Aggregate-and-Adapt Natural Language Prompts for Downstream Generalization of CLIP

Chen Huang, Skyler Seto, Samira Abnar, David Grangier, Navdeep Jaitly & Josh Susskind
Apple
{chen-huang,sseto,abnar,grangier,njaitly,jsusskind}@apple.com

Abstract

Large pretrained vision-language models like CLIP have shown promising generalization capability, but may struggle in specialized domains (*e.g.*, satellite imagery) or fine-grained classification (*e.g.*, car models) where the visual concepts are unseen or under-represented during pretraining. Prompt learning offers a parameter-efficient finetuning framework that can adapt CLIP to downstream tasks even when limited annotation data are available. In this paper, we improve prompt learning by distilling the textual knowledge from natural language prompts (either human- or LLM-generated) to provide rich priors for those under-represented concepts. We first obtain a prompt "summary" aligned to each input image via a learned prompt aggregator. Then we jointly train a prompt generator, optimized to produce a prompt embedding that stays close to the aggregated summary while minimizing task loss at the same time. We dub such prompt embedding as **A**ggregate-and-**A**dapted **P**rompt **E**mbedding (**AAPE**). AAPE is shown to be able to generalize to different downstream data distributions and tasks, including vision-language understanding tasks (*e.g.*, few-shot classification, VQA) and generation tasks (image captioning) where AAPE achieves competitive performance. We also show AAPE is particularly helpful to handle non-canonical and OOD examples. Furthermore, AAPE learning eliminates LLM-based inference cost as required by baselines, and scales better with data and LLM model size.

1 Introduction

Most existing vision-language tasks rely on large pretrained models like CLIP [40], which are often adapted to downstream tasks using a small amount of labeled data (as compared to the web-scale pretraining data). This is shown by many studies ([67, 68]) to be likely to obtain poor generalization performance in special domains, such as satellite imagery and fine-grained classification of car models or flower species. Such overfitting behavior is a result of limited data for those *tail class concepts* in both pretraining and downstream tasks. The domain gap between pretraining and downstream data further compounds the generalization problem. For instance, CLIP may not see enough image-text pairs to identify different car models during pretraining. This makes downstream generalization to fine-grained car models difficult, especially in low-data scenarios.

In this paper, we investigate using *pure text-based knowledge* to boost the downstream generalization of CLIP over different data distributions and tasks, with a special focus on few-shot and OOD tasks. Note similar ideas have been explored in recent works that leverage the implicit textual knowledge in Large Language Models (LLMs) to aid vision-language tasks. For example in [48, 56], GPT-2/3 [5] is used to generate image descriptions for the tasks of Visual Question Answering (VQA) and image captioning. While for CLIP-based image classification, [34, 39, 55] use GPT-3 to generate natural language attributes or captions for each class, and then classify images based on such information. Despite the success of these methods, they all suffer from large inference cost due to the use of LLM

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

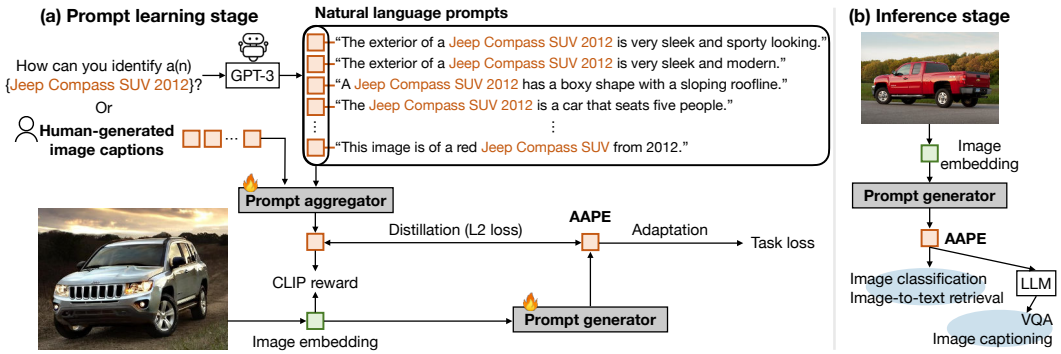


Figure 1: Aggregate-and-adapt the textual knowledge in natural language prompts for downstream tasks. **(a)** For classification of object-centric images, we query GPT-3 to obtain a list of prompts for each class, *e.g.*, the car model of "Jeep Compass SUV 2012". Note how redundant the reference prompts can be (*e.g.*, the first two), and how they can be irrelevant to the image (*e.g.*, the last prompt). Alternatively, for complex tasks like VQA, we use human-generated image captions to depict multi-object images. For all tasks, we first learn to aggregate the reference prompts into an image-aligned "summary" (prompt embedding) based on CLIP reward. Then a prompt generator is jointly trained to generate Aggregate-and-Adapted Prompt Embedding (AAPE), such that the distance between AAPE and the aggregated summary is minimized and the task loss is minimized too for adaptation purpose. **(b)** At test time, we only keep the prompt generator with the prompt aggregator discarded. Our AAPE is applicable to different vision-language tasks with strong generalization performance.

at test time. More critically, the LLM-generated texts are not necessarily beneficial, since they might be noisy and irrelevant to the considered task (see one example in Fig. 1(a)).

To address the two issues, we propose a new prompt learning method that is boosted by task-relevant language priors but does not incur any LLM cost at test time. The high-level idea is to learn prompts via distillation of input-adapted textual knowledge, which is especially useful to recognize under-represented visual concepts. Specifically, for classification of *object-centric images*, we follow [39] to first query GPT-3 for a set of natural language prompts that describe each class. While for more complex tasks like VQA, we use human-generated image captions that can depict *multi-object images* with object interactions in cluttered background. More importantly, for both cases, we *learn* to aggregate the collected reference prompts into a single prompt embedding, which is optimized by the CLIP reward to have high similarity with input image. This allows us to obtain a condensed prompt embedding that is image-aligned, ruling out redundant and irrelevant information, *e.g.*, ignoring the prompt elements about headlights for an image of rear facing car. Finally, we jointly train a prompt generator conditioned on input image to generate a prompt embedding with two objectives: 1) staying close to the aggregated embedding (*i.e.*, distillation from the condensed textual knowledge), 2) minimizing the task loss (*i.e.*, downstream adaptation).

Fig. 1 illustrates our "aggregate-and-adapt" method for prompt learning. Note prompt aggregation and distillation is only required for the learning stage. At test time, we will discard the aggregator and use the learned prompt generator as a standalone module. This leads to compute-efficiency when compared to prior works [34, 39, 48, 55, 56] since we entirely eliminate the LLM-induced inference cost. Our generated **A**ggregate-and-**A**dapted **P**rompt **E**mbedding (or **AAPE**) prove highly effective on various downstream vision-language tasks. We show AAPE is a new state-of-the-art for few-shot image classification on 11 datasets, under different OOD generalization settings. AAPE can also generalize zero-shot to tasks like image-to-text retrieval, image captioning and VQA. When finetuned on these tasks, AAPE achieves even better performance than SOTA vision-language models (*e.g.*, MAGMA [11]) whose entire image and text networks are fine-tuned at large cost.

To summarize, our **main contributions** are:

- A new prompt learning method that distills the textual knowledge from human- or LLM-generated natural language prompts to improve the downstream generalization of CLIP.
- Our learned AAPE achieves compelling performance on various downstream vision-language tasks, including image-to-text retrieval, few-shot classification, image captioning and VQA.
- We offer insightful findings that AAPE is especially helpful when there are under-represented concepts in few-shot and OOD settings or ambiguous visual cues in non-canonical image views. AAPE learning is also data-efficient and scales better than baselines with LLM model size.

2 Related Work

Vision-language models. Large-scale vision-language models achieve remarkable performance on a variety of downstream tasks. One learning paradigm is based on generative encoder-decoder models, which allows a sequence-to-sequence learning format that can connect visual data to free-form language prompts [2, 8, 11, 33]. Recent works like LiMBeR [35] show it is also possible for an LLM to operate on simple linear mappings of visual features. Another learning paradigm is based on contrastive learning with image-text pairs, which is popularized by CLIP [40] and numerous follow-ups [18, 24, 29, 41, 46, 59, 63, 66]. However, both categories of vision-language models (*e.g.*, CLIP and generative PaLI [8]) are found to struggle with special visual concepts or domains. In this paper, we focus on CLIP and improve its downstream generalization, especially under few-shot and OOD settings, via distillation of language priors. Nevertheless, our approach can be applied to other vision-language models, which we leave as future work.

Prompt learning is a parameter-efficient yet effective framework to finetune CLIP even in low-data settings. Most prompt learning methods learn text prompt vectors [67, 68] in place of hand-written sentence prompts. Other methods show the possibility of learning prompts in the image space [19], or in both image and text spaces [20, 64]. To reduce overfitting to seen classes during prompt learning, recent works focus on new class feature synthesis [52, 65], improved optimization [25, 45] and regularization [21] strategies. More related to our approach are [7, 58, 69] that align the learned prompts with hand-written prompts, with the goal of not forgetting the text knowledge from human input. These methods can be interpreted as a way of knowledge distillation from only short prompt templates. We will show the distillation from such prompt templates is suboptimal when compared to distillation from natural language prompts using LLMs.

Leveraging language in vision tasks. There is a long line of works on leveraging language to aid vision or multimodal tasks. One family of methods rely on external natural language datasets to retrieve text knowledge of image categories. For example, [6, 44] show improvements on ImageNet classification using the class descriptions retrieved from WordNet [36] and ImageNet-Wiki [6]. More recent works use LLMs to generate text for downstream tasks. GPT-3 is used in [48, 56] to help with the VQA and image captioning tasks. GPT-3 is also used in [34, 39, 55] to generate class-wise attributes or captions for CLIP-based classification. Unfortunately, all these prior works suffer from the noisy text that may be task-irrelevant. In this paper, we learn to adapt LLM-generated text to the target task, but without incurring any LLM-induced inference cost.

3 Method

Our "aggregate-and-adapt" method for prompt learning consists of three key components: 1) generating natural language prompts per image or class, 2) learning an aggregated prompt embedding that aligns with input image and 3) learning to generate Aggregate-and-Adapted Prompt Embedding (AAPE) for downstream tasks. In the following, we provide the details for each component. Note our method is based on CLIP [40], but it can be easily applied to other CLIP-like vision-language models.

3.1 Generating Natural Language Prompts

Prompt engineering. For CLIP-based image classification, the standard approach requires a set of hand-written prompts, *e.g.*, "a {} in a video game" and "a dark photo of a {}", which are completed with the class name. However, this is costly because one needs to hand-construct a different set of prompt templates for each dataset (*e.g.*, 80 for ImageNet in [40]), and that requires excessive prior knowledge about the target domain. Moreover, such prompt templates lack the descriptive details for discriminating fine-grained classes.

LLM-generated prompts. For image classification, we make use of the rich knowledge in LLMs to generate natural language prompts for a given class. One benefit of using LLMs is the ability to generate an arbitrary number of prompts, without relying on any domain knowledge. In particular, we follow the CuPL method [39] and query GPT-3 [5] for a prompt set in a scalable way. Specifically, GPT-3 is queried with a few *LLM-prompts* such as "Describe what a(n) {} looks like" and "How can you identify a(n) {}?". Then for each LLM-prompt, GPT-3 generates 10 reference prompts using a

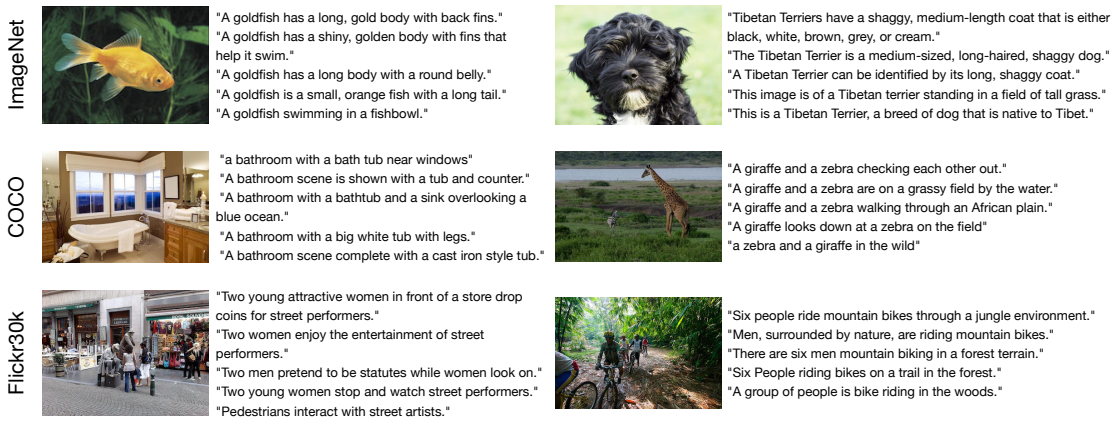


Figure 2: LLM-generated image prompts for ImageNet categories, and the hand-constructed image captions on COCO and Flickr30k datasets. Note ImageNet mainly contains **object-centric images** with relatively clean background, and the LLM-generated image prompts can describe distinct characteristics of the given classes. While COCO and Flickr30k contain **multi-object images** with cluttered background, and the hand-constructed captions can represent varying object relations.

high temperature of 0.99 for diversity. Fig. 1(a) shows some example prompts for a particular car model "Jeep Compass SUV 2012". Note how the reference prompts specify the car's discriminating characteristics in its sleek exterior. For the 11 classification datasets considered in our work, we follow the full generation setting in [39]: for each dataset, we use a different set of LLM-prompts (between 2 to 9), resulting in 20-90 reference prompts generated for each class.

Human-generated image captions. As illustrated in Fig. 2, the LLM-generated class-wise prompts are mainly suited for the classification task, where there are often object-centric images with clean background. For more complex vision-language tasks like VQA, deeper understanding is required for multi-object scenes with varying object interactions and cluttered background. To capture the textual knowledge for describing multi-object images, we use their image captions available from image-text datasets as a source of natural language prompts. Here we use COCO dataset [31] that consists of 5 human-annotated captions per image. It will be shown that our prompt embedding learned on COCO suffices to generalize to three difficult vision-language tasks (image-to-text retrieval, image captioning and VQA) with varying data distributions.

3.2 Input-Adapted Prompt Aggregator

The LLM-generated prompts have one notable issue: they are not necessarily a good representation of input image. For example in Fig. 1(a), it is inaccurate to describe the input image of a silver Jeep SUV as a red one in the last prompt, whereas other prompts are more relevant. Obviously, it would be detrimental to directly use the noisy prompts to supervise the following learning stage. Another issue with both the LLM- and human-generated prompts is that they are highly redundant with repeated information. To find a better supervisory signal, we propose to first aggregate the reference prompts into an image-aligned, condensed "summary". Such prompt summary is expected to have filtered noise as well as reduced redundancy.

Given n generated prompts, we first use the text encoder of CLIP to obtain the prompt embeddings $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n]$. Then we learn an adaptive prompt aggregator to condense $\boldsymbol{P}$ into m ($\ll n$) embeddings of the same size. Ideally, the aggregation should be invariant to permutations of $\boldsymbol{P}$, and scales as $\mathcal{O}(n+m)$. Here we set $m = 1$ for efficiency concerns. The aggregated result, a single prompt embedding, is denoted as $\boldsymbol{p}^a$. One simple aggregator that has the properties of permutation invariance and high efficiency is based on just averaging $\boldsymbol{P}$ into $\bar{\boldsymbol{p}}$. Simple averaging is widely used in prior works [34, 39]. However, the mean $\bar{\boldsymbol{p}}$ would still be compromised by the irrelevant information in $\boldsymbol{P}$. Here we introduce an attention-based aggregator which allow us to align the aggregated $\boldsymbol{p}^a$ with input image. At the same time, our prompt aggregator remains efficient and permutation invariant.

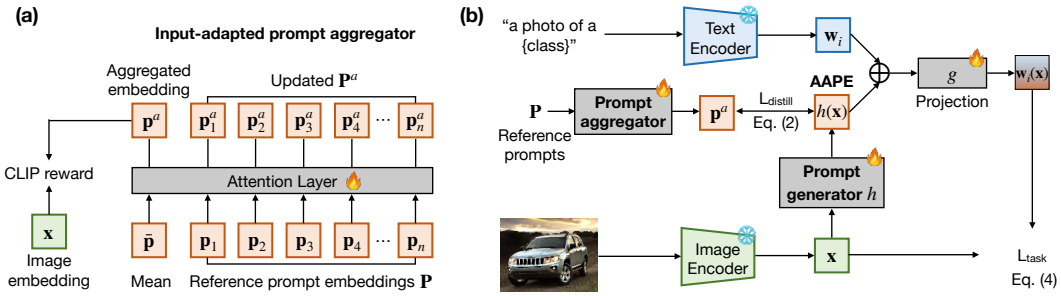


Figure 3: **(a)** Input-adapted prompt aggregator which aggregates the embeddings of reference prompts $\boldsymbol{P}$ into an image-aligned, condensed prompt embedding $\boldsymbol{p}^a$ based on CLIP reward. **(b)** Instantiation of our prompt learning approach for image classification. The CLIP model is kept frozen.

Fig. 3(a) shows the architecture of our input-adapted prompt aggregator based on just one attention layer. The attention layer takes as input the reference prompts $\boldsymbol{P}$ and a learnable prompt embedding (initialized as $\bar{\boldsymbol{p}}$). Then all the embeddings are updated as follows:

$$[\boldsymbol{p}^a, \boldsymbol{P}^a] = \texttt{AttentionLayer}\left([\bar{\boldsymbol{p}}, \boldsymbol{P}]\right), \tag{1}$$

where $\boldsymbol{p}^a$ is the desired prompt aggregation. The attention layer consists of standard multi-head cross-attention and feed-forward networks together with LayerNorm [3].

To make $\boldsymbol{p}^a$ semantically related to the input image (with embedding $\boldsymbol{x}$), we optimize our prompt aggregator using the CLIP reward [17] in form of $\texttt{CLIP-S}(\boldsymbol{x}, \boldsymbol{p}^a) = s \cdot \max(\cos(\boldsymbol{x}, \boldsymbol{p}^a), 0)$. The CLIP reward allows $\boldsymbol{p}^a$ to selectively blend image-related prompts through the attention mechanism. Fig. 7 in Appendix B.1 confirms that redundant or irrelevant reference prompts tend to have low attention scores, hence they are suppressed during prompt aggregation.

3.3 Learning AAPE

The per-image prompt aggregation $\boldsymbol{p}^a$ offers useful textual knowledge to supervise the following prompt learning stage. In this section, we elaborate how to improve prompt learning by distilling the aggregated text knowledge from $\boldsymbol{p}^a$. Note CLIP is kept frozen during prompt learning.

As a key innovation of this paper, we propose to train a prompt generator h that directly generates the prompt embedding $h(\boldsymbol{x})$ conditioned on image features $\boldsymbol{x}$. We parameterize h as a lightweight network with two fully connected layers and ReLU nonlinearity. h is trained using a distillation loss $\mathcal{L}_{\text{distill}}$ for knowledge distillation from $\boldsymbol{p}^a$, as well as a task loss $\mathcal{L}_{\text{task}}$ for downstream adaptation. We call such learned $h(\boldsymbol{x})$ as Aggregate-and-Adapted Prompt Embedding (AAPE). AAPE can also be viewed as an *image captioning embedding* in the latent space, since useful text knowledge is distilled in AAPE. In the following, we detail the two training losses.

Distillation loss is simply defined as the Euclidean distance between $h(\boldsymbol{x})$ and $\boldsymbol{p}^a$:

$$\mathcal{L}_{\text{distill}} = \|h(\boldsymbol{x}) - \boldsymbol{p}^a\|_2^2. \tag{2}$$

Task loss – image classification. Besides distilling the textual knowledge from $\boldsymbol{p}^a$, $h(\boldsymbol{x})$ should adapt to the downstream task too. Here we start with the instantiation of adapting $h(\boldsymbol{x})$ to the most studied task of image classification.

Note existing prompt learners for classification (*e.g.*, [67, 68]) learn *individual word tokens* $\{\boldsymbol{v}_l\}_{l=1}^L$ in a prompt, and then combine the learned tokens with class name embeddings $\boldsymbol{c}_{i\in[1,\ldots,C]}$ to obtain the full prompt. By contrast, we directly generate a full prompt embedding $h(\boldsymbol{x})$ without token-wise prediction. For classification, we simply combine $h(\boldsymbol{x})$ and the embedding of a prompt template "a photo of a `{class}`" to act as the classifier weights (to be matched to image features $\boldsymbol{x}$).

Fig. 3(b) shows the overall prompt learning framework. For the prompt template filled with the i-th class name, we use the text encoder of CLIP to obtain the template embedding $\boldsymbol{w}_i \in \mathbb{R}^d$. Next, we concatenate $\boldsymbol{w}_i$ and our $h(\boldsymbol{x}) \in \mathbb{R}^d$ followed by a projection g, giving $\boldsymbol{w}_i(\boldsymbol{x}) = g\left([\boldsymbol{w}_i^\top, h(\boldsymbol{x})^\top]^\top\right)$. Note $\boldsymbol{w}_i(\boldsymbol{x})$ does not involve any prompt engineering effort. We rely on $\boldsymbol{w}_i$ to mainly encode the class name, while $h(\boldsymbol{x})$ enriches that with input-adapted class descriptions in the latent space.

We parameterize g: $\mathbb{R}^{2d} \rightarrow \mathbb{R}^d$ by one fully connected layer with ReLU nonlinearity. The nonlinearity is important since it ensures $\boldsymbol{w}_i(\boldsymbol{x})$ is not trivially equivalent to the linear combination of $\boldsymbol{w}_i$ and $h(\boldsymbol{x})$, which will ignore $h(\boldsymbol{x})$ if we match the linear combination to $\boldsymbol{x}$ for classification. The classification probability is given as:

$$p\left(y=c \mid \boldsymbol{x}\right)=\frac{\exp\left(\cos\left(\boldsymbol{x}, \boldsymbol{w}_c(\boldsymbol{x})\right)/\tau\right)}{\sum_{i=1}^{C}\exp\left(\cos\left(\boldsymbol{x}, \boldsymbol{w}_i(\boldsymbol{x})\right)/\tau\right)}, \tag{3}$$

where C is the total number of classes, τ and $\cos(\cdot,\cdot)$ denote the temperature and cosine similarity. Appendix B.2 (Table 5) compares with an $h(\boldsymbol{x})$-only baseline for classification, without combining $\boldsymbol{w}_i$ or using projection g. Results show our default classification framework achieves solid gains over the $h(\boldsymbol{x})$-only baseline.

Finally, we arrive at the overall loss function to train the prompt generator h together with projection g for image classification:

$$\mathcal{L}=\lambda\mathcal{L}_{\text{distill}}+\mathcal{L}_{\text{task}}, \quad \text{where} \;\; \mathcal{L}_{\text{task}}=-\log p\left(y=c \mid \boldsymbol{x}\right), \tag{4}$$

and $\lambda=5$ is a weighting parameter. Table 6 in Appendix C provides the sensitivity analysis for λ, which shows performance is quite robust to the λ value in a wide range.

Note during testing we only use the prompt generator h, without querying LLM or using the prompt aggregator anymore. This removes the LLM-induced inference cost as required in [34, 39, 48, 55, 56], shifting such cost into our learning stage.

Task loss – beyond classification. Will the textual knowledge in $h(\boldsymbol{x})$ or AAPE benefit other vision-language tasks? We consider three tasks beyond classification: **image-to-text retrieval**, **image captioning** & **VQA**. Note these tasks involve multi-object images as mentioned in Section 3.1. Hence we use COCO image captions that contain textual descriptions of object relations. For all the three tasks, we train both the prompt aggregator and prompt generator h on 5 COCO captions per image. The same $\mathcal{L}_{\text{distill}}$ in Eq. (2) is used, while $\mathcal{L}_{\text{task}}$ is the CLIP loss.

For image-text retrieval, we simply use AAPE as the query and evaluate its zero-shot text retrieval performance on Flickr30k dataset [60]. We also evaluate the finetuning performance on Flickr30k when the prompt generator h is finetuned using $\mathcal{L}_{\text{distill}}$ and the corresponding retrieval loss $\mathcal{L}_{\text{task}}$.

For image captioning and VQA tasks, we conjecture that the textual knowledge encoded in AAPE will be especially useful when the visual cues are confusing or missing. To test this hypothesis, we use the COCO-trained AAPE in a straightforward manner. Concretely, we follow the LiMBeR baseline in [35] which linearly transforms the CLIP image representation into a sequence of prompt embeddings that an LLM can process. Then AAPE is appended to the prompt sequence as a "prefix" to offer rich language priors. We similarly evaluate performance of the zero-shot or finetuned prompt generator h on downstream datasets. For finetuning, h is optimized using $\mathcal{L}_{\text{distill}}$ and the corresponding task loss $\mathcal{L}_{\text{task}}$ (captioning or VQA).

4 Experimental Setup and Datasets

4.1 Few-shot Image Classification

Datasets. We use 11 datasets: ImageNet [10], Caltech101 [12], OxfordPets [38], StanfordCars [23], Flowers102 [37], Food101 [4], FGVC-Aircraft [32], SUN397 [54], UCF101 [47], DTD [9] and EuroSAT [14]. These datasets cover a wide range of generic objects and scenes, fine-grained object classes, as well as special domains with textural and satellite images. The various visual concepts in these datasets are perfect to test whether and when the textual knowledge in LLM will help. We further evaluate domain generalization on ImageNetV2 [42], ImageNet-Sketch [51], ImageNet-A [16] and ImageNet-R [15], which have different types of domain shift from ImageNet.

Implementation. We follow the prompt learning details in [67], including the CLIP vision backbone (ViT-B/16), learning rate schedule and the number of epochs for each dataset. Appendix C (Table 7) provides a detailed analysis of the compute cost measured on Nvidia V100 GPU, where all prompt learners are evaluated for fair efficiency comparisons.

Table 1: **Image-to-Text Retrieval:** zero-shot and finetuned results (Recall@K) on Flickr30k dataset. Note our AAPE is learned on the COCO dataset.

		COCO			Flickr30k		
		R@1	R@5	R@10	R@1	R@5	R@10
Train or Finetune	Unicoder-VL [26]	62.3	87.1	92.8	86.2	96.3	99.0
	Oscar [28]	73.5	92.2	96.0	-	-	-
	ERNIE-ViL [61]	-	-	-	88.7	98.0	99.2
	AAPE	**76.7**$_{\pm 0.1}$	**94.5**$_{\pm 0.1}$	**97.4**$_{\pm 0.1}$	**94.2**$_{\pm 0.2}$	**99.3**$_{\pm 0.1}$	**99.7**$_{\pm 0.1}$
Zero-shot	CLIP [40]	58.4	81.5	88.1	88.0	98.7	99.4
	SigLIP [66]	65.4	85.1	91.1	91.5	98.1	99.4
	Llip [24]	**68.1**	**87.6**	**92.5**	93.2	99.0	99.4
	BLIP-2 [27]	-	-	-	**96.9**	**100.0**	**100.0**
	AAPE	-	-	-	90.8$_{\pm 0.2}$	98.6$_{\pm 0.1}$	99.4$_{\pm 0.1}$

Evaluations. We follow the few-shot evaluation protocol in [40], using 1, 2, 4, 8 and 16 shots per class for training (default 16), and the full testset for evaluation. Two OOD generalization settings are considered as in [67]. 1) Generalization from base to new classes within one dataset, *i.e.*, training on the base class split but testing on both base and new class splits. This helps evaluate the ID and OOD performance under a class-incremental domain shift. We follow [53] to also measure the harmonic mean of base and new class accuracies to quantify the ID and OOD performance trade-off. 2) Domain generalization where one trains on ImageNet (with 16 shots) and evaluates on four ImageNet variants. For all experiments, we report results as an average over three random seeds.

4.2 Vision-Language Understanding and Generation Tasks

As mentioned in Section 3.3, we perform prompt learning on COCO dataset [31] before evaluation on three vision-language tasks. For the task of image-to-text retrieval, we use the same CLIP vision backbone VIT-L/14 as in [24, 40] for fair comparisons. We show both zero-shot and finetuned results (Recall@K) on Flickr30k [60].

For captioning and VQA tasks, we follow LiMBeR [35] to use the same language model and CLIP vision backbone (RN50x16). We evaluate on image captioning datasets COCO and NoCaps [1]. Zero-shot and finetuned results are reported in terms of CIDEr-D [50], CLIPScore, and Ref-CLIPScore [17]. For VQA, we prompt the model with the "[image] Q: [q] A:" format. The generation is truncated to the length of the longest ground truth answer. For evaluation, we use the VQA2 dataset [13] and follow the few-shot setting in [11] to report accuracy metric for every K-shots.

Appendix C (Table 7) shows the high efficiency with the straightforward use of AAPE for tasks beyond classification. When compared to the SOTA fully fine-tuned model MAGMA [11], AAPE is about 2.8/1.2 times faster for training/inference on Nvidia A100 GPU.

5 Results

5.1 Image-to-Text Retrieval

We use this task to verify if AAPE can act as a meaningful image captioning embedding, which is learned to distill image-aligned text knowledge from available image captions. We do not aim to push for state-of-the-art performance for the retrieval task. Table 1 shows that we can indeed achieve strong training and finetuning performance on COCO and Flickr30k datasets, respectively. This indicates our prompt learning method is competent with producing high-quality text or captioning embedding that can successfully fulfill the task at hand. It is also worth noting that our AAPE learned on COCO can perform zero-shot retrieval on Flickr30k, obtaining competitive results with SOTA zero-shot models (*e.g.*, CLIP and SigLIP). This further demonstrates the good generalization capability of AAPE over different data distributions.

Table 2: **Few-shot classification in the base-to-new class generalization setting**. OGEN denotes the OGEN+PromptSRC variant. Our AAPE follows CuPL to query an LLM to obtain natural language prompts, but further learns from those prompts. H: Harmonic mean of base and new class accuracies.

		Prompt learning without language priors						**Human-generated prompts**			**LLM-based**	
		CoOp	CoCoOp	MaPLe	CLIPood	PromptSRC	OGEN	ProGrad	KgCoOp	LASP-V	CuPL	AAPE
Avg across 11 datasets	Base	82.69	80.47	82.28	83.90	84.26	84.17	82.48	80.73	83.18	74.31	**84.72**$_{\pm 0.18}$
	New	63.22	71.69	75.14	74.50	76.10	76.86	70.75	73.60	76.11	75.25	**77.54**$_{\pm 0.29}$
	H	71.66	75.83	78.55	78.90	79.97	80.34	76.16	77.00	79.48	74.78	**80.97**$_{\pm 0.19}$
ImageNet	Base	76.47	75.98	76.66	77.50	77.60	77.50	77.02	75.83	76.25	75.05	**78.10**$_{\pm 0.11}$
	New	67.88	70.43	70.54	70.30	70.73	70.97	66.66	69.96	71.17	68.43	**71.98**$_{\pm 0.14}$
	H	71.92	73.10	73.47	73.70	74.01	74.09	71.46	72.78	73.62	71.59	**74.92**$_{\pm 0.12}$
Caltech101	Base	98.00	97.96	97.74	**98.70**	98.10	98.32	98.02	97.72	98.17	98.24	98.34$_{\pm 0.07}$
	New	89.81	93.81	94.36	94.60	94.03	94.76	93.89	94.39	94.33	94.34	**94.79**$_{\pm 0.09}$
	H	93.73	95.84	96.02	**96.60**	96.02	96.50	95.91	96.03	96.21	96.25	96.53$_{\pm 0.07}$
OxfordPets	Base	93.67	95.20	95.43	95.70	95.33	95.96	95.07	94.65	95.73	95.30	**96.89**$_{\pm 0.12}$
	New	95.29	97.69	97.76	96.40	97.30	97.48	97.63	97.76	97.87	97.74	**98.02**$_{\pm 0.16}$
	H	94.47	96.43	96.58	96.00	96.30	96.71	96.33	96.18	96.79	96.50	**97.45**$_{\pm 0.13}$
Stanford Cars	Base	78.12	70.49	72.94	**78.60**	78.27	77.59	77.68	71.76	75.23	68.88	77.51$_{\pm 0.39}$
	New	60.40	73.59	74.00	73.50	74.97	75.17	68.63	75.04	71.77	75.09	**77.37**$_{\pm 0.56}$
	H	68.13	72.01	73.47	75.90	76.58	76.38	72.88	73.36	73.46	71.85	**77.44**$_{\pm 0.42}$
Flowers102	Base	97.60	94.87	95.92	93.50	**98.07**	97.34	95.54	95.00	97.17	77.79	97.81$_{\pm 0.19}$
	New	59.67	71.75	72.46	74.50	76.50	77.67	71.87	74.73	73.53	78.10	**78.75**$_{\pm 0.31}$
	H	74.06	81.71	82.56	82.90	85.95	86.39	82.03	83.65	83.71	77.94	**87.25**$_{\pm 0.21}$
Food101	Base	88.33	90.70	90.71	90.70	90.67	90.69	90.37	90.50	91.20	90.56	**91.82**$_{\pm 0.08}$
	New	82.26	91.29	92.05	91.70	91.53	91.68	89.59	91.70	91.90	91.86	**92.66**$_{\pm 0.11}$
	H	85.19	90.99	91.38	91.20	91.10	91.19	89.98	91.09	91.54	91.21	**92.24**$_{\pm 0.09}$
FGVC Aircraft	Base	40.44	33.41	37.44	**43.30**	42.73	41.26	40.54	36.21	38.05	33.29	41.46$_{\pm 0.12}$
	New	22.30	23.71	35.61	37.20	37.87	40.26	27.57	33.55	33.20	37.60	**40.37**$_{\pm 0.28}$
	H	28.75	27.74	36.50	40.00	40.15	40.75	32.82	34.83	35.46	35.31	**40.91**$_{\pm 0.14}$
SUN397	Base	80.60	79.74	80.82	81.00	82.67	82.57	81.26	80.29	80.70	73.39	**82.93**$_{\pm 0.14}$
	New	65.89	76.86	78.70	79.30	78.47	78.83	74.17	76.53	79.30	75.69	**79.87**$_{\pm 0.27}$
	H	72.51	78.27	79.75	80.20	80.52	80.65	77.55	78.36	80.00	74.52	**81.37**$_{\pm 0.18}$
DTD	Base	79.44	77.01	80.36	80.80	83.37	83.75	77.35	77.55	81.10	62.45	**83.97**$_{\pm 0.45}$
	New	41.18	56.00	59.18	58.60	62.97	62.54	52.35	54.99	62.57	60.31	**63.64**$_{\pm 0.58}$
	H	54.24	64.85	68.16	67.90	71.75	71.60	62.45	64.35	70.64	61.36	**72.41**$_{\pm 0.39}$
EuroSAT	Base	92.19	87.49	94.07	**97.50**	92.90	93.40	90.11	85.64	95.00	65.38	95.40$_{\pm 0.33}$
	New	54.74	60.04	73.23	64.10	73.90	76.74	60.89	64.34	**83.37**	69.97	76.30$_{\pm 0.86}$
	H	68.69	71.21	82.35	77.30	82.32	84.25	72.67	73.48	**88.86**	67.60	84.79$_{\pm 0.63}$
UCF101	Base	84.69	82.33	83.00	85.70	87.10	87.44	84.33	82.89	85.53	77.13	**87.69**$_{\pm 0.23}$
	New	56.05	73.45	78.66	**79.30**	78.80	79.28	74.94	76.67	78.20	78.58	79.21$_{\pm 0.39}$
	H	67.46	77.64	80.77	82.40	82.74	83.16	79.35	79.65	81.70	77.85	**83.23**$_{\pm 0.31}$

5.2 Few-shot Image Classification

Base-to-new class generalization. Table 2 compares AAPE with two categories of prompt learning methods: 1) CoOp [67], CoCoOp [68], MaPLe [20], CLIPood [45], PromptSRC [21] and OGEN [65]. These methods learn prompt vectors without using any text-based knowledge, but heavily rely on advanced optimization and regularization strategies to improve generalization. Our AAPE, on the other hand, outperforms by distilling the textual knowledge from LLMs. Notably, on average (across 11 datasets), AAPE achieves better classification accuracies than the previous SOTA approach OGEN for both base and new classes, setting a new SOTA mean accuracy 80.97% (vs. 80.34%). 2) ProGrad [69], KgCoOp [58] and LASP-V [7]. These methods choose to align the learned prompts with hand-written ones like "a dark photo of a {`class`}", hence distilling knowledge from only basic, non-descriptive templates. This proves less effective than our LLM-derived natural language priors.

Recall our prompt learning method is built on top of the CuPL approach [39] to obtain the knowledge of GPT-3. Here we show both the LLM knowledge and our learning algorithm (that adaptively distills the knowledge) are indispensable. We first compare with the CuPL baseline that leverages LLM knowledge, but without any learning. Specifically, CuPL averages the LLM-generated prompts to perform zero-shot classification. This is contrasted with our method that *learns* to aggregate the noisy prompts into AAPE which is then adapted for classification. Table 2 shows such "aggregate-and-adapt" learning method leads to significant gains over the learning-free CuPL, especially for base classes.

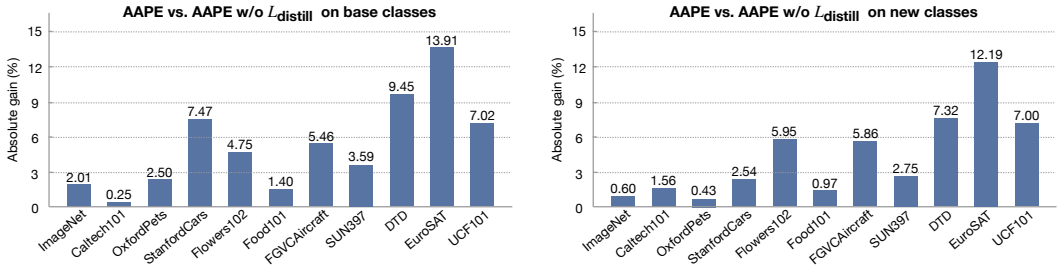


Figure 4: **Quantifying the role of LLM knowledge (distilled with $\mathcal{L}_{\text{distill}}$) in prompt learning.** $\mathcal{L}_{\text{distill}}$ consistently improves the base and new class accuracies on 11 classification datasets.

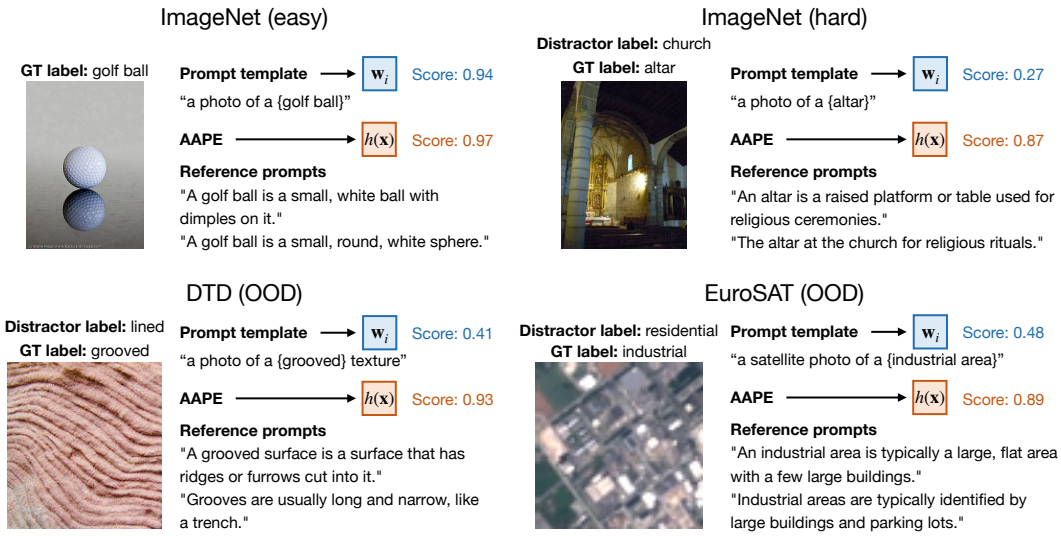


Figure 5: **AAPE helps disambiguate the classification task**. To highlight the textual knowledge encoded in AAPE, we show some reference prompts generated by GPT-3. For both the prompt template and AAPE (before concatenation and projection), we measure their Cosine similarity score with the image. Note the similarity score can be small when using a basic prompt template to match the "altar" class instance on ImageNet. Indeed, in this non-canonical image view, the altar is small and the whole scene can be classified as the easily confused class of "church". Whereas AAPE is able to eliminate confusion by providing additional cues like altar "is a raised table" often at the location of "church". This results in increased image-text similarity. Similarly, the textual cues from AAPE are helpful for the OOD examples in special domains of DTD and EuroSAT.

Next we compare with a variant of our approach, with only task loss $\mathcal{L}_{\text{task}}$ but no $\mathcal{L}_{\text{distill}}$ to distill LLM knowledge. This variant allows decoupling the contribution of LLM knowledge for prompt learning under a fair setting. Fig. 4 shows that using $\mathcal{L}_{\text{distill}}$ leads to consistent gains for both seen and unseen classes from all the considered datasets. The distilled textual knowledge makes an especially large impact for those fine-grained actions (UCF101) and visual classes (StanfordCars, Flowers102 and FGVCAircraft), which can be under-represented during both CLIP pretraining and prompt learning. Large gains are also observed for the special domains of textures (DTD) and satellite images (EuroSAT) with large distribution shift.

We further show how our distilled AAPE can disambiguate the classification task. Fig. 5 shows our AAPE augments the basic prompt template with descriptive details for each image, as exemplified by the reference prompts. Such input-specific details are often helpful for non-canonical views (*e.g.*, hard cases on ImageNet) and OOD examples (*e.g.*, on DTD and EuroSAT), where the visual cues are either ambiguous or barely visible (hence low similarity between the image and basic template). Eventually, we use a projection network to blend textual information from the template and AAPE, resulting in increased image-text similarity.

More comparisons. Table 8 in Appendix D includes **domain generalization** results. We see that AAPE is robust to different types of domain shift, outperforming prior works on 4 ImageNet variants. Table 9 and Appendix E further show the advantage of AAPE over two recent prompt learning methods ProText [22] and ArGue-N [49].

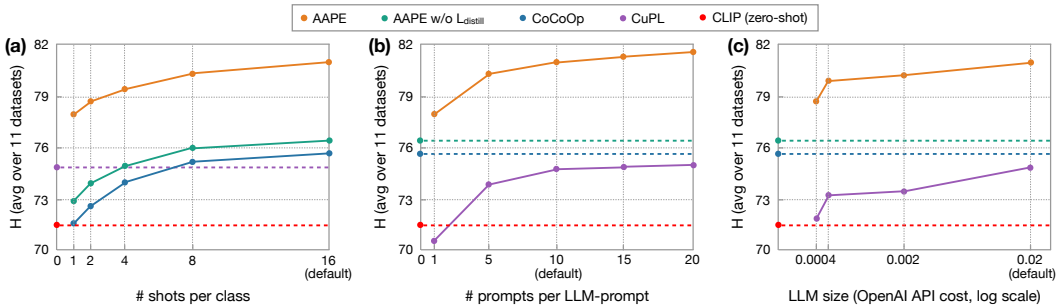


Figure 6: **AAPE scales better with data (a-b) and LLM size (c) than alternatives**. Experiments are conducted under the base-to-new generalization setting for few-shot classification. We measure the Harmonic mean (H) of base and new class accuracies. To adjust the total number of reference prompts per class to supervise AAPE learning, we vary the number of prompts generated by each LLM-prompt template. Four models of GPT-3 are considered: Ada, Babbage, Curie and Davinci.

Table 3: **Image captioning and VQA performance**. Note our AAPE is learned on COCO dataset, and we show both its zero-shot and finetuned results on the two tasks with different testing datasets.

	Image Captioning									VQA2 K-shots			
	COCO			NoCaps (CIDEr-D)				NoCaps (All)		0	1	2	4
	CIDEr-D	CLIP-S	Ref-S	In	Out	Near	All	CLIP-S	Ref-S				
MAGMA [11]	47.5	75.3	79.6	30.4	43.4	36.7	38.7	74.3	78.7	24.6	39.3	40.6	41.5
LiMBeR [35]	54.9	76.2	80.4	34.3	48.4	41.6	43.9	74.7	79.4	33.3	39.9	40.8	40.3
LiMBeR+AAPE (train/finetune)	**57.8**	**80.8**	**83.6**	**42.1**	**49.8**	**44.2**	**47.3**	**77.6**	**81.7**	**36.5**	**42.7**	**44.2**	**45.9**
LiMBeR+AAPE (zero-shot)	-	-	-	36.1	48.8	42.9	45.1	76.3	80.3	34.9	41.0	42.3	43.1

Ablation studies. Fig. 6(a) shows that AAPE learning is data-efficient. We see AAPE consistently outperforms two prompt learners that do not benefit from LLM's text knowledge, *i.e.*, AAPE w/o $\mathcal{L}_{\text{distill}}$ and a similar baseline CoCoOp [68]. Encouragingly, using 1 shot for AAPE is already far better than using 16 shots for the compared baselines. AAPE with varying shots is also consistently better than the LLM-based but learning-free approach CuPL [39]. Fig. 6(b) further shows that AAPE scales better with the number of used prompts than CuPL. Note when we use only 1 prompt generated by each LLM-prompt, CuPL is even worse than the CLIP baseline. Whereas AAPE performs much better by distilling task-related information from the limited number of prompts. Finally, Fig. 6(c) shows the benefits of AAPE over CuPL in terms of the scaling performance with LLM model size.

5.3 Image Captioning & VQA

Table 3 shows that AAPE can adapt to other vision-language tasks. Note AAPE is trained on COCO captions that describe complex scenes other than object-centric images. We once again find the benefits of distilling the text knowledge from image captions into AAPE. We observe consistent gains over the LiMBeR baseline, using either a trained or finetuned prompt generator (finetuned on NoCaps captioning and VQA2 tasks). More importantly, AAPE shows great generalization capability on both tasks. Its zero-shot performance is consistently better than that of LiMBeR and MAGMA, the latter of which finetunes both image and text networks. Fig. 9 in Appendix F exemplifies how AAPE can help on the captioning task, especially when the visual cues are ambiguous.

6 Conclusion

In this paper, we show the distillation of text-based knowledge into CLIP improves its downstream generalization. We propose a new prompt learning method where a prompt embedding AAPE is distilled from human- or LLM-generated natural language prompts. A prompt generator is trained to predict AAPE, which is shown to generalize to various vision-language tasks. We further demonstrate the benefits of AAPE for handling non-canonical examples as well as few-shot and OOD settings.

Limitations and future work. Given a sufficiently large set of image prompts, it is preferable to aggregate them into more than one prompt embeddings to model text diversity. However, learning to predict such an embedding set is hard on data-deficient tasks, since it will not only increase the computation cost but also incur performance degradation. For future work we hope to address this limitation by scaling up data to learn a diversified universal prompt generator. Another plan is to go beyond CLIP and apply AAPE learning to more vision-language models (contrastive or generative).

References

[1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. NoCaps: novel object captioning at scale. In *ICCV*, 2019.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014.

[5] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

[6] Sebastian Bujwid and Josephine Sullivan. Large-scale zero-shot image classification from rich and diverse textual descriptions. In *LANTERN*, 2021.

[7] Adrian Bulat and Georgios Tzimiropoulos. LASP: Text-to-text optimization for language-aware soft prompting of vision & language models. In *CVPR*, 2023.

[8] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Alexander Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *ICLR*, 2023.

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[11] Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. MAGMA – multimodal augmentation of generative models through adapter-based finetuning. *arXiv preprint arXiv:2112.05253*, 2022.

[12] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR workshop*, 2004.

[13] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, 2017.

[14] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 2019.

[15] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021.

[16] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021.

[17] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[18] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[19] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022.

[20] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. MaPLe: Multi-modal prompt learning. In *CVPR*, 2023.

[21] 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 *ICCV*, 2023.

[22] Muhammad Uzair khattak, Muhammad Ferjad, Naseer Muzzamal, Luc Van Gool, and Federico Tombari. Learning to prompt with text only supervision for vision-language models. *arXiv preprint arXiv:2401.02418*, 2024.

[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshops*, 2013.

[24] Samuel Lavoie, Polina Kirichenko, Mark Ibrahim, Mahmoud Assran, Andrew Gordon Wildon, Aaron Courville, and Nicolas Ballas. Modeling caption diversity in contrastive vision-language pretraining. *arXiv preprint arXiv:2405.00740*, 2024.

[25] Dongjun Lee, Seokwon Song, Jihee Suh, Joonmyeong Choi, Sanghyeok Lee, and Hyunwoo J. Kim. Read-only prompt optimization for vision-language few-shot learning. In *ICCV*, 2023.

[26] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, 2019.

[27] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[28] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020.

[29] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *ICLR*, 2022.

[30] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *NeurIPS*, 2022.

[31] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

[32] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[33] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier Biard, Sam Dodge, Philipp Dufter, Bowen Zhang, Dhruti Shah, Xianzhi Du, Futang Peng, Haotian Zhang, Floris Weers, Anton Belyi, Karanjeet Singh, Doug Kang, Ankur Jain, Hongyu He, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. MM1: Methods, analysis & insights from Multimodal LLM pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[34] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *ICLR*, 2023.

[35] Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly mapping from image to text space. In *ICLR*, 2023.

[36] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 1995.

[37] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

[38] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.

[39] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *ICCV*, 2023.

[40] 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 *ICML*, 2021.

[41] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. DenseCLIP: Language-guided dense prediction with context-aware prompting. In *CVPR*, 2022.

[42] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

[43] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[44] Sheng Shen, Chunyuan Li, Xiaowei Hu, Yujia Xie, Jianwei Yang, Pengchuan Zhang, Anna Rohrbach, Zhe Gan, Lijuan Wang, Lu Yuan, et al. K-lite: Learning transferable visual models with external knowledge. In *NeurIPS*, 2022.

[45] Yang Shu, Xingzhuo Guo, Jialong Wu, Ximei Wang, Jianmin Wang, and Mingsheng Long. CLIPood: Generalizing clip to out-of-distributions. In *ICML*, 2023.

[46] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A foundational language and vision alignment model. In *CVPR*, 2022.

[47] 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.

[48] Yixuan Su, Tian Lan, Yahui Liu, Fangyu Liu, Dani Yogatama, Yan Wang, Lingpeng Kong, and Nigel Collier. Language models can see: Plugging visual controls in text generation. *arXiv preprint arXiv:2205.02655*, 2022.

[49] Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. ArGue: Attribute-Guided Prompt Tuning for Vision-Language Models . In *CVPR*, 2024.

[50] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *CVPR*, 2015.

[51] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019.

[52] Zhengbo Wang, Jian Liang, Ran He, Nan Xu, Zilei Wang, and Tieniu Tan. Improving zero-shot generalization for clip with synthesized prompts. In *ICCV*, 2023.

[53] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *CVPR*, 2017.

[54] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[55] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *CVPR*, 2023.

[56] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of GPT-3 for few-shot knowledge-based VQA. In *AAAI*, 2022.

[57] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of LMMs: Preliminary explorations with GPT-4V(ision). *arXiv preprint arXiv:2309.17421*, 2023.

[58] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *CVPR*, 2023.

[59] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *ICLR*, 2022.

[60] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2014.

[61] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE-ViL: Knowledge enhanced vision-language representations through scene graph. In *AAAI*, 2021.

[62] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *CVPR*, 2023.

[63] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[64] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022.

[65] Yuhang Zang, Hanlin Goh, Joshua M. Susskind, and Chen Huang. Overcoming the pitfalls of vision-language model finetuning for OOD generalization. In *ICLR*, 2024.

[66] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[67] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022.

[68] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022.

[69] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. ICCV, 2023.

A Alternative Methods of Generating Natural Language Prompts

Text retrieval for object-centric image classification. It is possible to use the class name to retrieve class descriptions from WordNet [36] or image-text datasets, just like in previous works [6, 44]. Unfortunately, no single dataset, including the large-scale LAION-5B [43], contains arbitrary classes specified in downstream tasks (*e.g.*, fine-grained flower species). In other words, the retrieval-based approach can limit the possible classes that can be recognized. Furthermore, the image captions retrieved from image-text datasets can be noisy and irrelevant to the target class. One typical example is that the retrieved image captions are not really focused on the descriptive details of the target class instances. Instead, the captions can be depicting their relations with other objects in a multi-object image. Therefore, retrieved image captions may not be suited for object-centric classification. This is evidenced by Pratt et al. [39] who explored similar retrieval ideas for ImageNet classification, where each class is guaranteed to have class descriptions from WordNet [36] or Wikipedia articles (ImageNet-Wiki [6]). The retrieved captions prove less effective than LLM generated customized prompts in terms of ImageNet Top-1 accuracy.

Large Multimodal Models (LMMs) for object-centric classification and other vision-language tasks. One alternative way of acquiring natural language prompts is to query LMMs, *e.g.*, GPT-4V [57]. For the task of object-centric classification, it is easy to imagine that LMMs can be more competent than pure text-based LLMs in generating helpful prompts, since LMMs have both vision and language supervision in their pretraining. For the same reason, vision-language tasks like VQA are likely to benefit more from LMM-generated image prompts than human-annotated image captions. However, our goal here is to study the role of *text-only* knowledge for downstream generalization. Hence LMMs with additional vision supervision fall outside of the scope of our study.

B More Ablations and Analyses

B.1 Input-Adapted Prompt Aggregator

We rely on the input-adapted prompt aggregator to provide good textual supervision for prompt learning. The prompt aggregator is learned to aggregate all the reference prompts $\boldsymbol{P}$ into an image-aligned prompt embedding $\boldsymbol{p}^a$ with reduced noise and redundancy.

Fig. 7 confirms that redundant and noisy (irrelevant) reference prompts are often suppressed with low **attention scores** during prompt aggregation.

Next, we **compare with three alternative methods for prompt aggregation**:

- Two baselines are learning-free, obtaining $\boldsymbol{p}^a$ by either random sampling from $\boldsymbol{P}$ or simple averaging (*i.e.*, $\bar{\boldsymbol{p}}$).
- We further compare with a learning method using a different architecture other than the default attention network. Specifically, we start with the mean $\bar{\boldsymbol{p}}$ and learn to transform it via simple MLP layers such that the transformed embedding is aligned with input image. Hence we have $\boldsymbol{p}^a = \alpha f(\bar{\boldsymbol{p}}) + (1-\alpha)\bar{\boldsymbol{p}}$, where f is a two-layer bottleneck MLP with ReLU nonlinearity, while α is a learnable parameter to weight the residual connection. Note such MLP-based architecture differs from attention mechanism in that the MLP-aggregated $\boldsymbol{p}^a$ is not able to attend to individual prompts in $\boldsymbol{P}$ for dynamic information fusion.

Table 4: **Ablation study on our prompt aggregator.** We experiment under the base-to-new class generalization setting for few-shot classification. CLIP performance (zero-shot) is listed as a baseline. H: Harmonic mean of base and new class accuracies.

Avg across 11 datasets	Base	New	H
Zero-shot CLIP [40]	69.34	74.22	71.70
Random sampling	$70.13_{\pm 0.93}$	$74.61_{\pm 1.34}$	$72.30_{\pm 1.15}$
Mean $\bar{\boldsymbol{p}}$	$82.05_{\pm 0.13}$	$75.64_{\pm 0.21}$	$78.71_{\pm 0.15}$
MLP-based aggregation	$84.39_{\pm 0.21}$	$76.33_{\pm 0.28}$	$80.16_{\pm 0.22}$
Attention-based aggregation (default)	$\mathbf{84.72}_{\pm 0.18}$	$\mathbf{77.54}_{\pm 0.29}$	$\mathbf{80.97}_{\pm 0.19}$

Table 4 summarizes the comparison results on few-shot classification under the base-to-new class generalization setting. It is clear that random sampling is not a good choice, whose performance has

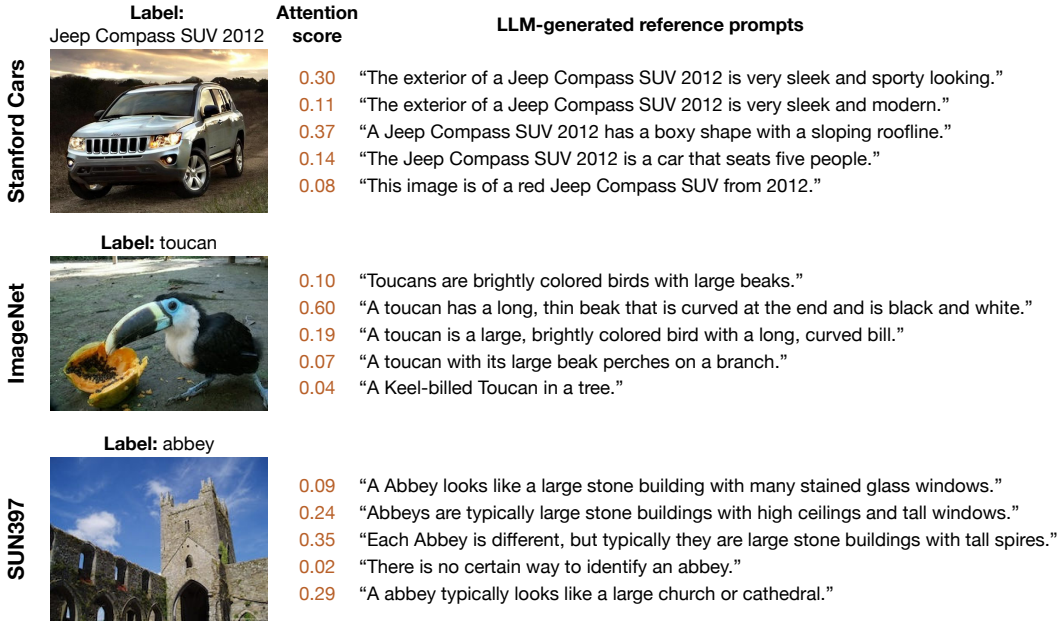


Figure 7: **Visualizing the attention score of each reference prompt during prompt aggregation**. Note the attention score is re-normalized among the illustrated prompt samples. We observe low attention scores for prompts that are redundant or noisy (irrelevant to input image), *e.g.*, the 2nd and 5th prompt in the car image.

Table 5: **Comparison with the $h(\boldsymbol{x})$-only baseline for few-shot classification.** We experiment under the base-to-new class generalization setting. H: Harmonic mean of base and new class accuracies.

	Avg across 11 datasets	Base	New	H
LLM-based	AAPE (default)	**84.72**$_{\pm 0.18}$	**77.54**$_{\pm 0.29}$	**80.97**$_{\pm 0.19}$
	(AAPE) $h(\boldsymbol{x})$-only	84.01$_{\pm 0.12}$	75.93$_{\pm 0.16}$	79.77$_{\pm 0.13}$
	CuPL [39]	74.31$_{\pm 0.00}$	75.25$_{\pm 0.00}$	74.78$_{\pm 0.00}$
w/o priors	AAPE w/o $\mathcal{L}_{\text{distill}}$	79.47$_{\pm 0.22}$	73.25$_{\pm 0.34}$	76.23$_{\pm 0.22}$
	CoCoOp [68]	80.47$_{\pm 0.21}$	71.69$_{\pm 0.37}$	75.83$_{\pm 0.24}$

large variance and is only marginally better than that of zero-shot CLIP. This is because the reference prompts are often noisy and not related to input image. Hence a randomly sampled prompt is likely a poor source of textual knowledge to distill from. When we use the mean prompt $\bar{\boldsymbol{p}}$ as supervision, significant gains are observed due to reduced noise as well as enriched information. MLP-based aggregation leads to larger gains, since learning is introduced now to find a better and image-aligned supervisory signal. With the attention mechanism, we achieve the best results with dynamic prompt aggregation, which is our default approach.

B.2 Simple Classification Framework Based on AAPE Only

In the main paper, we introduce our default classification framework in Section 3.3. The classifier weights are built from the combination of AAPE $h(\boldsymbol{x})$ and a template embedding $\boldsymbol{w}_i$ using a projection g. Here we compare with a simpler $h(\boldsymbol{x})$-only baseline for classification, without combining $\boldsymbol{w}_i$ or using projection g. This baseline uses the standard text classifier weights $\boldsymbol{w}_i$, while $h(\boldsymbol{x})$ acts as a proxy image query to be matched to $\boldsymbol{w}_i$. Such setup is extremely similar to the image-to-text retrieval task. Evaluating the $h(\boldsymbol{x})$-only baseline is a more direct quantification of how well the text knowledge in AAPE, the image captioning embedding, can distinguish different classes.

Table 5 shows the $h(\boldsymbol{x})$-only baseline achieves reasonable performance for both base and new classes, which indicates AAPE's good generalization on the classification task. When compared to CuPL that captures text knowledge by simply ensembling LLM-generated image prompts, the $h(\boldsymbol{x})$-only baseline is much more performant by learning an adaptive prompt aggregation. While the AAPE

Table 6: **Sensitivity analysis of the distillation loss weight** λ. We report the few-shot classification results (base-to-new class generalization setting) averaged across 11 datasets in terms of H, the Harmonic mean of base and new class accuracies.

λ	1	3	4	5	6	7	9
H	$79.14_{\pm 0.27}$	$80.65_{\pm 0.23}$	$80.84_{\pm 0.18}$	$\mathbf{80.97}_{\pm 0.19}$	$80.90_{\pm 0.14}$	$80.93_{\pm 0.16}$	$80.76_{\pm 0.21}$

Table 7: **Inference cost for few-shot classification and 3 other tasks beyond classification**. For few-shot classification, we report accuracy averaged over 11 datasets (base-to-new setting). We implement CoCoOp† to have a bigger prompt prediction network than CoCoOp, such that CoCoOp† has matching parameter count with AAPE.

Method	Comment	Compute cost			Few-shot accuracy		
		# params	GFLOP	FPS	Base	New	H
CoOp [67]	**Non-adaptive text prompts**	2k	162.5	1344	82.69	63.22	71.66
OGEN [65]		2k	162.5	1351	84.17	76.86	80.34
MaPLe [20]	**Text+image prompts**	3.55M	162.7	1365	82.28	75.14	78.55
PromptSRC [21]		46k	162.8	1380	84.26	76.10	79.97
CoCoOp [68]	**Input-adaptive text prompts**	35k	162.5	15.08	80.47	71.69	75.83
CoCoOp†		84k	162.6	14.67	81.05	70.12	75.19
AAPE $h(\boldsymbol{x})$+g for classification		82k	162.6	14.92	**84.72**	**77.54**	**80.97**
AAPE $h(\boldsymbol{x})$ beyond classification		33k	162.5	15.06	-	-	-

baseline w/o $\mathcal{L}_{\text{distill}}$ and CoCoOp both learn input-adapted prompts but without language supervision. We can observe evident benefits of our $h(\boldsymbol{x})$-only baseline over the language-free methods.

Lastly, our default AAPE-based classification framework consistently outperforms the $h(\boldsymbol{x})$-only baseline, only at a small overhead incurred by projection g. In the meantime, since the default classification framework simply combines and projects AAPE $h(\boldsymbol{x})$ and $\boldsymbol{w}_i$, it enables easy interpretation of the roles of the two components for classification, see Fig. 5.

C Hyperparameter Sensitivity and Compute Cost

Sensitivity analysis of the distillation loss weight λ in Eq. (4). Table 6 reports the results for the few-shot classification task. It is shown that AAPE performs robustly with overlapping confidence intervals when $\lambda \in [3, 9]$. AAPE even outperforms the strong baseline OGEN (average H: 80.34) in this wide range of λ. We set $\lambda = 5$ by default.

Compute cost. Table 7 compares the inference cost for different tasks in terms of # parameters, GFLOP and FPS. For the few-shot classification task, we compare AAPE with three types of prompt learning methods. CoOp and OGEN are the first type of methods that learn fixed text prompts. The efficiency benefits of these methods are evident: the number of learned parameters is small (2k), and high inference speed (FPS) can be achieved without requiring a forward pass to predict adaptive prompts for every input image. MaPLe and PromptSRC belong to the multimodal prompting methods that learn prompts for both text and image. These methods have comparable GFLOP and FPS with fixed prompt learners but have much more parameters to learn, thus risk generalization with sub-optimal accuracy for new classes.

CoCoOp and our AAPE both learn input-adaptive text prompts, with reasonable parameter count and GFLOP. However, they suffer from low FPS because of the input-conditional prompt prediction. This low speed also translates to the learning stage. The training time (min) for AAPE and CoCoOp are 41.92 and 39.53 respectively, in comparison to 10.08 of CoOp. Despite the equally low time efficiency, AAPE outperforms CoCoOp drastically in accuracy, and scales much better with model size than CoCoOp-style methods. To show this, we implement a CoCoOp† baseline that has a similar parameter count with AAPE. As expected, CoCoOp† has lower speed than CoCoOp. CoCoOp† is also found to have lower new class accuracy, *i.e.*, worse generalization.

Table 8: **Few-shot classification in the domain generalization setting.** Note our AAPE follows CuPL to query an LLM to obtain natural language prompts, but further learns from those prompts.

		Source	Target			
		ImageNet	ImageNetV2	ImageNet-Sketch	ImageNet-A	ImageNet-R
Zero-shot	CLIP [40]	66.73	60.83	46.15	47.77	73.96
Prompt learning w/o language priors	MaPLe [20]	70.72	64.07	49.15	50.90	76.98
	CoCoOp [68]	71.02	64.07	48.75	50.63	76.18
	PromptSRC [21]	71.27	64.35	49.55	50.90	**77.80**
	CoOp [67]	71.51	64.20	47.99	49.71	75.21
	CLIPood [45]	71.60	64.90	49.30	50.40	77.20
	RPO [25]	71.67	65.13	49.27	50.13	76.57
	UPT [64]	72.63	64.35	48.66	50.66	76.24
	TaskRes [62]	73.07	65.30	49.13	50.37	77.70
Basic prompts	LASP [7]	71.10	63.96	49.01	50.70	77.07
	KgCoOp [58]	71.20	64.10	48.97	50.69	76.70
	ProGrad [69]	72.24	64.73	47.61	49.39	74.58
LLM-based	CuPL [39]	68.86	63.14	47.85	48.63	75.11
	AAPE	**73.56**$_{\pm 0.12}$	**65.97**$_{\pm 0.18}$	**50.12**$_{\pm 0.23}$	**51.62**$_{\pm 0.22}$	77.52$_{\pm 0.14}$

When we turn our attention to complex vision-language tasks beyond classification, AAPE shines in both efficiency and performance. Note in the tasks of text retrieval, image captioning and VQA, AAPE is used as a standalone image captioning embedding (without projection g) to provide rich language priors. As shown in Table 7 (last row), this setup involves fewer parameters than the classification setting (33k vs. 82k), and leads to slightly higher inference speed. But on the vision-language tasks, AAPE not only achieves SOTA performance (Table 3) but also has higher efficiency than prior works, *e.g.*, about 2.8/1.2 times faster than MAGMA [11] for training/inference.

In summary, AAPE is designed to be a universal text embedding directly applicable to various vision-language tasks. This is not possible with most prompting methods designed for classification, only that AAPE's generality sacrifices the efficiency in the classification task. We leave as future work to speed up AAPE inference in the classification setting, *e.g.*, via pruning or distillation techniques to simplify the forward pass of prompt prediction.

D Results of Few-Shot Classification under Domain Generalization

Table 8 shows our approach is robust to the different types of domain shifts on 4 ImageNet variants. Overall, our AAPE outperforms prior works on all but the ImageNet-R dataset (where AAPE is still competitive with the state-of-the-art PromptSRC method). AAPE outperforms most prompt learners that do not leverage any language priors, sometimes by a large margin. When compared to the prompt learners using basic prompt templates, AAPE shows notable gains thanks to the rich knowledge contained in LLMs. The zero-shot CuPL method is based on LLM too, but lags far behind due to the lack of learning components for downstream adaptation.

E Comparison with Recent Prompt Learners ProText and ArGue-N

Table 9 compares AAPE with two recent prompt learning methods on the few-shot classification task. The compared methods ProText and ArGue-N are closely related to AAPE since the former similarly learn text prompts from LLM-generated image prompts or visual attributes. However, both ProText and ArGue-N learn fixed text prompts, and are not adapted to input image. By contrast, AAPE is input-adaptive during both prompt aggregation and prompt prediction – recall that our prompt aggregation is aligned with the input image, and AAPE prediction is image-conditional.

We conjecture that our input-adaptive framework makes better use of LLM's textual knowledge. The resulting image-aligned AAPE also promotes image-text alignment, giving rise to improved optimization during prompt learning. Table 9 confirms this with better performance of AAPE on seen class data from the base split or source dataset. Input-adaptive AAPE improves generalization too,

Table 9: **Comparison with ProText and ArGue-N** for few-shot classification in both settings of base-to-new class generalization and domain generalization.

	Base-to-New Generalization			**Domain Generalization**				
	Avg across 11 datasets			Source	Target			
	Base	New	H	ImageNet	-V2	-Sketch	-A	-R
ProText [22]	72.95	76.98	74.91	70.22	63.54	49.45	51.47	77.35
ArGue-N [49]	83.77	**78.74**	**81.18**	71.84	65.02	49.25	51.47	76.96
AAPE (ours)	**84.72**	77.54	80.97	**73.56**	**65.97**	**50.12**	**51.62**	**77.52**

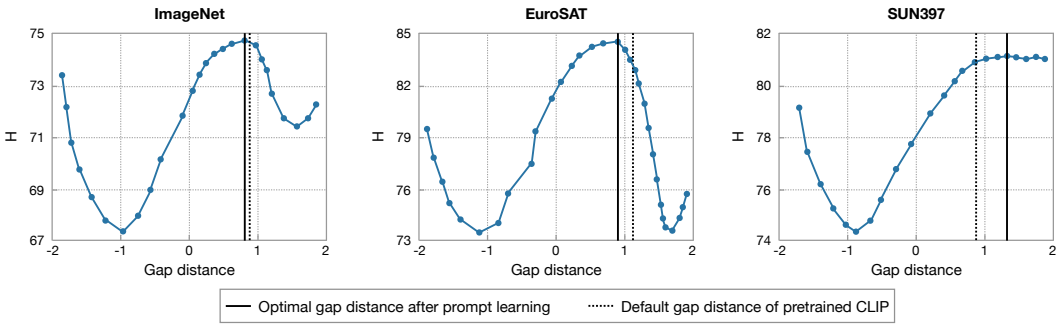


Figure 8: **Optimal gap distance obtained by AAPE learning** on the few-shot classification task (base-to-new class generalization setting). Y axis indicates the Harmonic mean (H) of base and new class accuracies on each dataset. X axis indicates the gap distance varied by shifting the image and text embeddings following [30]. We see the optimal gap distance can be increased (on SUN397) or decreased (on ImageNet and EuroSAT) over the default gap obtained from pretraining.

achieving comparable or stronger generalization performance in two generalization settings than the fixed prompt learners. More importantly, AAPE can be used as a standalone captioning vector in various vision-language tasks beyond classification. This is not possible with ProText or ArGue-N.

F Additional Image Captioning Results

Fig. 9 shows example captioning results on the NoCaps dataset. We compare LiMBeR [35] with and without our AAPE learned on COCO and transferred zero-shot to NoCaps. Thanks to the textual knowledge encoded in AAPE, it often augments the image features to generate more descriptive captions even when the visual cues are ambiguous (*e.g.*, in the first image, AAPE identifies the sea turtles which can be easily confused with rocks).

G Discussion on the Image-Text Modality Gap for AAPE Learning

In this section, we examine the image-text modality gap, a concept introduced in [30]. We aim to gain insights behind the strong downstream performance and generalization of AAPE-based prompt learning. Is that because AAPE learning can mitigate modality gap?

To answer this question, we first note that AAPE is predicted from input image features $\boldsymbol{x}$ via a conditional prompt generator h. Such h can be viewed as an image-to-text mapping function, which should at least maintain (if not improve) feature alignment between the image and text modalities. For empirical evidence, we measure the average cosine feature similarity for the image-prompt pairs on ImageNet. We find AAPE scores 0.91, slightly higher than that of pretrained CLIP (0.89). This fact makes it seem like AAPE should be able to reduce modality gap. Here we perform a systematic analysis following [30], which defines modality gap as the *Euclidean distance between the centers of image and text features*.

Fig. 8 shows the analysis results for few-shot classification on three example datasets. As expected, after AAPE-based prompt learning, the optimal gap distance that attains maximum accuracy deviates

from the default gap distance of pretraind CLIP. We see the optimal gap is increased on SUN397 dataset, while decreased on ImageNet and EuroSAT datasets. In fact, the optimal gap is reduced on 7 out of a total of 11 datasets and moderately increased on the remaining 4. On the other hand, AAPE consistently improves classification accuracy on each dataset, see Table 2 and Fig. 4. This suggests that **modality gap is not highly correlated with downstream generalization**, which is in line with one of the main arguments in [30] that **good generalization does not necessarily need a reduced modality gap**.

Then why does AAPE generalize when the modality gap is not reduced? We hypothesize that our multi-task learning reshapes the loss landscape in a way that encourages generalizable solutions. Specifically, we optimize two loss functions for AAPE: the task loss that (over)fits the seen class data, and the distillation loss that moves AAPE closer to some aggregation of external text knowledge. We find the distillation loss often promotes generalization (Fig. 4) and avoids overfitting caused by the task loss. Hence the two losses could move AAPE in opposite directions, modifying the modality gap differently on the loss landscape. As shown in Fig. 8, the gap change (increase or decrease) is highly dependent on the image-text distribution on the considered dataset.

Note the above optimization perspective is not limited to classification. We can use our hypothesis to similarly explain AAPE's good generalization in complex vision-language tasks like VQA where a multi-task loss is used (distillation + task loss). We leave as future work to study 1) how multi-task learning affects modality gap dynamically and 2) the relationship between modality gap and downstream generalization.

H Broader Impact

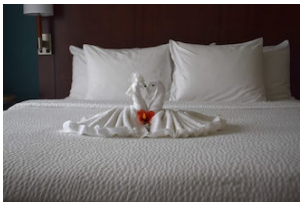
The main contribution of this work is the use of text-based knowledge to improve the downstream generalization of CLIP. The textual knowledge is distilled from either human-annotated image captions or LLM-generated natural language prompts. Such knowledge significantly improves CLIP's downstream performance but carries potential societal impacts. Specifically, when the image captions/prompts reflect (unintentional) biases, our distillation and learning methods could inherit or amplify these biases in the learned feature embeddings. This would potentially lead to perpetual unfair or discriminative outcomes in a variety of vision-language tasks and more critical applications such as AI-driven planning and decision-making.

Ground Truth:
"Sea turtles lie on the beach while a wave pushes in the background."
LiMBeR:
“Hard rocks by the beach."
LiMBeR+AAPE:
"Sea turtles are on the sand of the beach."

Ground Truth:
"A laboratory contains lots of scientific equipment, rolling chairs, and long black countertops.”
LiMBeR:
“An office with no people in it.”
LiMBeR+AAPE:
"Laboratory with black countertops and medical equipment.”

Ground Truth:
"Crowd of people in bright wigs and clothes."
LiMBeR:
“A group of people gathering together.”
LiMBeR+AAPE:
“A group of people dressed up in colorful clothes.”

Ground Truth:
"A lone camel is standing on a hill in the desert sand."
LiMBeR:
“A camel is standing on top of a sand hill.”
LiMBeR+AAPE:
“A camel is walking on a sand hill ahead of a plant.”



Ground Truth:
"A neatly made bed with swans made out of linens on it."
LiMBeR:
“Two swans sitting on a bed.”
LiMBeR+AAPE:
“A bed with swans made of towels.”

Ground Truth:
"A white jack o lantern is sitting on the hay with other orange pumpkins around."
LiMBeR:
“A group of pumpkins are smiling.”
LiMBeR+AAPE:
“A lit and smiling pumpkin sitting on stacks of hay.”

Ground Truth:
"A group on people on a paddle boat with lifejacket on a river."
LiMBeR:
“A group of people in the river.”
LiMBeR+AAPE:
“A group of people paddling against the waves on a river.”

Figure 9: **Captioning results on NoCaps dataset**: LiMBeR vs. LiMBeR+AAPE (zero-shot).

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 main claims are stated in both Abstract and Introduction sections, and each claim is supported by experimental results. We further enumerate our contributions in the Introduction 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: We discuss the limitations and future work in the Conclusion section 6.

   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: [NA]

Justification: No theoretical results are included in the paper.

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: Experimental setup and dataset details are included in Section 4.

   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: Section 4 describes the experimental setup, dataset and implementation details to run and reproduce experiments.

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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 describes all the experimental setup, dataset and training/testing details.

   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: We report the average result and the standard deviation of three runs with random seeds for our experiments/ablations on image classification and text retrieval. Please refer to Table 1, 2, 4, 5, 6 and 8.

   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: This information is included in Section 4 and Appendix C.

   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: We confirm that the work presented in this paper is performed in a manner consistent with NeurIPS Ethics Guidelines.

   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: [Yes]

    Justification: We discuss the broader impact of our work in Appendix H.

    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: Our work presents a finetuning algorithm of CLIP models so this question is not applicable.

    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: We include this information in experimental details in Section 4.

    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` 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: This 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: This 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: This 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.