Fig. 3, by plotting the t-SNE [53] of prompted visual features extracted by different models on CIFAR-10, we find that for models with performance degradation, the features of prompted images get mixed up compared to clean images. It indicates that when training visual prompts, they primarily amplify task-specific features that are only useful for the current model, *i.e.*, overfitting to the model for training. However, the feature changes on other models render the knowledge from pre-training ineffective and disrupt the predictions for those models without prompt learning.



(a) InstructBLIP-ZS-88.11%

(b) InstructBLIP-Prompted-82.07%

(c) BLIVA-ZS-89.23%

(d) BLIVA-Prompted-78.85%

Figure 3. t-SNE visualization of visual features from InstructBLIP and BLIVA on CIFAR-10 with and without the visual prompt, which is trained on MiniGPT-4 using VP [3]. When the images are prompted, the visual features of different categories get mixed together, leading to performance degradation.

3.3. Transferable Visual Prompting

To alleviate the issue of feature corruption and further improve the transferability, we present Transferable Visual Prompting (TVP) by integrating two novel strategies with traditional VP techniques. We introduce them as follows.

3.3.1 Feature Consistency Alignment

The above analysis reveals that visual prompts significantly alter image features, leading to a loss of the general knowledge gained from pre-training and a reduction of performance when transferring across models. To counter this, we propose to impose a specific constraint on the divergence between the prompted features and non-prompted features. This is intended to avoid exceptional feature corruption and guide the prompted features to maintain the task-agnostic general knowledge during prompt learning.

We encourage the prompted features to be consistent with original features by a loss of feature consistency alignment (FCA), so that the task-agnostic features and inherent knowledge can be aligned. Given a white-box model $(P_\phi, f_\theta, h_\psi)$ and an input tuple $(\mathbf{X}, \mathbf{t}, \mathbf{r})$, we will get a feature of the plain image from the visual encoder $f_\theta(\mathbf{X})$ and a prompted feature $f_\theta(\mathcal{T}_p(\mathbf{X}, \boldsymbol{\delta}))$ accordingly. The ℓ_2 distance between these features are adopted to measure the divergence and the FCA loss can be computed as $\mathcal{L}_{\text{FCA}}(\boldsymbol{\delta}) =$

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{t},\mathbf{r})\sim\mathcal{D}} \left[\frac{1}{2} \| f_\theta(\mathcal{T}_p(\mathbf{X}, \boldsymbol{\delta})) - f_\theta(\mathbf{X}) \|_2^2 \right]. \tag{4}$$

Here, we condition the prompted features with original features to depress changes in features and preserve effective task-agnostic semantic information. It is expected to make the learned visual prompts more transferable because the prompts are more mild to exploit useful visual representations from unseen models.

From another perspective, the FCA loss can be considered as a form of regularization applied to the prompted feature space, serving as a means of enhancing generalization. Regularization techniques have been commonly used to prevent overfitting and guarantee the model's generalization to unseen data. The main difference is that previous works [28, 30, 54, 67] primarily focus on the generalization across data (*e.g.*, base-to-novel, domain generalization), while our research is centered around the generalization across models. By employing regularization to the prompted features, we seek a balance between maximizing the performance on supervised tasks and maintaining the inherent knowledge embedded in various models.

3.3.2 Task Semantic Enrichment

Besides end-to-end supervised training, we want to explicitly make visual prompts contain richer task-specific semantic information to further boost the performance of visual prompts. This is expected to enhance the performance of diverse models on the target tasks by fostering a shared semantic enhancement across them. CLIP [44] is a vision-language foundation model and has abundant knowledge by image-text alignment. Based on this, many studies have taken CLIP as a bridge to introduce language as additional supervision and guidance for visual tasks [26, 47]. Yet, in the context of prompt learning, it has not been widely studied to use CLIP as a guidance rather than the target model.

We propose another loss of task semantic enrichment (TSE) to explicitly enhance the task-related semantics of prompted images by leveraging CLIP. CLIP consists of a visual encoder g_{image} and a text encoder g_{text}, mapping image input $\mathbf{X} \in \mathbb{R}^{3\times H\times W}$ and text input $\mathbf{t} \in \mathbb{V}^N$ to a shared embedding space $\mathbb{R}^d$ respectively. The correspondence between images and texts can be obtained by computing the distance between their features. By designing task-specific descriptions according to the images, we can maximize their similarity to better embed the task semantics into the prompted images. Referring to the contrastive loss of CLIP, we present an auxiliary loss as $\mathcal{L}_{\text{TSE}}(\boldsymbol{\delta}) =$

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{t}_\mathbf{X})\sim\mathcal{D}} [\exp(\tau \text{sim}(g_{\text{image}}(\mathcal{T}_p(\mathbf{X}, \boldsymbol{\delta})), g_{\text{text}}(\mathbf{t}_\mathbf{X})))], \tag{5}$$

where $\text{sim}(\cdot,\cdot)$ is the cosine similarity and τ is the temperature. $\mathbf{t}_\mathbf{X}$ is the text description of image $\mathbf{X}$ under the target task. Descriptions for different tasks are in Appendix A.1.

By integrating FCA loss and TSE loss along with supervised loss of $\mathcal{L}_{\text{LLM}}$ in Eq. (3), we guide the visual prompt to consolidate and strengthen task-agnostic and task-specific representations while improving the model performance. The overall training objective is in the form of

$$\mathcal{L}(\boldsymbol{\delta}) = \mathcal{L}_{\text{LLM}}(\boldsymbol{\delta}) + \lambda_1 \mathcal{L}_{\text{FCA}}(\boldsymbol{\delta}) - \lambda_2 \mathcal{L}_{\text{TSE}}(\boldsymbol{\delta}), \tag{6}$$

where λ_1 and λ_2 are hyperparameters. For this training objective, we follow EVP [56], which introduces the concepts of input diversity and gradient normalization to improve the performance, and update the learnable prompt at step t by

$$\boldsymbol{\delta}^{t+1} = \boldsymbol{\delta}^t - \gamma \frac{\nabla_{\boldsymbol{\delta}^t} \mathcal{L}(\boldsymbol{\delta}^t)}{||\nabla_{\boldsymbol{\delta}^t} \mathcal{L}(\boldsymbol{\delta}^t)||_2}, \tag{7}$$

where γ is the learning rate.

4. Experiments

In this section, we conduct substantial experiments to verify the effectiveness of the proposed TVP in boosting the transferability of visual prompts.

4.1. Experimental Settings

Here, we briefly list the basic settings for the following experiments. More details are provided in Appendix A.

Datasets and Metrics. We consider 10 datasets involving diverse visual or multimodal tasks. For visual tasks, we take 8 datasets of object recognition (*e.g.*, CIFAR-10 [31], ImageNette [19] and SVHN [42]) and object counting (*e.g.*, CLEVR [25]) for illustration. We further focus on two challenging multimodal problems including multimodal reasoning (Hatefulmemes [29]) and hallucination (POPE [35]), to better demonstrate the effectiveness of our methods in the realm of MLLM. AUC score is taken as the metric for Hatefulmemes while top-1 accuracy is used for the rest.

Models. We select 6 modern MLLMs with diverse capabilities as evaluated by [15] to demonstrate that the generated visual prompts by our method can universally elevate their performance. To be specific, we take MiniGPT-4 [69] and InstructBLIP [9] for training visual prompts respectively and transfer the prompts to BLIP2 [34], VPG-Trans [63], BLIVA [21] and VisualGLM [1].

Recognition: CIFAR-10		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
	Zero-Shot	87.35	88.11	82.41	84.63	89.23	91.81	0.00
MiniGPT-4	VP [3]	92.40*	82.07	78.58	85.82	78.85	81.29	-4.09
	EVP [56]	97.97*	84.57	83.39	86.93	86.45	85.92	+0.28
	TVP (ours)	**98.33***	92.82	91.69	88.70	87.48	87.53	**+3.83**
InstructBLIP	VP [3]	82.80	91.22*	87.46	83.71	85.92	85.21	-1.20
	EVP [56]	87.52	97.81*	86.18	87.16	94.39	90.28	+3.30
	TVP (ours)	91.69	**98.07***	**96.02**	91.09	**97.78**	89.01	**+6.69**
Ensemble	VP [3]	90.84*	90.95*	91.91	88.84	92.00	81.43	+2.07
	EVP [56]	97.69*	97.63*	89.40	91.43	95.47	88.29	+6.06
	TVP (ours)	97.18*	96.21*	95.46	**92.92**	96.20	**92.06**	**+7.75**

Recognition: ImageNette		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
	Zero-Shot	79.82	74.78	95.03	82.57	77.02	73.20	0.00
MiniGPT-4	VP [3]	83.39*	71.12	**95.59**	81.40	80.33	73.73	+0.52
	EVP [56]	96.79*	68.15	91.36	79.08	75.82	76.05	+0.81
	TVP (ours)	**97.71***	78.34	94.98	86.34	84.51	75.34	**+5.80**
InstructBLIP	VP [3]	85.50	93.22*	90.88	79.67	92.94	76.71	+6.08
	EVP [56]	84.05	96.92*	94.65	80.87	90.11	76.46	+6.77
	TVP (ours)	85.58	**98.24***	92.71	83.95	**96.79**	**80.46**	**+9.22**
Ensemble	VP [3]	85.48*	86.55*	93.99	79.08	86.27	71.77	+3.45
	EVP [56]	97.63*	97.16*	92.37	83.19	90.04	76.49	+9.08
	TVP (ours)	97.22*	97.00*	91.54	**92.60**	94.37	76.69	**+11.17**

Recognition: SVHN		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
	Zero-Shot	38.82	28.96	33.14	31.58	33.32	20.87	0.00
MiniGPT-4	VP [3]	65.58*	37.80	52.66	42.97	27.02	20.12	+9.91
	EVP [56]	74.24*	41.59	48.87	**57.61**	36.11	**33.12**	+17.48
	TVP (ours)	75.17*	54.32	61.95	51.10	60.28	32.17	**+24.72**
InstructBLIP	VP [3]	29.53	73.20*	51.63	36.82	50.98	21.86	+12.89
	EVP [56]	39.80	87.55*	45.24	31.37	44.74	17.34	+13.23
	TVP (ours)	54.37	**89.87***	70.61	46.82	70.92	25.38	**+28.55**
Ensemble	VP [3]	58.41*	64.65*	47.95	47.25	50.79	25.00	+17.89
	EVP [56]	70.14*	70.23*	53.27	29.08	65.66	22.35	+20.67
	TVP (ours)	**82.58***	81.82*	**73.06**	37.09	**73.46**	28.61	**+31.54**

Counting: CLEVR		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
	Zero-Shot	9.50	40.33	12.73	13.13	26.27	13.43	0.00
MiniGPT-4	VP [3]	35.73*	33.57	12.77	8.80	29.07	12.77	+2.89
	EVP [56]	52.17*	39.03	20.17	8.00	34.23	13.60	+8.64
	TVP (ours)	51.00*	42.90	22.07	19.50	36.00	13.00	**+11.51**
InstructBLIP	VP [3]	12.53	56.27*	15.57	21.00	44.20	18.03	+8.70
	EVP [56]	12.73	61.93*	13.40	10.40	39.90	13.87	+6.14
	TVP (ours)	21.80	**63.60***	**33.73**	**26.27**	**46.93**	**35.63**	**+18.76**
Ensemble	VP [3]	35.03*	39.43*	16.70	20.83	40.43	13.03	+8.34
	EVP [56]	46.33*	54.63*	24.87	23.77	38.27	13.43	+14.22
	TVP (ours)	**53.60***	58.33*	25.50	23.63	39.53	14.36	**+16.49**

Reasoning: HM		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
	Zero-Shot	53.37	61.62	53.44	59.54	62.28	57.55	0.00
MiniGPT-4	VP [3]	55.55*	57.26	55.54	55.47	56.05	52.73	-2.53
	EVP [56]	57.58*	60.66	55.34	56.87	60.64	57.27	+0.09
	TVP (ours)	56.93*	62.38	56.20	60.19	64.09	58.15	**+1.69**
InstructBLIP	VP [3]	53.20	61.73*	56.24	61.53	62.22	56.26	+0.56
	EVP [56]	55.78	**63.05***	51.19	60.98	61.67	58.00	+0.48
	TVP (ours)	55.65	62.54*	55.52	61.61	62.77	**59.01**	**+1.55**
Ensemble	VP [3]	54.81*	61.88*	55.51	61.54	62.39	57.66	+1.00
	EVP [56]	58.85*	62.26*	54.09	59.21	62.10	58.22	+1.16
	TVP (ours)	**62.33***	62.77*	**58.77**	**62.23**	**64.67**	58.47	**+3.57**

Hallucination: POPE		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
	Zero-Shot	53.07	73.87	49.80	58.60	78.07	70.87	0.00
MiniGPT-4	VP [3]	53.87*	69.27	50.00	61.07	72.13	69.80	-1.36
	EVP [56]	68.06*	69.80	50.00	61.07	71.33	69.40	+0.90
	TVP (ours)	**68.73***	75.13	51.40	64.47	72.67	71.00	**+3.19**
InstructBLIP	VP [3]	50.67	77.33*	50.00	60.87	74.40	67.47	-0.59
	EVP [56]	51.20	**79.33***	50.00	63.47	72.87	69.87	+0.41
	TVP (ours)	54.53	77.80*	50.20	**64.93**	75.20	70.47	**+1.48**
Ensemble	VP [3]	58.33*	76.73*	50.60	64.87	72.13	71.07	+1.57
	EVP [56]	64.67*	75.87*	50.00	64.07	71.80	62.13	+0.71
	TVP (ours)	59.13*	77.53*	**52.40**	**64.93**	**75.27**	**72.40**	**+2.90**

Table 1. Results on 6 datasets of different tasks from object recognition and counting to multimodal reasoning and hallucination. Visual prompts are trained on MiniGPT-4 [69], InstructBLIP [9] and their ensemble with different methods, and further tested on 6 modern MLLMs. We display the average improvements over all models on the last column and $*$ denotes the results of the model for prompt training. AUC score is the metric for Hatefulmemes (HM) and top-1 accuracy (%) is used for the rest. Besides highlighting the best overall average improvement, we mark the best result in **bold** for each model.

Baselines. We mainly focus on the transferability of visual prompting across diverse models and compare the proposed TVP method with VP [3] and EVP [56], which are general methods of visual prompting, rather than those focusing on generalization across different data distributions.

Implementations. Hyperparameters for TVP include two balance weights for the proposed loss terms. Following [24, 43], we set them optimal by grid-search within small ranges on validation sets. The search ranges and other details are introduced in Appendix A.3.

4.2. Main Results

We train visual prompts on either MiniGPT-4 or InstructBLIP for target downstream tasks using different methods and examine their performance on the 6 selected models. The results are displayed in Tab. 1. Several findings are summarized in the following context.

First of all, this is the first time that visual prompting technique [3] is proved to be effective for MLLMs on multimodal tasks involving reasoning besides recognition. Beyond that, VP [3] yields the least favorable outcomes, often leading to an overall performance degradation, indicating poorer transferability. EVP brings more significant improvements for the models training visual prompts due to the stable optimization with normalized gradients, which is aligned with [56], but its benefits for other models are limited. In contrast, while the proposed TVP achieves similar effectiveness on models for training, it also boosts the performance of other models with larger margins. The overall effectiveness is manifested as the highest average delta in growth across 6 models on all downstream tasks, which better achieves our goal of enhancing different models with a single set of shared external parameters.

Moreover, we have some more in-depth findings regarding the results of TVP. The improvements of VisualGLM [1] are relatively modest compared to other models. This suggests that the visual prompts are more challenging to transfer to VisualGLM. The language model of VisualGLM is GLM [13] while the language models for the rest models are based on the LLaMA architecture [49]. This difference in language models can raise the difficulty of effective prompt transferring. Also, we notice that the transferability of visual prompts generated by InstructBLIP is better than those from MiniGPT-4 in general. Though sharing similar architecture, InstructBLIP is tuned systematically on 13 vision-language datasets [9], which is far more plentiful than MiniGPT-4 [69]. This means that InstructBLIP possesses more internal knowledge concerning visual-language tasks, which should help the learning of visual prompts and benefit other models with more improvements.

Similar conclusions are drawn on more practical datasets like CIFAR-100 [31], Oxford-Pets [23], FGVC-Aircraft [41] and Food101 [6], in Tab. 6, Appendix B.

4.3. Model Ensemble

Ensembling methods have been proven effective to enhance model generalization [4, 52] and transfer attack [7, 10, 12]. The idea of ensemble can also be applied to transferable visual prompting. The intuition is that if the visual prompts are effective for multiple models, it is more likely that they contain more shareable informative semantics and transfer better to other models. We simply optimize the visual prompts by averaging over losses of different models, and here take MiniGPT-4 and InstructBLIP for ensemble.

The results of ensemble prompt learning are presented in Tab. 1. We can see that model ensembling is a general technique that can enhance the transferability of visual prompts trained with different methods. Meanwhile, the trend that TVP has the best performance remains unchanged and with the aid of ensemble, it achieves the best overall improvements on four tasks. However, more models bring more computational and storage overheads for ensemble. A balance between training cost and prompt transferability needs to be sought when adopting this technique.

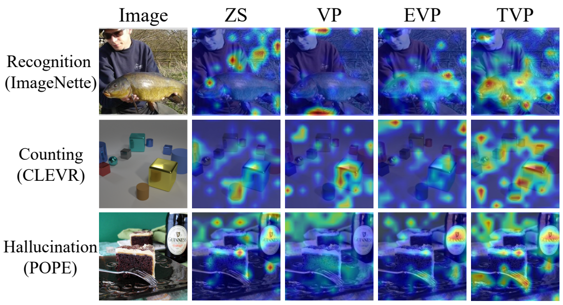


Figure 4. GradCAM [46] of VPGTrans [63] on 3 different tasks. TVP encourages the model to attend to task-related objects.

FCA	TSE	CIFAR-10	IN	SVHN	CLEVR	HM	POPE
✗	✗	0.28	0.81	17.45	8.64	0.09	0.90
✓	✗	3.81	2.73	20.84	10.83	0.99	2.33
✗	✓	1.91	3.36	24.68	10.14	0.31	2.15
✓	✓	**3.83**	**5.80**	**24.72**	**11.51**	**1.69**	**3.19**

Table 2. The average performance improvements with different combinations of FCA and TSE.

Prompt Width	5	10	20	30	40	50	80
MiniGPT-4	+2.84	+3.23	**+3.97**	+3.83	+2.71	+1.93	-3.03
InstructBLIP	+3.71	+5.42	+5.06	**+6.69**	+3.15	+2.91	-2.12
Ensemble	+4.66	+3.38	+5.45	**+7.75**	+5.87	+5.12	+2.67

Table 3. The average improvements on CIFAR-10 with prompts of different widths by TVP using different training models.

4.4. Ablation Studies

We conduct ablation studies on TVP to further verify our design. In the rest context, we only report average performance for analysis and detailed results are in Appendix B.

4.4.1 Strategies of FCA and TSE

We first examine the impacts of the proposed strategies, FCA and TSE, on the performance of visual prompts trained with MiniGPT-4. As shown in Tab. 2, individually applying either strategy can improve the average performance to some extent. When they are combined together, it can maximize the assistance of visual prompts for diverse models. This indicates that both task-agnostic prior knowledge motivating FCA and task-related feature extraction enhanced by TSE can contribute to the transferability of visual prompts. Visualization in Fig. 4 also suggests that TVP can help models better locate the objects beneficial for task completion.

4.4.2 Prompt Width

Prompt width, defining the number of parameters, could be critical for TVP's performance. The results on CIFAR-10 in Tab. 3 show that there is a trade-off between the number of learnable parameters and the scaled image size, and the performance of TVP reaches the peak at a moderate prompt width around 20-30, which validates our choice of 30.

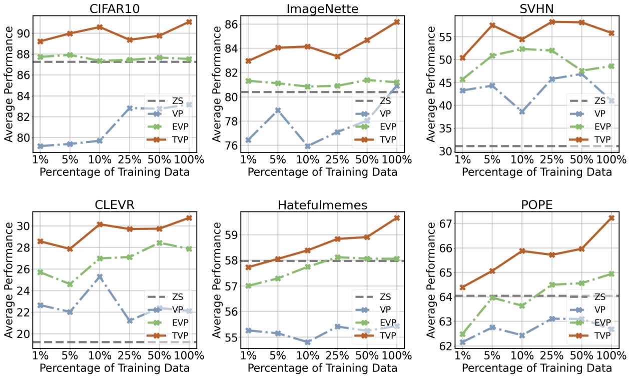


Figure 5. Curves of average performance as the training data scale changes. TVP can effectively enhance the performance of different models even with only 1% of the data, and its overall performance improves as the data size increases.

4.5. In-depth Analyses

We present several in-depth analyses below, to demonstrate the effectiveness of TVP under different conditions and validate its practicality in real scenarios.

4.5.1 Data Scale

Considering that the data available for adaptation is sometimes limited in practical scenarios, it's essential to study the impact of the training data size on the performance of the learned visual prompts. Following [24, 56], we examine the varying trends of different methods within a larger range from 1% to 100%. We plot the curves of the average performance of different models armed with the visual prompts trained using MiniGPT-4 as the amount of data changes in Fig. 5. As expected, the performance of TVP tends to improve as the data scale increases in general. The overall conclusion in Sec. 4.2 that TVP performs better than VP and EVP remains consistent regardless of data sizes. It is worth noting that TVP can enhance the overall performance of multiple models even when the data amount is only 1%, and the effect of visual prompts is sometimes close to that trained with the complete datasets. This further illustrates the effectiveness of our method and makes its application in real-world scenarios more promising.

4.5.2 Generalization across Datasets

It is worth investigating the generalization of trained visual prompts to further confirm the practicality of TVP in real scenarios. We take object recognition as an example. Following the common practice in prompt learning [27, 65], we apply the visual prompts generated with ensemble method on CIFAR-100, which has the most categories for common objects, to other datasets of the recognition task. As shown in Tab. 4, besides the best transferability on source dataset, TVP also gets the most or comparable improvements on other datasets, showing good generalization across diverse datasets within the same task.

Avg. Δ	C-100	C-10	IN	SVHN	Pet	Air	Food
VP	+2.87	+1.57	+1.24	-0.29	+2.87	**+0.97**	-1.04
EVP	+6.21	-0.11	+0.14	-2.84	+1.21	+0.75	-1.05
TVP	**+7.57**	**+4.10**	**+2.33**	**+1.57**	**+3.87**	+0.35	**+1.89**

Table 4. Cross-dataset generalization with visual prompts using the ensemble of MiniGPT-4 and InstructBLIP on CIFAR-100.

4.5.3 Robustness to Corruptions

We also test the robustness of visual prompts to common image corruptions [17]. The visual prompt generated by TVP on MiniGPT-4 gets an average improvement of 2.30% on CIFAR-10-C while both VP and EVP result in performance degradation of -6.62% and -3.87%. Results on corrupted datasets are in Appendix B.2 due to space limit.

4.6. Discussion on Computational Efficiency

We further compare the performance and efficiency of TVP with those of other fine-tuning methods and visual prompting methods, to support the motivation of the proposed problem and the corresponding solution of TVP. Due to the page limit, we present detailed discussion in Appendix C.

5. Conclusion

In this paper, we propose to optimize a set of *shared parameters* for diverse MLLMs to adapt them to downstream tasks in a resource-friendly and flexible manner, which can avoid the computation and storage overheads with model-specific fine-tuning. Concretely, we introduce **Transferable Visual Prompting** to boost the performance of a group of models by adopting visual prompts as the shared parameters and improving their transferability with one-time training on only one model. Existing methods of visual prompting usually fail to enhance unseen models by satisfying margins due to feature corruption. We address this with two key strategies, Feature Consistency Alignment and Task Semantics Enrichment, which maintain the inner prior knowledge of large-scale pre-trained models and strengthen the task-related features extracted by models. Through extensive experiments on 10 datasets of diverse tasks from recognition and counting to multimodal reasoning and hallucination correction, we demonstrate the effectiveness of the proposed TVP to promote different models simultaneously.

Acknowledgement

This work was supported by the National Key Research and Development Program of China (No. 2020AAA0106302), NSFC Projects (Nos. 92370124, 62350080, 92248303, 62276149, U2341228, 62061136001, 62076147), BNRist (BNR2022RC01006), Tsinghua Institute for Guo Qiang, and the High Performance Computing Center, Tsinghua University. J. Zhu was also supported by the XPlorer Prize.

Y. Dong was also supported by the China National Postdoctoral Program for Innovative Talents and Shuimu Tsinghua Scholar Program.

References

[1] Visualglm-6b. https://github.com/THUDM/VisualGLM-6B/, 2023. 5, 7, 1

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 23716–23736, 2022. 1, 2

[3] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 1, 2, 3, 4, 6, 5

[4] Yijun Bian and Huanhuan Chen. When does diversity help generalization in classification ensembles? *IEEE Transactions on Cybernetics*, 52(9):9059–9075, 2021. 7

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *Computer Vision – ECCV 2014*, pages 446–461, 2014. 7, 1

[7] Huanran Chen, Yichi Zhang, Yinpeng Dong, and Jun Zhu. Rethinking model ensemble in transfer-based adversarial attacks. *arXiv preprint arXiv:2303.09105*, 2023. 7

[8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 1

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023. 1, 2, 3, 4, 5, 6, 7

[10] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9185–9193, 2018. 3, 7

[11] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4312–4321, 2019. 2

[12] Yinpeng Dong, Huanran Chen, Jiawei Chen, Zhengwei Fang, Xiao Yang, Yichi Zhang, Yu Tian, Hang Su, and Jun Zhu. How robust is google's bard to adversarial image attacks? *arXiv preprint arXiv:2309.11751*, 2023. 7

[13] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022. 7

[14] Gamaleldin F. Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. In *International Conference on Learning Representations*, 2019. 3, 4

[15] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 1, 2, 5

[16] Anisha Gunjal, Jihan Yin, and Erhan Bas. Detecting and preventing hallucinations in large vision language models. *arXiv preprint arXiv:2308.06394*, 2023. 2

[17] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 8, 2, 4

[18] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799, 2019. 2, 3

[19] Jeremy Howard. imagenette. "https://github.com/fastai/imagenette/". 5, 1

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 3

[21] Wenbo Hu, Yifan Xu, Y Li, W Li, Z Chen, and Z Tu. Bliva: A simple multimodal llm for better handling of text-rich visual questions. *arXiv preprint arXiv:2308.09936*, 2023. 1, 3, 5

[22] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Weiming Zhang, Feifei Wang, Gang Hua, and Nenghai Yu. Diversity-aware meta visual prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10878–10887, 2023. 3

[23] C. V. Jawahar, A. Zisserman, A. Vedaldi, and O. M. Parkhi. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3498–3505, 2012. 7, 1

[24] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision – ECCV 2022*, pages 709–727, 2022. 2, 3, 6, 8

[25] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2910, 2017. 5, 1

[26] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14829–14838, 2022. 5

[27] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19113–19122, 2023. 3, 8

[28] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15190–15200, 2023. 3, 5

[29] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, pages 2611–2624, 2020. 5, 1

[30] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9619–9628, 2021. 5

[31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5, 7, 1

[32] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 1, 2

[33] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large-language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*, 2023. 2

[34] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1, 2, 5

[35] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 2, 5, 1

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, 2014. 1

[37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023. 1, 2, 4

[38] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, 2022. 2, 3

[39] Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*, 2023. 2

[40] Gen Luo, Yiyi Zhou, Tianhe Ren, Shengxin Chen, Xiaoshuai Sun, and Rongrong Ji. Cheap and quick: Efficient vision-language instruction tuning for large language models. In *Advances in Neural Information Processing Systems*, pages 29615–29627, 2023. 3

[41] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 7, 1

[42] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 1, 5

[43] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24224–24235, 2023. 4, 6

[44] 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 *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021. 2, 3, 5

[45] Konstantinos I Roumeliotis and Nikolaos D Tselikas. Chatgpt and open-ai models: A preliminary review. *Future Internet*, 15(6):192, 2023. 1

[46] 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. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 7

[47] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? In *International Conference on Learning Representations*, 2022. 5

[48] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355*, 2023. 2

[49] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 3, 7

[50] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[51] 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 *Proceedings of the 37th International Conference on Machine Learning*, pages 9614–9624, 2020. 3

[52] Naonori Ueda and Ryohei Nakano. Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, pages 90–95. IEEE, 1996. 7

[53] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 4

[54] Twan Van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017. 5

[55] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. 1

[56] 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. *arXiv preprint arXiv:2212.10556*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[57] Qiong Wu, Wei Yu, Yiyi Zhou, Shubin Huang, Xiaoshuai Sun, and Rongrong Ji. Parameter and computation efficient transfer learning for vision-language pre-trained models. In *Advances in Neural Information Processing Systems*, pages 41034–41050, 2023. 3

[58] Yixin Wu, Rui Wen, Michael Backes, Pascal Berrang, Mathias Humbert, Yun Shen, and Yang Zhang. Quantifying privacy risks of prompts in visual prompt learning. *arXiv preprint arXiv:2310.11970*, 2023. 2, 3

[59] Peng Xu, Wenqi Shao, Kaipeng Zhang, Peng Gao, Shuo Liu, Meng Lei, Fanqing Meng, Siyuan Huang, Yu Qiao, and Ping Luo. Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models. *arXiv preprint arXiv:2306.09265*, 2023. 1, 2

[60] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023. 2

[61] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022. 1

[62] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*, 2023. 1

[63] Ao Zhang, Hao Fei, Yuan Yao, Wei Ji, Li Li, Zhiyuan Liu, and Tat-Seng Chua. Transfer visual prompt generator across llms. In *Advances in Neural Information Processing Systems*, 2023. 2, 5, 7, 1

[64] Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. Mmicl: Empowering vision-language model with multi-modal in-context learning. *arXiv preprint arXiv:2309.07915*, 2023. 2

[65] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16816–16825, 2022. 3, 8

[66] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3

[67] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2023. 5

[68] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *Computer Vision – ECCV 2018*, pages 452–467, 2018. 2, 3, 4

[69] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 2, 3, 4, 5, 6, 7

Exploring the Transferability of Visual Prompting for Multimodal Large Language Models

Supplementary Material

A. Detailed Experimental Settings

Here we describe the detailed experimental settings to guarantee the reproducibility. All experiments are conducted on NVIDIA A100-80GB GPUs.

A.1. Datasets

In this work, we adopt 10 datasets in total to validate the effectiveness of the proposed TVP. We categorize them into 4 visual or multimodal tasks and we will introduce them respectively.

Object Recognition. Following [2, 9], we take close-ended evaluation for recognition, restricting the vocabulary to the category names of the datasets. To be specific, The prompt given to the models is "*This is a photo of a*" and the target for text completion will be the ground-truth label in text. We concatenate each candidate category after the prompt and select the one with maximum log-likelihood as the prediction. The description for TSE is in the template of "*This is a photo of a* $\{$*ground-truth label*$\}$".

We take 7 datasets for this task, including CIFAR-10, CIFAR-100 [31], ImageNette [19] (a subset of ImageNet), which are commonly used for image classification, and SVHN [42], Oxford Pets [23], FGVCAircraft [41] (manufacturer level), Food101 [6], which are popular datasets for fine-grained classification in specific domains. By default, we take the `train` split for training, `val` split for validation and `test` split for testing as provided in the dataset. If `val` split is not provided, we sample a certain proportion for validation.

Object Counting. We take CLEVR [25] as an example. Unlike recognition, we take an open-ended evaluation for object counting. We ask the models "*How many objects are there in this image? Answer with a single number.*" and generate the response with do_sample set False and other parameters as default. We evaluate the response as correct or not by checking whether the answer of number appears in it. The corresponding description for TSE is "*There are* $\{$*number*$\}$ *objects in this image*". We take the `train` split for training and sample 10% and 20% out of `val` split for validation and testing respectively.

Multimodal Reasoning. We take Hatefulmemes [29] for multimodal reasoning, which ask the models to decide whether the text on the meme and the visual content combined together convey hatred. Following [9], the prompt is "*This is an image with "*$\{\}$*" written on it. Is it hateful?*", and we take the ranking method used for recognition here with "Yes" and "No" as labels. We use the normalized log-likelihood to calculate ROC AUC score. The description for TSE is "*This is (not) hateful*". We take 90% of `train` split for training, the rest 10% for validation and `dev` split for testing.

Hallucination Correction. We take POPE [35], which ask the models whether there is a certain object in the image or not to evaluate their hallucination. The prompt given to the model is consistent with the default setting in official code, as "*Is there a "*$\{\}$*" in the image?*" and we also take "Yes" and "No" as labels. The description for TSE is in the template of "*There are* $\{$*object list*$\}$ *in the image.*" based on the annotations from MSCOCO [36]. We take the public release split (3000 samples) for testing and generate another dataset of 12000 samples for training and validation with 90%-10% random split. In this work, we only adopt datasets built with adversarial negative sampling strategy to challenge the models at utmost.

A.2. Models

We select 6 modern MLLMs for experiments. These models have different implementations, for instance BLIVA [21] uses two projection layers to better address visual-text alignment and VPGTrans [63] introduces the concept of visual prompt generator to transfer pre-trained visual encoder across different LLMs. We clone the official codebase of different models and unify the interface for training and inference to better incorporate different models.

The detailed configuration for them mainly involves the the choices of LLMs. We take Vicuna-7B-v0 [8] for MiniGPT-4 [69], BLIVA and VPGTrans, Vicuna-7B-v1.1 for InstructBLIP [9], Flan-T5-XL [55] for BLIP2 [34], and ChatGLM-6B [61] for VisualGLM-6B [1]. For visual encoders, these MLLMs share the structure of ViT-G/14, but with different projection layers and training paradigms, which guarantee the model diversity. These models can be deployed conveniently following the official instructions provided in the repositories.

As for the CLIP's visual encoder for TSE, we use ViT-B/32, a lightweight and popular version for studying CLIP. Since TSE is to introduce extra task knowledge, it does not need to have the same visual encoder as MLLMs.

A.3. Hyperparameters

We introduce the setting of hyperparameters in this work. The design of visual prompts has been introduced in Sec. 3.1. The batch size for training is 16. The learning rate γ in Eq. (7) is 10 by default. The maximal number of training epochs is 10 with cosine scheduler following [56].

For the weights for the proposed FCA and TSE loss terms, we set them optimal by searching within {0.0005, 0.001, 0.003, 0.005, 0.008} and {0.0001, 0.0005, 0.001} respectively on validation set, while keeping other hyperparameters consistent with baselines.

B. Additional Results

B.1. Results on Other Datasets

Besides the 6 datasets displayed in the main paper, we also validate the effectiveness of our method on 4 commonly used classification datasets and demonstrate the results in Tab. 6.

Apart from the coarse-grained classification dataset CIFAR-100, the zero-shot performance of modern MLLMs on these fine-grained datasets in specific domains is far from satisfactory, further emphasizing the necessities for adapting MLLMs to downstream tasks.

The observations and conclusions in Sec. 4.2 remain consistent. We can see that visual prompts generated by TVP on a single model (MiniGPT-4 or InstructBLIP) bring the most significant improvements to 6 models. Moreover, by ensembling two models for training visual prompts, the performance is further boosted to higher levels.

B.2. Results on Corrupted Datasets

Robustness has been a crucial issue for deep neural network, concerning the stability of model in applications. It is natural to evaluate the robustness of visual prompts to image common corruptions [17]. We examine the performance of visual prompts generated by MiniGPT-4 on corrupted datasets like CIFAR-10-C and ImageNette-C. We set the severity level as 3 and test with 15 corruptions. We use the official release of CIFAR-10-C and the official code[1] to generate corresponding corrupted dataset for ImageNette.

The results are shown in Tab. 9. Visual prompts generated by VP and EVP cannot effectively improve the 6 models on average under the corruptions imposed to CIFAR-10, while TVP can still bring 2.30% and 3.09% on CIFAR-10-C and ImageNette-C respectively. The results indicate that the consolidation of task-agnostic representations and enhancement of task-related semantics by TVP effectively strengthen the robustness of learned visual prompts to common image corruptions.

B.3. Detailed Results for Ablations and Analyses

Due to space limit, we only report the average performance or average delta in performance for ablation studies in Sec. 4.4 and in-depth analyses in Sec. 4.5. Here, we display the results for each setting and each model in detail. Detailed results for Tab. 2 are in Tab. 10, those for Tab. 3 are in Tab. 7, those for Fig. 5 are in Tab. 11 and those for Tab. 4 are in Tab. 8.

C. Discussion on Computational Efficiency

As we target on efficient adaptation for diverse MLLMs rather than fine-tuning each of them respectively, we here discuss the computational efficiency of the proposed TVP.

C.1. Comparison with Fine-tuning Methods

We conduct additional experiments on an A100-80G GPU with half precision and the same batch size as TVP. If the training exceeds GPU memory (e.g., BLIVA), we adopt gradient accumulation. Here we use CIFAR-10 and the prompts trained on InstructBLIP to compare with full fine-tuning and LoRA. Results are displayed in Tab. 5. Though FFT and LoRA have moderately higher accuracy than TVP due to much larger numbers of trainable parameters (≥4B for FFT, ≥8M for LoRA and ∼70K for TVP), TVP has the minimal computation overhead, which is reflected in the smallest memory demand and the shortest average training time. When the computation resources are limited to fine-tuning, off-the-shelf visual prompts trained by TVP are expected to achieve black-box adaptation with no cost. This supports the motivation of our method.

	InstructBLIP			BLIP2			MiniGPT-4			BLIVA		
	FFT	LoRA	TVP	FFT	LoRA	TVP	FFT	LoRA	TVP	FFT	LoRA	TVP
Acc (%)	99.16	98.78	98.07	99.09	98.08	96.02	99.27	95.18	91.69	99.07	98.14	97.78
Mem. (GB)	63.5	33.8	31.1	36.9	21.8	$9.2^{\dagger}$	62.4	35.6	$18.3^{\dagger}$	66.5	55.2	$18.5^{\dagger}$
Time (min)	30	26	27	28	26	0	29	25	0	118	92	0

Table 5. Comparison of performance, memory costs and training time with fine-tuning methods. gray for black-box models, † for inference mode, since they need no training for TVP.

C.2. Comparison with Baseline Visual Prompting

Compared to the baselines, VP and EVP, TVP demands additional forward passes through vision encoders. Taking MiniGPT-4 for example, VP and EVP need one forward pass in each iteration and take around 820GFLOPs. For TVP, the combination of FCA and TSE demands an extra forward pass through the MLLM's visual encoder (∼260GFLOPs, FCA) and another forward pass through CLIP (∼7GFLOPs, TSE). While extra computation for TSE is negligible, FCA brings around 32% more computation overloads, with a similar increase in training time. However, the original visual features only need to be computed once, thus the cost for FCA can be distributed to each epoch and will only bring around 3% extra computations when trained for 10 epochs, which is acceptable. The computation overheads can be further alleviated in the future.

[1] https://github.com/hendrycks/robustness

Recognition: CIFAR-100		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.∆
	Clean	61.85	58.41	60.65	58.00	56.34	12.71	0.00
MiniGPT-4	VP [3]	63.54*	44.40	60.05	59.93	53.36	12.15	-2.42
	EVP [56]	71.05*	48.91	56.43	59.23	56.44	20.10	+0.70
	TVP (ours)	**75.36***	65.10	64.15	57.84	53.58	**21.34**	**+4.90**
InstructBLIP	VP [3]	60.65	76.16*	58.60	58.32	58.40	9.47	+2.27
	EVP [56]	62.24	**78.68***	61.66	57.37	59.86	12.13	+4.00
	TVP (ours)	63.92	77.92*	63.72	62.62	56.09	12.97	**+4.88**
Ensemble	VP [3]	65.48*	71.77*	63.48	60.25	55.04	9.15	+2.87
	EVP [56]	70.13*	74.89*	62.40	62.07	60.76	14.96	+6.21
	TVP (ours)	73.33*	77.62*	**64.19**	**62.79**	**62.18**	13.26	**+7.57**

Recognition: Pet37		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.∆
	Clean	30.50	27.23	11.53	16.52	22.21	31.07	0.00
MiniGPT-4	VP [3]	42.38*	33.69	11.80	23.14	25.81	29.46	+4.54
	EVP [56]	56.67*	30.44	13.22	22.40	27.91	28.37	+6.66
	TVP (ours)	**59.53***	39.00	**16.57**	25.27	30.53	29.35	**+10.20**
InstructBLIP	VP [3]	40.23	37.80*	13.27	17.83	29.71	29.54	+4.89
	EVP [56]	40.83	65.25*	12.16	17.63	31.70	30.36	+9.81
	TVP (ours)	41.05	**66.86***	14.28	22.27	42.95	30.44	**+13.13**
Ensemble	VP [3]	46.77*	43.80*	15.10	22.13	32.73	**30.69**	+8.69
	EVP [56]	56.99*	66.31*	13.55	15.10	32.76	29.60	+12.54
	TVP (ours)	51.35*	61.60*	13.87	**26.30**	**48.11**	28.02	**+15.03**

Recognition: Aircraft		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.∆
	Clean	8.55	10.26	6.54	14.34	8.19	4.05	0.00
MiniGPT-4	VP [3]	30.15*	8.67	6.93	14.97	11.13	4.02	+3.99
	EVP [56]	32.52*	9.36	6.42	17.64	11.25	4.02	+4.88
	TVP (ours)	**33.99***	9.81	**7.41**	20.76	7.20	4.02	**+5.21**
InstructBLIP	VP [3]	12.90	16.92*	4.59	12.18	8.97	4.02	+1.27
	EVP [56]	22.92	31.35*	5.28	**23.97**	11.04	4.02	+7.78
	TVP (ours)	30.48	**36.03***	4.02	20.76	11.85	4.04	**+9.21**
Ensemble	VP [3]	28.68*	25.50*	7.29	13.02	11.52	4.05	+6.35
	EVP [56]	26.76*	26.34*	6.45	17.13	12.22	4.02	+6.83
	TVP (ours)	30.27*	24.84*	4.02	23.10	**24.00**	**4.59**	**+9.82**

Recognition: Food101		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.∆
	Clean	32.99	28.99	47.29	30.42	36.08	5.90	0.00
MiniGPT-4	VP [3]	50.14*	32.08	34.10	23.49	31.92	4.67	-0.88
	EVP [56]	63.72*	30.93	45.43	27.64	33.74	3.88	+3.95
	TVP (ours)	**64.16***	37.66	48.95	29.43	36.36	5.54	**+6.74**
InstructBLIP	VP [3]	19.68	41.23*	33.70	26.85	36.28	**8.71**	-2.54
	EVP [56]	37.47	64.95*	48.87	31.25	43.37	3.84	+8.01
	TVP (ours)	38.49	**68.51***	48.55	31.13	44.75	6.02	**+9.46**
Ensemble	VP [3]	53.03*	59.21*	47.29	27.68	**46.57**	6.42	+9.75
	EVP [56]	63.48*	66.50*	48.20	27.49	26.46	4.12	+9.10
	TVP (ours)	63.92*	66.22*	**51.80**	**34.61**	44.44	5.47	**+14.13**

Table 6. Results on 4 more datasets of object recognition. Visual prompts are trained on MiniGPT-4, InstructBLIP and their ensemble with different methods, and further tested on 6 modern MLLMs. Top-1 accuracy (%) is reported.

Trained on	Prompt Wid.	MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg. ∆
MiniGPT-4	5	94.29	84.17	89.40	91.60	90.94	90.15	+2.84
	10	96.00	84.03	93.17	91.82	92.73	85.17	+3.23
	20	96.82	91.26	86.68	88.49	93.71	90.39	+3.97
	40	95.70	89.44	88.49	87.68	89.69	88.78	+2.71
	50	96.77	87.29	87.62	88.26	88.05	87.15	+1.93
	80	94.21	78.21	86.41	84.02	85.91	76.62	-3.03
InstructBLIP	5	89.73	96.41	85.08	91.95	93.64	88.97	+3.71
	10	88.03	97.06	92.47	91.89	94.90	91.72	+5.42
	20	88.04	98.04	82.78	93.96	97.95	93.13	+5.06
	40	85.16	98.24	86.37	89.21	89.88	93.55	+3.15
	50	84.52	97.81	93.58	87.33	91.50	86.28	+2.91
	80	82.38	94.75	83.15	81.85	88.31	80.39	-2.12
Ensemble	5	91.64	94.53	94.72	88.97	94.71	86.94	+4.66
	10	95.08	95.65	92.73	87.58	94.00	78.77	+3.38
	20	95.19	96.55	93.37	90.59	96.23	84.33	+5.45
	40	97.73	97.41	84.59	91.52	97.17	90.34	+5.87
	50	96.60	97.31	88.85	88.20	95.02	88.31	+5.12
	80	92.01	95.99	84.14	81.64	94.14	91.62	+2.67

Table 7. Detaile results for the ablation study about the impact of prompt width on the performance of TVP on CIFAR-10 in Tab. 3.

Datasets	Model	MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg. ∆
CIFAR-10	VP [3]	87.97	94.20	82.59	88.64	89.13	90.47	+1.57
	EVP [56]	81.89	89.44	82.19	90.11	84.15	95.14	-0.11
	TVP (ours)	88.80	94.38	89.92	91.53	89.91	93.63	+4.10
ImageNette	VP [3]	84.25	74.70	92.00	81.50	84.59	72.79	+1.24
	EVP [56]	83.18	77.38	87.39	83.31	79.82	72.18	+0.14
	TVP (ours)	88.36	77.20	94.01	82.78	80.15	73.86	+2.33
SVHN	VP [3]	39.42	30.00	32.87	33.57	27.49	21.62	-0.29
	EVP [56]	35.34	24.27	33.20	34.63	21.97	20.24	-2.84
	TVP (ours)	41.98	30.02	26.88	39.49	31.55	26.23	+1.57
Pet37	VP [3]	34.15	33.01	14.64	20.82	25.40	28.26	+2.87
	EVP [56]	33.39	31.34	9.46	16.54	29.35	26.25	+1.21
	TVP (ours)	38.05	30.01	14.99	23.58	28.56	27.12	+3.87
Aircraft	VP [3]	16.50	7.74	5.88	13.08	10.56	4.02	+0.97
	EVP [56]	19.05	9.24	4.05	11.01	9.09	4.02	+0.75
	TVP (ours)	15.15	11.58	4.05	10.68	8.58	4.02	+0.35
Food101	VP [3]	31.45	27.33	40.48	29.19	41.70	5.31	-1.04
	EVP [56]	33.03	33.27	37.90	26.85	40.24	4.08	-1.05
	TVP (ours)	37.47	43.84	38.89	28.95	38.93	4.95	+1.89

Table 8. Detailed results for the analysis on the generalization of TVP using ensemble across diverse recognition datasets in Tab. 4.

Corruption Types	Fog	JPEG Compression	Zoom Blur	Glass Blur	Shot Noise	Defocus Blur	Elastic Transform	Frost	Brightness	Snow	Gaussian noise	Motion Blur	Contrast	Impulse Noise	Pixelate	Avg.Δ
Clean	85.73	69.41	82.87	70.72	71.13	85.93	82.87	83.12	86.69	82.69	65.85	80.24	86.72	79.38	81.75	0.00
VP [3]	81.96	55.36	79.68	58.69	60.05	82.67	80.42	77.66	83.59	78.71	52.67	74.12	82.86	71.06	76.26	-6.62
EVP [56]	85.31	57.10	82.51	58.49	63.76	85.93	83.90	80.46	86.78	82.33	56.20	76.57	85.96	75.04	76.67	-3.87
TVP (ours)	89.46	68.65	87.46	67.95	71.92	89.95	87.85	85.58	90.82	86.76	66.12	83.01	90.02	80.71	83.34	**+2.30**

(a) Average performance under different common corruptions at level 3 on CIFAR-10 with visual prompts generated on MiniGPT-4.

Corruption Types	Fog	JPEG Compression	Zoom Blur	Glass Blur	Shot Noise	Defocus Blur	Elastic Transform	Frost	Brightness	Snow	Gaussian noise	Motion Blur	Contrast	Impulse Noise	Pixelate	Avg.Δ
Clean	79.15	80.50	69.91	71.26	76.77	75.92	72.24	74.36	79.71	76.23	76.96	76.52	79.44	76.98	81.15	0.00
VP [3]	78.76	80.62	69.40	71.77	78.09	76.36	72.82	73.78	80.72	75.57	78.31	76.74	78.92	78.42	80.54	+0.25
EVP [56]	79.33	81.58	67.01	70.95	78.52	76.74	73.44	73.38	82.07	75.68	78.76	76.78	78.46	78.65	82.18	+0.43
TVP (ours)	82.53	84.20	70.31	73.25	81.00	80.32	74.76	76.51	83.58	79.19	81.07	79.85	82.07	81.05	83.79	**+3.09**

(b) Average performance under different common corruptions at level 3 on ImageNette with visual prompts generated on MiniGPT-4.

Table 9. Average performance under common corruptions [17] of different methods on CIFAR-10 and ImageNette. Visual prompts generated by the proposed TVP still lead to the most significant improvements, showing better robustness to common corruptions.

FCA	TSE	MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
✗	✗	97.97	84.57	83.39	86.93	86.45	85.92	0.28
✓	✗	97.95	86.97	90.58	90.94	92.18	87.82	3.82
✗	✓	97.94	86.93	85.74	90.32	92.78	81.30	1.91
✓	✓	98.33	92.82	91.68	88.70	87.48	87.53	3.83

(a) CIFAR-10

FCA	TSE	MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
✗	✗	96.79	68.15	91.36	79.08	75.82	76.05	0.81
✓	✗	95.87	75.87	96.51	82.24	78.19	70.11	2.73
✗	✓	97.81	67.59	91.64	78.35	84.23	82.93	3.36
✓	✓	97.71	78.34	94.98	86.34	84.51	75.34	5.80

(b) ImageNette

FCA	TSE	MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
✗	✗	74.24	41.59	48.87	57.61	36.11	33.12	17.48
✓	✗	74.81	56.69	52.98	51.96	50.35	24.93	20.84
✗	✓	81.39	53.41	50.99	59.46	56.60	32.91	24.68
✓	✓	75.17	54.32	61.95	51.10	60.28	32.17	24.72

(c) SVHN

FCA	TSE	MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
✗	✗	52.17	39.03	20.17	8.00	34.23	13.60	8.64
✓	✗	50.57	36.03	22.30	20.93	32.53	18.03	10.83
✗	✓	54.07	31.33	16.60	20.33	32.87	21.07	10.15
✓	✓	51.00	42.90	22.07	19.50	36.00	13.00	11.51

(d) CLEVR

FCA	TSE	MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
✗	✗	57.58	60.66	55.34	56.87	60.64	57.27	0.09
✓	✗	56.99	63.24	54.15	58.82	63.00	57.52	0.99
✗	✓	58.31	61.65	55.20	56.43	61.66	56.40	0.31
✓	✓	56.93	62.38	56.20	60.19	64.09	58.15	1.69

(e) Hatefulmemes

FCA	TSE	MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.Δ
✗	✗	68.06	69.80	50.00	61.07	71.33	69.40	0.90
✓	✗	69.60	74.00	50.13	59.47	74.80	70.27	2.33
✗	✓	69.00	75.13	49.93	61.27	72.40	69.47	2.15
✓	✓	68.73	75.13	51.40	64.47	72.67	71.00	3.19

(f) POPE

Table 10. Detailed results for the ablation study on different combinations of FCA and TSE in Tab. 2.

Model		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.
CIFAR-10								
VP [3]	1%	83.08	74.95	79.80	80.35	81.67	75.28	79.19
	5%	82.06	76.07	79.20	80.97	80.07	77.98	79.39
	10%	84.29	77.58	80.00	80.99	81.96	73.38	79.70
	25%	90.97	80.35	82.42	81.34	84.63	77.28	82.83
	50%	90.53	81.14	77.45	83.29	84.31	79.84	82.76
EVP [56]	1%	97.11	86.67	83.74	87.06	89.23	82.60	87.74
	5%	97.85	85.56	83.06	86.64	87.92	86.49	87.92
	10%	97.93	83.15	83.00	88.81	84.78	86.49	87.36
	25%	98.24	85.16	82.29	87.53	85.71	85.72	87.44
	50%	98.00	84.07	83.86	87.39	86.66	86.01	87.67
TVP (ours)	1%	97.80	86.04	85.27	87.90	88.65	89.69	89.23
	5%	97.24	87.79	88.32	87.28	90.09	89.14	89.98
	10%	97.85	87.69	90.82	87.36	91.97	87.87	90.59
	25%	98.23	84.86	89.20	87.61	89.96	86.34	89.37
	50%	97.68	87.59	93.57	86.33	88.12	85.25	89.76
SVHN								
VP [3]	1%	67.63	47.78	43.42	36.19	38.35	26.21	43.26
	5%	66.96	38.52	47.80	44.07	33.59	34.96	44.32
	10%	81.06	35.09	21.37	40.87	32.99	20.47	38.64
	25%	58.26	41.86	50.85	58.74	35.33	29.64	45.78
	50%	73.98	50.65	46.92	45.37	43.52	20.82	46.88
EVP [56]	1%	75.05	31.80	52.44	42.97	44.37	27.53	45.69
	5%	77.55	44.95	48.97	57.99	39.28	36.55	50.88
	10%	80.57	42.22	61.53	53.69	53.69	22.24	52.32
	25%	76.78	44.75	51.18	59.47	46.58	33.14	51.98
	50%	75.98	41.35	47.41	55.47	34.93	30.37	47.59
TVP (ours)	1%	79.68	38.39	56.55	46.34	48.16	33.36	50.41
	5%	85.53	48.70	64.67	58.15	48.28	39.79	57.52
	10%	80.52	47.70	61.37	56.63	45.20	35.24	54.44
	25%	80.70	51.20	62.89	59.59	59.93	35.20	58.25
	50%	82.58	51.23	65.77	59.79	53.57	35.91	58.14
Hatefulmemes								
VP [3]	1%	57.04	56.22	60.26	51.79	49.67	56.58	55.26
	5%	57.08	56.44	54.53	53.84	54.68	54.31	55.15
	10%	57.50	55.73	57.94	55.40	53.38	48.88	54.81
	25%	60.16	58.05	57.49	55.54	53.21	48.01	55.41
	50%	57.27	55.97	53.82	54.38	54.70	55.34	55.25
EVP [56]	1%	48.33	60.13	55.84	61.23	59.54	56.94	57.00
	5%	54.29	62.50	52.35	55.90	61.70	57.02	57.29
	10%	55.02	62.45	50.84	57.80	63.04	57.31	57.74
	25%	58.26	59.88	54.90	54.60	62.16	58.92	58.12
	50%	57.42	61.64	57.07	52.98	61.64	57.64	58.07
TVP (ours)	1%	51.60	61.32	55.06	59.24	61.36	57.80	57.73
	5%	53.65	62.68	53.30	58.00	62.48	58.22	58.06
	10%	54.75	61.40	54.52	59.10	62.53	58.02	58.39
	25%	59.83	61.81	53.86	57.17	62.88	57.45	58.83
	50%	55.34	62.45	53.51	59.42	64.65	58.06	58.91

Model		MiniGPT-4	InstructBLIP	BLIP2	VPGTrans	BLIVA	VisualGLM	Avg.
ImageNette								
VP [3]	1%	77.58	64.08	93.50	77.73	73.53	72.25	76.45
	5%	81.91	64.28	91.80	81.40	80.48	73.45	78.89
	10%	78.22	65.17	95.13	77.12	67.52	72.43	75.93
	25%	82.42	60.25	93.12	79.29	71.41	76.10	77.10
	50%	82.01	62.14	92.64	76.66	77.83	76.94	78.04
EVP [56]	1%	93.01	62.00	94.96	74.07	83.38	80.57	81.33
	5%	97.61	62.80	89.10	77.10	79.88	80.25	81.12
	10%	98.00	71.87	90.37	72.97	76.05	75.85	80.85
	25%	97.44	76.08	89.15	63.99	84.58	74.24	80.91
	50%	97.40	62.06	85.12	74.70	82.52	86.57	81.39
TVP (ours)	1%	96.13	72.08	89.30	79.06	90.78	70.45	82.97
	5%	97.61	62.70	91.80	76.87	83.46	91.95	84.07
	10%	97.20	72.25	94.70	83.21	82.93	74.68	84.16
	25%	97.63	72.48	89.86	85.96	80.46	73.71	83.35
	50%	98.09	73.99	90.29	84.66	86.09	75.03	84.69
CLEVR								
VP [3]	1%	33.17	31.57	12.73	12.73	32.87	12.87	22.66
	5%	38.37	27.57	12.83	21.63	19.23	12.57	22.03
	10%	40.13	36.00	23.33	10.97	28.67	12.63	25.29
	25%	39.81	26.44	12.95	12.50	21.84	13.77	21.22
	50%	39.30	28.80	12.77	12.00	28.63	12.83	22.39
EVP [56]	1%	47.03	35.53	15.47	11.37	31.80	13.03	25.71
	5%	25.07	35.97	27.93	9.77	34.60	14.30	24.61
	10%	49.80	34.57	16.43	15.60	32.40	13.10	26.98
	25%	44.70	32.43	20.67	16.40	35.57	12.90	27.11
	50%	53.70	42.60	20.10	7.47	33.87	12.90	28.44
TVP (ours)	1%	45.60	37.70	15.13	25.50	34.70	12.77	28.57
	5%	43.10	17.50	26.60	15.03	41.80	23.17	27.87
	10%	48.77	45.93	14.53	20.60	37.67	13.40	30.15
	25%	47.13	42.77	19.80	21.13	34.53	12.87	29.71
	50%	47.37	42.37	24.43	15.70	35.27	13.30	29.74
POPE								
VP [3]	1%	54.67	68.13	49.87	58.07	72.53	69.67	62.16
	5%	51.53	66.87	50.00	62.87	73.73	71.53	62.76
	10%	52.73	70.67	50.00	59.07	73.00	69.13	62.43
	25%	53.93	70.73	49.87	59.27	73.27	71.60	63.11
	50%	51.00	71.27	50.00	63.07	73.93	69.33	63.10
EVP [56]	1%	51.60	74.00	50.00	58.67	72.53	68.13	62.49
	5%	60.73	67.60	50.04	61.24	74.77	69.47	63.97
	10%	64.73	65.33	50.00	59.67	71.67	70.47	63.65
	25%	58.73	73.93	50.00	60.67	74.20	69.47	64.50
	50%	61.36	72.13	49.95	61.27	72.69	69.98	64.56
TVP (ours)	1%	61.00	71.47	50.13	59.00	75.00	69.80	64.40
	5%	62.73	69.47	50.00	65.20	72.47	70.53	65.07
	10%	59.87	75.47	49.87	65.13	74.27	70.67	65.88
	25%	71.13	69.93	49.60	60.60	72.67	70.40	65.72
	50%	70.47	68.67	49.87	61.00	75.47	70.33	65.97

Table 11. Detailed results for the analysis on the impact from different training data scales in Fig. 5.