Amortized Prompt: Guide CLIP to Domain Transfer Learning

Xin Zhang\textsuperscript{1}, Yusuke Iwasawa\textsuperscript{1}, Yutaka Matsuo\textsuperscript{1}, and Shixiang Shane Gu\textsuperscript{1,2}

\textsuperscript{1} The University of Tokyo
\textsuperscript{2} Google Brain
xin@weblab.t.u-tokyo.ac.jp

Abstract. Domain generalization (DG) is a problematic Domain Transfer Learning problem aiming to learn a generalizable model to unseen domains. Recent massive pre-trained models such as CLIP and GPT-3, i.e. foundation models (FMs), are robust to many distribution shifts and therefore should lead to substantial improvements in DG. In this work, we study generic ways to adopt CLIP for DG problems in image classification. We evaluate Test-Time Adaptation (TTA) and full DG learning settings on several standard benchmarks. We propose AP (Amortized Prompt) as a novel prompt strategy for domain inference in the form of prompt generation. Moreover, we show that combining domain prompt inference with CLIP enables the model to outperform strong DG baselines and other prompt strategies. Since AP generate prompt to automatically adapt to the target domain, it can be seen as a TTA method. Therefore, we also conduct a fair comparison with SOTA TTA methods. The results demonstrate AP can outperform all baselines with a significant margin. Then, we further analyze the properties of AP with insightful ablation experiments. We hope the simplicity and success of our approach emphasize the importance of and lead to broader adoption and analysis of foundation models in the field of TTA and DG.

Keywords: Domain Generalization, Test Time Adaptation, Unsupervised Domain Adaptation, Foundation Model

1 Introduction

Transfer Learning is practical in grounded projects of AI technologies. Where pre-train then fine-tune is a popular paradigm: many natural language processing (NLP) applications are based on BERT \cite{BERT}, and many computer vision (CV) applications are based on pre-trained ResNet \cite{ResNet} or Vision Transformer (ViT) \cite{ViT}. However, in practical application scenarios, domain shifts pose a substantial challenge for successfully transferring models \cite{DomainShift}. Many works in domain adaptation (DA) \cite{DomainAdaptation,DomainAdaptation2} and domain generalization (DG) \cite{DomainGeneralization,DomainGeneralization2,DomainGeneralization3} have been made to address this domain shift issue, but it remains an unsolved and critical problem.

Transfer Learning is practical in grounded projects of AI technologies. Where pre-train then fine-tune is a popular paradigm: many natural language processing (NLP) applications are based on BERT \cite{BERT}, and many computer vision (CV) applications are based on pre-trained ResNet \cite{ResNet} or Vision Transformer (ViT) \cite{ViT}. However, in practical application scenarios, domain shifts pose a substantial challenge for successfully transferring models \cite{DomainShift}. Many works in domain adaptation (DA) \cite{DomainAdaptation,DomainAdaptation2} and domain generalization (DG) \cite{DomainGeneralization,DomainGeneralization2,DomainGeneralization3} have been made to address this domain shift issue, but it remains an unsolved and critical problem.
The concept of Amortized Prompt. We assume that the domain information could provide a model with some hints to solve the task. For example, in this case, ‘cartoon’ or ‘art painting’ information. We use them as prompts for our CLIP model. At first, we allow the model to learn how to extract the domain feature on the source domain. Ideally, it might be the style information. Then we test the model on the unseen target domain. We suppose the model can generate a prompt by extracting the information related to ‘sketch’. The goal of Amortized Prompt is to generate the appropriate prompt in an unseen domain.

(CV) applications are based on pre-trained ResNet [21] or Vision Transformer (ViT) [11]. However, in practical application scenarios, domain shifts pose a substantial challenge for successfully transferring models [43]. Many works in domain adaptation (DA) [37,53], and domain generalization (DG) [55,52,43] have been made to address this domain shift issue. However, it remains an unsolved and critical problem. First, we consider naively adopting CLIP to domain generalization. However, it is difficult and unclear how to design an appropriate prompt for an unknown target domain, though a better prompt could improve the performance [38]. We assume a prompt that includes domain-specific information could help the model perform in domain generalization. Then, based on these insights, we propose Amortized Prompt (AP) to amortize the prompt for the target domain. AP is supposed to generate domain-specific prompts by extracting domain-specific information from input images. Ideally, AP can generate prompts on arbitrary target domain and avoid the uncertain prompt engineering of CLIP for unknown domain adaptation.

To evaluate AP, we conduct experiments on four standard datasets and strictly follow the experiment setup in [20,23], such as parameter tuning and model selection. First, we show that CLIP + AP outperforms the strong DG baselines. Then, we conducted a fair comparison with the exiting State-Of-The-Art Test-Time Adaptation (TTA) methods. Since CLIP + AP is always deployed with a batch of data, which can be seen as a TTA method. Additionally, we con-
duct numerous ablation experiments, including various backbones and frozen backbones. All of these results confirm the effectiveness of CLIP + AP.

In summary, our main contributions are:

1. We introduce a foundation model (CLIP) to domain generalization and show its solid empirical performances.
2. We propose a novel domain-prompt inference architecture to more effectively apply CLIP for Domain Transfer Learning, including DG and TTA.
3. We conduct experiments on various standard DG datasets, compare with a series of strong baselines and SOTA TTA methods and show our Amortized Prompt outperforms previous approaches.

2 Related Work

2.1 Domain Generalization

Over the past decade, various studies proposed different approaches to solve DG. Most prior works focus on regularizing the model using the knowledge from multiple source domains. For example, domain-invariant representation learning [17] is a significant branch of domain generalization, aiming to reduce domain gaps in the space of latent representations. There are many different approaches to measure the domain gaps, including adversarial classifier [31,16,17], kernel mapping [5,19], metric learning [36,24], and invariant risk minimization [1]. Similarly, several works try to generate samples with diverse styles so that models can learn domain-invariant features through different domains [42,57,7]. Other methods use meta-learning to learn how to regularize the model to improve robustness [12,30].

Our work investigates the importance of CLIP [38] in DG and proposes a lightweight module to adopt the CLIP for DG. Several recent observations motivate us to benchmark CLIP in the DG setup. First, [20] shows that many prior approaches do not provide significant improvement compared to simple supervised learning. The results imply that regularizing the model is not enough to achieve high performance in DG. Secondly, despite significant works on this front, most studies focus on medium-scale pre-trained models such as ResNet18 or ResNet50. Very large-scale models often lead to substantial improvements. Notably, the latest work [23] compares more large-scale backbone networks, including big transfer [26] (BiT-M-R50x3, BiT-M-R101x3, and BiT-M-R152x4), vision transformer (ViTB16 and ViT-L16 [11], Hybrid ViT, DeiT [47]), and MLP-Mixer [46] (Mixer-L16), and show that the selection of backbone networks does indeed matter in DG. Differently from [23], we show that CLIP performs surprisingly well without fine-tuning the entire model in source domains, which is time-consuming in practice.

From the methodological point of view, our work relates to several prior works [17,57,7] that tried leveraging domain features rather than discarding them as in domain-invariant learning. However, our work differs in that we propose
a CLIP-specific way to leverage the domain features by combining these features with prompt tuning. Our work also relates to [23] where both approaches modulate their prediction given unlabeled data available at test time. Specifically, [23] proposes T3A that replaces the linear classifier using pseudo-labelling and prototypical classification and shows that it stably improves the performance in unseen domains. However, T3A cannot be directly applied to CLIP, as it assumes a simple linear classifier that CLIP does not employ.

2.2 Prompt Tuning

The success of GPT-3 demonstrated the importance of prompt tuning. There are various prompting strategies, such as discrete natural language prompts and continuous prompts, and many other prompting strategies have appeared [34].

Many works focus on learning from discrete natural language prompts. AutoPrompt [44] elicits knowledge from language models with automatically generated discrete prompts. PADA [4] proposed a domain adaptation algorithm that trains T5 [39], a language foundation model, to generate unique domain-relevant features for each input. The motivation of PADA is similar to AP, but with a discrete prompt for NLP applications, and our AP with continuous prompt in computer vision.

Lately, there have been many works [32, 28] directly tuning prompts in continuous vector forms. Because often, the primary purpose of prompt tuning is to extract knowledge from language models and not keep interpretability for humans. In another recent work, P-Tuning v2 [35] shows that continuous prompt tuning achieves the same performance as fine-tuning in various settings.

Due to the successful applications of CLIP, prompt tuning is also of great interest in computer vision. For example, CoOp (Context Optimization) [56] demonstrated that the performance of CLIP is highly sensitive to prompts and that a suitable prompt can improve performance for the image recognition task. Therefore, CLIP-Adapter [18] was proposed to learn with an additional adapter network.

In this work, we first introduce CLIP to DG for image recognition. Then, we propose a prompt tuning method that involves predicting the domain labels. Unlike CoOp and CLIP-Adapter, which need class labels to tune prompts, AP generate domain prompts from input images as a simple way to amortize prompt inference to an unseen domain.

3 Method

In this section, we first introduce the notations and definitions of domain generalization following [52]. Then, we explain how to use CLIP in DG and introduce Amortized Prompt to effectively enhance CLIP’s performance in DG.
3.1 Problem Setup of Domain Generalization

Let $X$ denote an input space and $Y$ an output space. A domain is composed of data that is sampled from a distribution. We denote the datasets from a distribution as $S^i = (x^i_j, y^i_j)_{j=1}^{n_i} \sim \mathcal{P}_{XY}^i$, where $x \in X \subset \mathbb{R}^d$ is an input image, $y \in Y$ denotes the class associated with $x$, and $\mathcal{P}_{XY}^i$ denotes the joint distribution of the sample and output label in domain $i$. $X, Y$ denote the corresponding random variables.

In domain generalization, we are interested in the performance of predictor $h$ on data from an unseen domain $S^t$. To achieve the goal, prior works fine-tune a pre-trained image encoder $f$ (usually ResNet18 or ResNet50) in conjunction with randomly initialized classification head $g$ (linear classifier) using data from multiple different datasets. Specifically, given $M$ datasets $S^i$ collected from several different domains $i \in \{1, \cdots, M\}$, the $f$ and $g$ is updated by

$$\min_{f,g} \frac{1}{M} \sum_{i=1}^{M} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(g \circ f(x^i_j), y^i_j),$$

where $\ell(\cdot)$ is a loss function. In the most simple case, $\ell$ is a simple cross-entropy loss, then minimizing eq. 2 is called empirical risk minimization (ERM). As discussed in Section 2.1, different methods in DG use other loss functions by designing regularization terms to prevent overfitting to specific domains. These datasets are often denoted as source domains and distinguished as target domain where we want the model to work well.

3.2 CLIP for DG

This paper adapts CLIP in domain generalization setups. As usual, CLIP is consists of two parts: an image encoder $f_{clip}$ and an language model $g_{clip}$. There are two notable differences between our works (using CLIP in DG) and the convention of DG. First, the image encoder $f_{clip}$ is pre-trained on massive datasets. We are interested in how such massively trained models are robust to unknown distribution shifts that may not be included in the training dataset. Second, CLIP classifies the image features based on the similarity between embedding of a text prompt $p$, such as ‘dog’ or ‘class label’, rather than using the classification head trained from scratch. Specifically, given an image $x$ and $K$ class prompt $p_k$, CLIP output a prediction using both $f_{clip}$ and $g_{clip}$:

$$\hat{y}_{clip} = \arg \max_k \langle f_{clip}(x), g_{clip}(p_k) \rangle$$

where $K$ is the number of category and $\langle \cdot, \cdot \rangle$ is cosine similarity.

It is worth noting that we fixed $f_{clip}(\cdot)$ and $g_{clip}(\cdot)$ during whole experiments. In other words, we evaluated CLIP in a zero-shot manner, rather than fine-tuning backbone networks in source domains as with prior works. Instead, we change the text prompt $p$ by the class labels used in each dataset. By doing so, we
Fig. 2. The architecture of CLIP + AP. At first, encode the input images to get image embeddings with the pre-trained CLIP’s image encoder $f(\cdot)$. Feed the image embeddings into domain prompt generator $F(\cdot)$ to generate domain prompt embeddings. At the same time, encoder all kinds of labels with the pre-trained CLIP’s language encoder $g(\cdot)$ to get label prompt embeddings. Then, add label embeddings with domain prompt embeddings to generate amortized prompt embeddings. Since these prompt embeddings could be automatically generated from input images and labels, we call them Amortized Prompt Embeddings. Finally, to get the output in probability, we calculate the cosine similarity $\langle \cdot, \cdot \rangle$ with Image Embeddings and Amortized Prompt Embeddings. We only train the prompt generator $F(\cdot)$, which is coloured as blue.

show how powerful the representation of massively pre-trained models (CLIP) for DG setup is, without any additional computational costs to re-train such large models entirely. In addition, we propose a novel way to design the prompt $p$ to improve the performance in an unseen dataset without fine-tuning the entire model.

3.3 AP: Amortized Prompt for CLIP in DG

As discussed in Section 2.2, designing a prompt is a powerful way to improve the performance of transformer-based models. Not only it is powerful, but it should be easier to train since the dimension of prompts is overwhelmingly smaller than the entire parameters of $f$ and $g$. For example, suppose we can access to supervised dataset from the target domain. In that case, we can optimize a prefix vector $p_{pre}$ by simple supervised loss:

$$
\min_{p_{pre}} \mathbb{E}_{x,y \sim \mathcal{S}} \ell(\hat{y}_{clip*}, y),
$$

where $\hat{y}_{clip*}$ is

$$
\hat{y}_{clip*} = \arg \max_k \langle f_{clip}(x), g_{clip}(p_{k}^*) \rangle,
$$
where $\mathbf{p}_k^i$ is a concatenation of trainable parameters $\mathbf{p}_{pre}$ and $\mathbf{p}_k$. Note that, $g_{clip}$ outputs the fixed length vector regardless of the input dimension (i.e., size of $\mathbf{p}_k$). The size of $\mathbf{p}_k$ is a hyperparameter.

Unfortunately, such labelled training data is unavailable in DG for the target domain. We proposed AP to replace the optimization process of $\mathbf{p}_{pre}$ in each domain, by training of novel prompt generators $F(\cdot)$ that generate a prompt $\mathbf{p}_{pre}$ given small unlabeled images from a distribution. Specifically, we use a fully connected network $F(\cdot)$ to generate a prompt $\mathbf{p}_{ap}$ from input images:

$$\mathbf{p}_{ap}^i = \frac{1}{N} \sum_{j=1}^{N} F(f(x_j^i)), \quad (5)$$

where $N$ is batch size for each domain and $x_j^i$ denote the images from $i$-th distribution. Given a batch of data from multiple source distributions, we optimize $F$ using the following loss function:

$$\min_{F} \frac{1}{M} \sum_{i=1}^{M} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(\hat{y}_{ap}^i, y_j^i), \quad (6)$$

and

$$\hat{y}_{ap}^i = \arg \max_k \langle f_{clip}(x^i), g_{clip}(\mathbf{p}_k^i) \rangle, \quad (7)$$

where $\mathbf{p}_k^i$ is a concatenation of pre-defined $\mathbf{p}_k$ and $\mathbf{p}_{ap}^i$. We show the architecture of CLIP + AP in figure 8.

4 Experiments

This section shows that CLIP + AP outperforms the strong Domain Generalization (DG) and Test-Time Adaptation (TTA) baselines on four standard datasets. First, we clarified the datasets, hyperparameters and model selection strategy, and the details of implemen ts. Then, we demonstrate that CLIP + AP outperforms the effective existing DG and TTA SOTA methods. Finally, our ablation experiments, including variants backbone comparison and different prompt strategies study, show the efficiency of CLIP + AP.

Datasets We choose four real-world datasets from the DomainBed benchmark and we show the more examples in Appendix A. VLCS [14] gathers four photographic datasets $d \in \{\text{Caltech101} [15], \text{LabelMe} [40], \text{SUN09} [9], \text{VOC2007} [13]\}$, containing 10,729 samples of 5 classes. PACS [29], comprises four domain datasets $d \in \{\text{art, cartoons, photos, sketches}\}$, with 9,991 samples and 7 classes. Office-Home [49], includes domains $d \in \{\text{art, clipart, product, real}\}$, with 15,588 samples and 65 classes. TerraIncognita [3] includes a photo of wild animals taken by a camera at different locations. Following [20], we used datasets of $d \in \{\text{Location 100, Location 38, Location 43, Location 46}\}$, with total 24,788 samples and classes.
| DomainBed          | VLCS    | PACS    | OfficeHome | Terra   | Avg   |
|-------------------|---------|---------|------------|---------|-------|
| ERM (CLIP)        | 82.7 ± 0.3 | 92.9 ± 1.9 | 78.1 ± 2.1 | 50.2 ± 1.7 | 75.9  |
| CORAL             | 82.0 ± 0.2 | 93.2 ± 1.1 | 78.9 ± 1.9 | 53.5 ± 0.7 | 76.9  |
| DANN              | 83.2 ± 1.2 | 93.8 ± 1.3 | 78.8 ± 1.1 | 52.2 ± 2.0 | 77.0  |
| CLIP*             | 76.6 ± 0.0 | 95.8 ± 0.1 | 79.9 ± 0.1 | 36.4 ± 0.1 | 72.2  |
| CLIP* (template)  | 82.3 ± 0.1 | 96.1 ± 0.1 | 82.3 ± 0.2 | 34.1 ± 0.1 | 73.7  |
| CLIP* (domain name)| 78.8 ± 0.2 | 96.6 ± 0.1 | 81.9 ± 0.1 | 28.4 ± 0.2 | 71.4  |
| CLIP* + AP (ours) | 84.3 ± 0.4 | 97.3 ± 0.2 | 84.2 ± 0.2 | 52.6 ± 0.6 | 79.6  |

Table 1. Comparison experiments on VLCS, PACS, OfficeHome, TerraIncognita. We bold the best results and underlined the second-best results. * means w/o the fine-tuning image encoder.

Hyperparameters and model selection. We setup experiments on DomainBed \(^3\), and implemented AP based on CLIP \(^4\). Since the model selection affects performance significantly, we strictly follow the basic selection criterion \([20]\). We select hyperparameters by standard training-domain validation, which uses the subset of each training domain to choose a model \([20]\). We pool the subsets of each training domain together. Then split the data of each domain into 80% and 20% for the training model and selecting hyperparameters. We randomly searched 20 trials over a joint distribution of all hyperparameters. Then, we run three trials of each hyperparameter setting and hold out one domain for testing and the rest for training. Finally, we select the hyperparameters that maximize validation accuracy over the training domains and report final accuracy averaged over all three trials.

Detail of the implements As shown in architecture figures, the network we need to train is only the prompt generator network, in contrast with the fine-tuned source domains with the classification loss. We implement a multilayer perceptron network (MLP) and use Stochastic Gradient Descent \([8]\) with momentum as optimizer. We apply the transforms proposed in CLIP for each image instead of the data augmentation used in DomainBed. We report more detail of implements in Appendix B, and please check our source code.

4.1 Comparison with existing DG methods

Baselines We used ERM, CORAL and DANN as our baselines. As \([20]\) pointed out, Empirical Risk Minimization (ERM) \([48]\) is a strong DG baseline when the experiments are under a unified specification. We used the class name as a standard prompt for CLIP for the prompt strategy, such as ‘cat’ and ‘dog’. We also adopt a template prompt ’a photo of a class label.’, since previous

\(^3\) [https://github.com/facebookresearch/DomainBed](https://github.com/facebookresearch/DomainBed)

\(^4\) [https://github.com/openai/CLIP](https://github.com/openai/CLIP)
Table 2. Detailed results on VLCS, PACS, OfficeHome, TerraIncognita. The domain names are from original datasets folders and used in previous works. * means w/o the fine-tuning image encoder. (1) refers to the zero-shot adaptation of CLIP*. (2) refers to adaption AP on CLIP*. (2) - (1) means the improvement of adaption AP on CLIP*. We highlight the most improved result for each dataset.

Table 1 shows that CLIP without fine-tuning the image and language encoders can achieve an average of 72.2% accuracy. Using a template can improve accuracy to 73.7%. Using the domain name as a prompt can improve accuracy in VLCS, PACS and OfficeHome, but not in TerraIncognita. We consider that the Terra dataset’s domain name is entirely unusable information for classification. Relatively our AP achieve 79.6% accuracy, even beyond the ERM, CORAL and DANN, which fine-tuning image encoder. This result supports our hypothesis that the prompt that includes domain information can steadily improve CLIP accuracy in the unseen domain. Especially in the case of uncertain
domain information like TerraIncognita, AP substantially outperforming other prompt strategies. This property is desirable in Domain Generalization.

**Fig. 3.** The result of each method comparison with CLIP*. Here, we used blue for the methods training and red for freezing their backbones. CO means CORAL, DA means DANN. +T and +D mean using the template and the domain class as prompt for CLIP*. +AP means using Amortized Prompt for CLIP*. * means w/o the fine-tuning image encoder.

**Table 2** shows the detailed performance of AP in different domains of each standard dataset. **Figure 3** visualized the score of each method could improve from CLIP*. We observe that AP ourperforms all baselines and is the only method that steadily improves CLIP* on all datasets. The surprising thing is that even ERM, CORAL and DANN fine-tuning their image conders, the accuracies on PACS and OfficeHome are not better than CLIP*. We discuss more details in the ablation section. Moreover, we found that both template prompt and domain name prompt perform worse in TerraIncognita, in which domain shift is hard to be verbalized (as shown in **Figure 4**). Since AP suppose to extract domain-specific features from input images, the performance of AP is as good as ERM, CORAL and DANN that conduct fine-tuning. Based on these results, we propose that AP should be an effective method in Domain Generalization.

**Fig. 4.** The samples from each domain of each dataset. We show the images labelled as ‘dog’ in VLCS and PACS, ‘toys’ in OfficeHome and ‘birds’ in Terra (or TerraIncognita). Each example is the first image by the default order. Please refer to the Appendix A for details.
Table 3. Comparison with TTA methods. Here, ⋆ means updating the linear classifier, and ⋆⋆ means updating the feature extractor to minimize entropy reported in table 3 of the T3A paper. We bolded the best results and underlined the second-best results. Note that all experiments in this table are conducted on a cluster of A100 GPUs for speeding up. The numbers of ERM, CORAL, ours (+AP) are different from them reported in the Table 1 due to the different devices. (Refer to Appendix C for details).

| Methods         | VLCS     | PACS     | OfficeHome | Terra  | Avg  |
|-----------------|----------|----------|------------|--------|------|
| ERM             | 81.4 ± 0.3 | 91.9 ± 0.7 | 78.4 ± 1.1 | 47.8 ± 3.1 | 74.9 |
| +T3A            | **82.2 ± 0.1** | **88.2 ± 0.0** | 76.9 ± 0.9 | 48.2 ± 3.2 | 73.9 |
| +Pseudo Label   | 81.1 ± 1.0 | 92.1 ± 0.4 | 78.0 ± 2.5 | **50.5 ± 4.4** | 75.4 |
| +Pseudo Label⋆  | 82.1 ± 0.4 | 87.6 ± 0.0 | 76.6 ± 1.9 | 46.9 ± 3.1 | 73.3 |
| +SHOT           | **82.2 ± 0.4** | 87.8 ± 0.0 | 76.5 ± 1.2 | 46.7 ± 3.2 | 73.3 |
| +SHOT⋆⋆         | 80.4 ± 0.4 | 93.8 ± 0.8 | 80.7 ± 0.8 | 40.5 ± 1.7 | 73.9 |
| CORAL           | 81.2 ± 0.3 | 91.1 ± 1.9 | 78.7 ± 0.8 | 48.6 ± 2.9 | 74.9 |
| +T3A            | 80.8 ± 0.5 | 91.2 ± 1.9 | 79.1 ± 0.9 | 49.0 ± 3.0 | 75.0 |
| +Pseudo Label   | 80.0 ± 1.4 | 93.1 ± 2.0 | 79.8 ± 1.2 | 44.5 ± 3.3 | 74.4 |
| +Pseudo Label⋆  | 81.4 ± 0.1 | 91.2 ± 1.9 | 78.8 ± 0.9 | 48.6 ± 2.9 | 75.0 |
| +Tent⋆          | 81.3 ± 0.2 | 91.2 ± 1.9 | 78.6 ± 0.8 | 48.7 ± 2.6 | 75.0 |
| +SHOT           | 78.7 ± 1.9 | 93.0 ± 1.2 | 80.7 ± 0.9 | 41.9 ± 2.0 | 73.6 |
| +SHOT⋆⋆         | 78.5 ± 2.0 | 93.1 ± 1.1 | **80.7 ± 0.9** | 41.9 ± 2.0 | 73.5 |
| +AP             | 81.0 ± 1.1 | **95.9 ± 0.0** | **82.3 ± 0.7** | **49.4 ± 1.1** | **77.2** |

4.2 Comparison with existing Test-Time Adaptation methods

In practice, we deploy AP on a batch of data at both training and testing time due to the expensive computational resources. In this case, AP should be considered as a Test-Time Adaptation algorithm. Therefore, we performed a fair comparison with several existing algorithms to validate AP.

Baselines We adopt the baselines used in the T3A paper, such as Pseudo Label [27], SHOT [33], Tent [50], including T3A [23]. Our implement is based on [23], with 64 batch sizes during Test-Time. We train all models with the same backbone, ViT-B16, provided by CLIP [38]. Note again that all experiments are carried out according to the model selection, hyperparameter selection strategy and evaluation method proposed in T3A and DomainBed.

As shown in Table 3, our AP beats the most effective TTA methods within our knowledge. The result demonstrates AP can consistently improve the model’s generalisation performance at test time. We believe this is sufficient evidence that the main idea of AP, extracting target unseen domain feature to help model adapting at test time, is practical.
| Backbone Model    | VLCS       | PACS       | OfficeHome | Terra       | Avg    |
|------------------|------------|------------|------------|-------------|--------|
| ResNet18†        | 73.2 ± 0.9 | 80.3 ± 0.4 | 55.7 ± 0.2 | 40.7 ± 0.3  | 62.5   |
| ResNet50†        | 75.5 ± 0.1 | 83.9 ± 0.2 | 64.4 ± 0.2 | 45.4 ± 1.2  | 67.3   |
| Mixer-L16†       | 76.4 ± 0.2 | 81.3 ± 1.0 | 69.4 ± 1.6 | 37.1 ± 0.4  | 66.1   |
| BiT-M-R50x3†     | 76.7 ± 0.1 | 84.4 ± 1.2 | 69.2 ± 0.6 | **52.5 ± 0.3** | 70.7   |
| BiT-M-R101x3†    | 75.0 ± 0.6 | 84.0 ± 0.7 | 67.7 ± 0.5 | 47.8 ± 0.8  | 68.6   |
| BiT-M-R152x2†    | 76.7 ± 0.3 | 85.2 ± 0.1 | 71.3 ± 0.6 | 51.4 ± 0.6  | 71.1   |
| ViT-B16†         | 79.2 ± 0.3 | 85.7 ± 0.1 | 78.4 ± 0.3 | 41.8 ± 0.6  | 71.3   |
| ViT-L16†         | 78.2 ± 0.5 | 84.6 ± 0.5 | 78.0 ± 0.1 | 42.7 ± 1.9  | 70.9   |
| DeiT†            | 79.3 ± 0.4 | 87.8 ± 0.5 | 76.6 ± 0.3 | 50.0 ± 0.2  | 73.4   |
| HViT†            | 79.2 ± 0.5 | 89.7 ± 0.4 | 80.0 ± 0.2 | 51.4 ± 0.9  | 75.1   |
| CLIP (ViT-B16)   | 82.7 ± 0.3 | 92.9 ± 1.9 | 78.1 ± 2.1 | 50.2 ± 1.7  | 75.9   |
| CLIP*+AP         | **84.3 ± 0.4** | **97.3 ± 0.2** | **84.2 ± 0.2** | **52.6 ± 0.6** | **79.6** |

Table 4. Results of ERM with various backbone networks on DG benchmark. † means the numbers are taken from Table 2 in [23]. The best scores are bolded, and the second-best scores are underlined. * means w/o the fine-tuning image encoder. The details of each model are shown in Appendix D.

4.3 Backbone Ablation

**Different Backbones** Many proposed DG methods are evaluated with the standard ResNet backbone. However, currently more and more large models are studied, and their validity is experimentally demonstrated [6,51]. Therefore, we reported the performance of ResNet18 and ResNet50, Mixer-16 [46], Vision Transformer (ViT) [11] and several variations of ViT, such as BiT [26], DeiT [47], HViT in Table 4.

From the result, we noticed that the ViT-B16 backbone with Contrastive Language Image Pre-trained (CLIP) on YFCC100M [45] performs as well as HViT. Moreover, CLIP + AP surpassed HViT by a large margin (4.2%) with only a simple MultiLayer Perceptron (MLP) network. This result shows CLIP is a high-quality backbone on DomainBed and can be efficiently enhanced by AP. Note that comparing with different backbones is not fair since different backbones are due to the amounts of parameters, training data, training paradigm and other implemented details.

**Frozen Backbone** Fine-tuning a large model like CLIP or other Foundation Models requires many computing resources. One goal of AP was adapting CLIP to the target domain with minimum computing. We wondered if only training an MLP classifier could help model transfer with freezing the backbone. Here, we conducted a novel ablation experiment based on standard ERM and named the method as Frozen ERM.

From Table 5, we surprisingly found that Frozen ERM outperforms the standard ERM with all different backbones in OfficeHome. This result points to
Table 5. The results of frozen backbone ablation with ERM. Each block represents a backbone. Frozen means using the frozen backbone. † means the numbers are taken from Table 2 in [23]. The highlighted numbers mean the Frozen ERM outperforms the standard ERM with fine-tuning backbones. (3) refers to AP, which scores beat all others and are bolded.

| Backbone          | VLCS  | PACS  | OfficeHome | Terra | Avg  |
|-------------------|-------|-------|------------|-------|------|
| (1) Frozen        | 76.0 ± 0.3 | 66.0 ± 0.7 | 61.7 ± 0.5 | 25.5 ± 1.8 | 57.3 |
| (2) ResNet18†     | 73.2 ± 0.9 | 80.3 ± 0.4 | 55.7 ± 0.2 | 40.7 ± 0.3 | 62.5 |
| (2) - (1)         | -2.8    | +14.3  | -6.0       | +15.2  | +4.2 |
| (1) Frozen        | 77.4 ± 0.3 | 67.2 ± 0.4 | 68.0 ± 0.3 | 35.4 ± 1.5 | 62.0 |
| (2) ResNet50†     | 75.5 ± 0.1 | 83.9 ± 0.2 | 64.4 ± 0.2 | 45.4 ± 1.2 | 67.3 |
| (2) - (1)         | -1.9    | +16.7  | -3.6       | +10.0  | +5.3 |
| (1) Frozen        | 77.5 ± 0.4 | 74.3 ± 0.3 | 77.4 ± 0.2 | 43.4 ± 0.3 | 68.2 |
| (2) DeiT †        | 79.3 ± 0.4 | 87.8 ± 0.5 | 76.6 ± 0.3 | 50.0 ± 0.2 | 73.4 |
| (2) - (1)         | +1.8    | +13.5  | -0.8       | +6.6  | +5.2 |
| (1) Frozen        | 79.2 ± 0.1 | 76.6 ± 0.4 | 81.1 ± 0.2 | 35.7 ± 0.7 | 68.1 |
| (2) HViT†         | 79.2 ± 0.5 | 89.7 ± 0.4 | 80.0 ± 0.2 | 51.4 ± 0.9 | 75.1 |
| (2) - (1)         | -0.0    | +13.1  | -1.1       | +15.7  | +7.0 |
| (1) Frozen        | 82.6 ± 0.3 | 96.9 ± 0.1 | 83.2 ± 0.2 | 46.5 ± 2.1 | 77.3 |
| (2) ViT-B16(CLIP) | 82.7 ± 0.3 | 92.9 ± 1.9 | 78.1 ± 2.1 | 50.2 ± 1.7 | 75.9 |
| (2) - (1)         | +0.1    | -4.0   | -5.1       | +3.7  | -1.4 |
| (3) AP (ours)     | 84.3 ± 0.4 | 97.3 ± 0.2 | 84.2 ± 0.2 | 52.6 ± 0.6 | 79.6 |
| (3) - (1)         | +1.7    | +0.4   | 1.0        | +6.1  | +2.3 |

the critical training strategy in Domain Generalization, including ResNet (18 and 50 layers) backbones. Moreover, a similar insight also can be confirmed in VLCS. On the other hand, AP steadily improving the performance on all datasets demonstrates the robustness of AP.

A similar phenomenon, fine-tuning does not constantly improve performance in Domain Generalization, is also observed in subsection 4.1. In this work, we only evaluated several backbones with different sizes due to the limitation of computing resources. However, we believe that adaptation with minimum computing is a promising area for feature work, significantly when leveraging Foundation Models.

5 Conclusions

We attempted to apply the Foundation Model CLIP to Domain Transfer Learning. For this purpose we proposed a novel approach called Amortized Prompt (AP). By amortizing the unknown target domain prompt conditional on input images, CLIP + AP brings substantial improvements over standard baselines and naive CLIP with other prompt strategies. Then, we evaluated CLIP with the several ef-
fective Test-Time Adaptation methods and conducted many backbone ablation experiments. We verified that AP can stabilize performance and also present meaningful insights about existing datasets and backbone fine-tuning strategy. We hope our work can expand and inspire the roles of foundation models in domain transfer learning.

5.1 Limitation

Label Shift In the technical perspective, our AP cannot capture the domain shift outside of the images, such as label shift included in TerraIncognita. This is because AP uses domain features extracted from images. So AP has no idea to capture such non-visual domain migration. Unfortunately, the label shift exists in the actual-world applications [2].

Social impact perspective We should note that many images and text descriptions of web data are directly used to train CLIP. Though CLIP benefits from low-cost data that without manually labelling, it inevitably leads to a lot of bias and privacy being included in CLIP and other foundation models [6]. This requires us to spend more time paying attention to the opportunities and risks of Foundation Models.

5.2 Future Work

At first, interpretability is essential in both Domain Transfer Learning and Foundation Model. So we are considering generating a interpretable language prompt based on our proposed vector morphology of AP. This direction will provide a better way to understand how AP works in an unseen target domain. Moreover, it might improve generalization capabilities further by utilizing powerful language models.

Secondly, as discussed in subsection 4.1 we found that the impact of the data in Domain Generalization is severe. As contemporaries, similar problems pointed out by [54]. They focused on the methodology and experiments in pre-training, while we worked on the development of new algorithms under a specific specification. At the same time, data-centric AI is also in the spotlight. Therefore, we believe it would be meaningful to conduct more research on domain generalization from data-centric AI.

There has been a lot of research in developing more efficient and practical Domain Generalization datasets [22,25,41]. However, the requirements for datasets based on Foundation Models might shifted from before. Because the traditional data assumption of Domain Generalization that the test domain is definitely unseen will be difficult to guarantee. Since there is hard to ensure what kind of data did Foundation Model used. Therefore, we believe that how to make a benchmark for Foundation Model is still an open question.
References

1. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization (2020) 3
2. Azizzadenesheli, K., Liu, A., Yang, F., Anandkumar, A.: Regularized learning for domain adaptation under label shifts. arXiv preprint arXiv:1903.09734 (2019) 14
3. Beery, S., van Horn, G., Perona, P.: Recognition in terra incognita (2018) 7
4. Ben-David, E., Oved, N., Reichart, R.: Pada: A prompt-based autoregressive approach for adaptation to unseen domains (2021) 4
5. Blanchard, G., Lee, G., Scott, C.: Generalizing from several related classification tasks to a new unlabeled sample. Advances in neural information processing systems 24, 2178–2186 (2011) 3
6. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021) 12, 14
7. Borlino, F.C., D’Innocente, A., Tommasi, T.: Rethinking domain generalization baselines (2021) 3
8. Bottou, L.: Stochastic gradient descent tricks. In: Neural networks: Tricks of the trade, pp. 421–436. Springer (2012) 8
9. Choi, M.J., Lim, J.J., Torralba, A., Willsky, A.S.: Exploiting hierarchical context on a large database of object categories. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 129–136. IEEE (2010) 7
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019) 1
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 1, 2, 3, 12
12. Dou, Q., Coelho de Castro, D., Kamnitsas, K., Glocker, B.: Domain generalization via model-agnostic learning of semantic features. Advances in Neural Information Processing Systems 32, 6450–6461 (2019) 3
13. Everingham, M., Winn, J.: The pascal visual object classes challenge 2007 (voc2007) development kit (2009) 7
14. Fang, C., Xu, Y., Rockmore, D.N.: Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1657–1664 (2013) 7
15. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence 28(4), 594–611 (2006) 7
16. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation (2015) 3
17. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks (2016) 3
18. Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., Qiao, Y.: Clip-adapter: Better vision-language models with feature adapters (2021) 4
19. Grubinger, T., Birlutiu, A., Schöner, H., Natschläger, T., Heskes, T.: Domain generalization based on transfer component analysis. In: International Work-Conference on Artificial Neural Networks. pp. 325–334. Springer (2015) 3
20. Gulrajani, I., Lopez-Paz, D.: In search of lost domain generalization. arXiv preprint arXiv:2007.01434 (2020) 2, 3, 7, 8
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 1, 2

22. He, Y., Shen, Z., Cui, P.: Towards non-iid image classification: A dataset and baselines. Pattern Recognition 110, 107383 (2021) 14

23. Iwasawa, Y., Matsuo, Y.: Test-time classifier adjustment module for model-agnostic domain generalization. In: Thirty-Fifth Conference on Neural Information Processing Systems (2021), https://openreview.net/forum?id=e_yvNqkJKA 2, 3, 4, 11, 12, 13, 20

24. Jin, X., Lan, C., Zeng, W., Chen, Z.: Feature alignment and restoration for domain generalization and adaptation (2020) 3

25. Koh, P.W., Sagawa, S., Marklund, H., Xie, S.M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R.L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B.A., Haque, I.S., Beery, S., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., Liang, P.: WILDS: A benchmark of in-the-wild distribution shifts. In: International Conference on Machine Learning (ICML) (2021) 14

26. Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., Houlsby, N.: Big transfer (bit): General visual representation learning (2020) 3, 12

27. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML. vol. 3, p. 896 (2013) 11

28. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning (2021) 4

29. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE international conference on computer vision. pp. 5542–5550 (2017) 7

30. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Learning to generalize: Meta-learning for domain generalization (2017) 3

31. Li, H., Pan, S.J., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5400–5409 (2018) 3

32. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021) 4

33. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: International Conference on Machine Learning. pp. 6028–6039. PMLR (2020) 11

34. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing (2021) 4

35. Liu, X., Ji, K., Fu, Y., Du, Z., Yang, Z., Tang, J.: P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks (2021) 4

36. Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization (2017) 3

37. Patel, V.M., Gopalan, R., Li, R., Chellappa, R.: Visual domain adaptation: A survey of recent advances. IEEE Signal Processing Magazine 32(3), 53–69 (2015). https://doi.org/10.1109/MSP.2014.2347059 1, 2

38. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021) 2, 3, 11
39. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019) 4
40. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. International journal of computer vision 77(1-3), 157–173 (2008) 7
41. Sagawa, S., Koh, P.W., Lee, T., Gao, I., Xie, S.M., Shen, K., Kumar, A., Hu, W., Yasunaga, M., Marklund, H., Beery, S., David, E., Stavness, I., Guo, W., Leskovec, J., Saenko, K., Hashimoto, T., Levine, S., Finn, C., Liang, P.: Extending the wilds benchmark for unsupervised adaptation. In: NeurIPS Workshop on Distribution Shifts (2021) 14
42. Shankar, S., Piratla, V., Chakrabarti, S., Chaudhuri, S., Jyothi, P., Sarawagi, S.: Generalizing across domains via cross-gradient training (2018) 3
43. Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., Cui, P.: Towards out-of-distribution generalization: A survey. arXiv preprint arXiv:2108.13624 (2021) 1, 2
44. Shin, T., Razeghi, Y., au2, R.L.L.I., Wallace, E., Singh, S.: Autoprompt: Eliciting knowledge from language models with automatically generated prompts (2020) 4
45. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. Communications of the ACM 59(2), 64–73 (2016) 12
46. Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., Dosovitskiy, A.: Mlp-mixer: An all-mlp architecture for vision (2021) 3, 12
47. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021) 3, 12
48. Vapnik, V., Vapnik, V.: Statistical learning theory wiley. New York 1(624), 2 (1998) 8
49. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation (2017) 7
50. Wang, D., Shellhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. arXiv preprint arXiv:2006.10726 (2020) 11
51. Wang, H., Ma, S., Dong, L., Huang, S., Zhang, D., Wei, F.: Deepnet: Scaling transformers to 1,000 layers. arXiv preprint arXiv:2203.00555 (2022) 12
52. Wang, J., Lan, C., Liu, C., Ouyang, Y., Zeng, W., Qin, T.: Generalizing to unseen domains: A survey on domain generalization. arXiv preprint arXiv:2103.03097 (2021) 1, 2, 4
53. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing 312, 135–153 (Oct 2018). https://doi.org/10.1016/j.neucom.2018.05.083, http://dx.doi.org/10.1016/j.neucom.2018.05.083 1, 2
54. Yu, Y., Jiang, H., Bahri, D., Mobahi, H., Kim, S., Rawat, A.S., Veit, A., Ma, Y.: An empirical study of pre-trained models on out-of-distribution generalization (2021) 14
55. Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C.: Domain generalization in vision: A survey (2021) 1, 2
56. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. arXiv preprint arXiv:2109.01134 (2021) 4
57. Zhou, K., Yang, Y., Hospedales, T., Xiang, T.: Deep domain-adversarial image generation for domain generalisation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 13025–13032 (2020) 3
6 Appendix

In the Appendix we added a lot of details about the data and experiments. We have also submitted all the source code used in this work in the supplement. Moreover, we will be made available to the community in a more concise and easy-to-use form after acceptance in order to facilitate the development of research in the field.

Fig. 5. The samples of PACS. From the first row to the last row are art painting, cartoon, photo and sketch domains.

Fig. 6. The samples of OfficeHome. From the first row to the last row are Art, Clipart, Product and Real World domains.

A. Detail of dataset
Fig. 7. The samples of VLCS. From the first row to the last row are Caltech101, LabelMe, SUN09 and VOC2007 domains.

Fig. 8. The samples of terra incognita. From the first row to the last row are location 100, location 38, location 43 and location 46 domains.
B. Detail of implement We have adhered to the specifications of [23] as strictly as possible. The following points are different from the previous works. 1. We adopt Transform proposed by CLIP to train the model by default. 2. We adopted the SDG optimizer that is also used in CoOp. The default of DomainBed to Adam, which is more stable than SDG, because the parameters of the optimizer are also chosen randomly in DomainBed’s specification. However, we found that it is unstable Adam training CLIP model, which is also pointed out in previous CoOp.

We have attached more details of parameters and code inside the submission.

C and D. Detail of Test-Time Adaptation experiment and Backbone Ablation. We show the detail of Test-Time Adaptation in following tables. The detail results of different backbones are reported in Appendix in the T3A paper [23].

| VLCS          | Caltech101 | LabelMe | SUN09     | VOC2007   | Avg  |
|---------------|------------|---------|-----------|-----------|------|
| CLIP + AP     | 100.0 ± 0.0| 69.5 ± 1.1| 79.9 ± 0.9| 87.6 ± 0.3| 84.3 |
| PACS          |            |         |           |           |      |
| CLIP + AP     | 98.1 ± 0.2 | 99.0 ± 0.1| 99.9 ± 0.1| 92.3 ± 0.4| 97.3 |
| OfficeHome    |            |         |           |           |      |
| CLIP + AP     | 82.5 ± 0.7 | 71.7 ± 0.4| 91.2 ± 0.6| 91.5 ± 0.3| 84.2 |
| Terra         |            |         |           |           |      |
| CLIP + AP     | 58.2 ± 2.1 | 57.2 ± 1.0| 50.1 ± 0.2| 44.9 ± 2.6| 52.6 |

Table 6. Detailed results on VLCS, PACS, OfficeHome, TerraIncognita.