Understanding and Improving Visual Prompting: A Label-Mapping Perspective

Aochuan Chen¹, Yuguang Yao¹, Pin-Yu Chen², Yihua Zhang¹, Sijia Liu¹,²
¹Michigan State University, ²MIT-IBM Watson AI Lab, IBM Research

Abstract

We revisit and advance visual prompting (VP), an input prompting technique for vision tasks. VP can reprogram a fixed, pre-trained source model to accomplish downstream tasks in the target domain by simply incorporating universal prompts (in terms of input perturbation patterns) into downstream data points. Yet, it remains elusive why VP stays effective even given a ruleless label mapping (LM) between the source classes and the target classes. Inspired by the above, we ask: How is LM interrelated with VP? And how to exploit such a relationship to improve its accuracy on target tasks? We peer into the influence of LM on VP and provide an affirmative answer that a better ‘quality’ of LM (assessed by mapping precision and explanation) can consistently improve the effectiveness of VP. This is in contrast to the prior art where the factor of LM was missing. To optimize LM, we propose a new VP framework, termed ILM-VP (iterative label mapping-based visual prompting), which automatically re-maps the source labels to the target labels and progressively improves the target task accuracy of VP. Further, when using a contrastive language–image pretrained (CLIP) model for VP, we propose to integrate an LM process to assist the text prompt selection of CLIP and to improve the target task accuracy. Extensive experiments demonstrate that our proposal significantly outperforms state-of-the-art VP methods. As highlighted below, we show that when reprogramming an ImageNet-pretrained ResNet-18 to 13 target tasks, ILM-VP outperforms baseline methods by a substantial margin, e.g., 7.9% and 6.7% accuracy improvements in transfer learning to the target Flowers102 and CIFAR100 datasets. Besides, our proposal on CLIP-based VP provides 13.7% and 7.1% accuracy improvements on Flowers102 and DTD respectively. Code is available at https://github.com/OPTML-Group/ILM-VP.

1. Introduction

When learning new knowledge, humans typically start to compare and connect it with the knowledge that they were familiar with. The same idea is also applied in ML. For example, in the ‘pretraining + finetuning’ paradigm, an ML model (e.g., deep neural network or DNN) is first trained on a (usually large) source dataset. When a relevant downstream task is present, the pre-trained model is then finetuned over the target dataset. This learning paradigm has been predominant in the classical transfer learning [3–8] as well as in the recent deep representation learning [9–13]. However, finetuning the pre-trained model requires either partial or entire model modifications. If the pre-trained model is of large size, then it becomes too costly to store a modified copy of the pre-trained model for each downstream task. In contrast, visual prompting (VP) (see Fig. 1), also known as model reprogramming or adversarial reprogramming, provides a new alternative to finetuning [1, 2, 14–17]. Instead of directly modifying the pre-trained source model, VP integrates an input transforma-
tion and/or an output transformation to reprogram the fixed source model to accomplish a new target task; see an illustration of existing VP framework in Fig. 1. The input transformation is typically realized by incorporating (data-agnostic) input perturbations (i.e., prompts) into input samples, and the output transformation is given by a function that maps source labels to target labels, known as label mapping (LM). Recently, VP has shown great promise in various applications of foundation models, ranging from pretrained vision models [1, 14, 15, 17–20] to language-vision models [2, 21–23].

The idea of prompt learning originated from in-context learning or prompting in natural language processing (NLP) [24–26]. However, when it is introduced to the vision domain [1, 2], new questions arise. First, the recent work [1, 14, 27] showed that VP remains powerful even if the target task largely deviates from the source domain. For example, a new performance record on target medical datasets is achieved in [1] when using VP to reprogram the fixed, ImageNet pre-trained source model. The ‘mystery’ in this example is that LM is conducted between two seemingly irrelevant source and target domains. Despite the lack of interpretability, VP can still leverage such connected source labels and the source model to effectively predict target data points. This raises the first open question: What is the rationality behind LM and how to explore its influence on VP? Second, unlike prompt learning in the NLP domain, input prompts in the vision domain are typically given by ‘noisy’ perturbations to image pixels; see illustration in Fig. 1. Together with the lack of interpretability of LM, the second open question is: How to interpret LM and the seemingly random perturbation pattern in VP?

As mentioned above, the lack of understanding of LM and the poor interpretability of VP drive our studies in this work. We develop a new visual prompting framework, termed ILM-VP (iterative label mapping-based visual prompting), which provides an interactive and explainable design between LM and prompt learning (i.e., input prompt generation); see Fig. 1 for the schematic overview. Our proposal can automatically adjust LM between the source domain and the target domain by taking both mapping precision and explanation into consideration, and can leverage the optimized LM to further improve the accuracy and the explainability of prompt learning. Although some prior work [1, 17, 27] attempted to improve the quality of LM as well as the overall performance of VP, they are different from our proposal in two major aspects. First, none of the prior work co-designed LM and VP. For example, the prior art [1] used a pre-prompt prediction frequency to determine the LM function. However, we find significant inconsistency between the pre-prompt and post-prompt prediction frequency of the same source model, which explains the sub-optimality of the current VP methods due to the lack of mapping precision. Second, to the best of our knowledge, VP is still treated as a ‘black box’ in the prior work. Yet, our design can provide graceful visual explanations to the underlying mechanisms of VP. Third, we for the first time show that LM can provide a unified solution to improving the accuracy of VP to re-purpose both vision and language-vision source models. Our contributions are unfolded below.

1. We revisit the LM problem in VP and uncover the deficiencies of existing LM methods: the lack of mapping precision and the lack of explanation.
2. Given the importance of LM, we propose the first LM-VP co-design framework, termed ILM-VP, through a novel bi-level optimization viewpoint.
3. Beyond LM for vision models, we show that LM can also be generalized to assist the text prompt selection of CLIP (contrastive language–image pretraining) and to improve the target task accuracy of VP using the CLIP model.
4. We empirically demonstrate the accuracy and explanation merits of our proposal across multiple source models and target datasets.

2. Related Work

Prompting in NLP. Prompting is used to prepend language instruction to the input text for a language model to better accomplish a given task [28]. While prompting makes a significant contribution to the generalization ability of large pre-trained language models (e.g., GPT-3) [24], it requires hand-crafting prompt design by experts. Recent work proposed to directly optimize the prompting embeddings through gradients together with lightweight finetuning the model, which is called prompt tuning [25, 29]. It is shown that this method is effective and efficient, which achieves competitive performance to the finetuning of the full language model.

Visual prompting and model reprogramming. VP was first defined in [2] to mimic the prompting idea in NLP. Prior to that, a very similar idea was used in computer vision (CV) but with a different name, known as model reprogramming or adversarial reprogramming [14–17, 30–33]. They both focus on re-purposing a fixed, pre-trained vision model for a new task by leveraging a universal input pattern and an output LM function. Although not outperforming full fine-tuning in transfer learning, VP yields an advantage of parameter-efficient fine-tuning, which requires a much smaller parameter storage space. Furthermore, the smaller parameter space requires less training data to converge. Beyond traditional pre-trained vision models, the work [2] studied the effectiveness of VP in the language-vision model CLIP for the first time. Assisted by CLIP, VP can generate a prompting pattern of image data without resorting to source-target label mapping. In [23], VP and text prompt are jointly optimized in the CLIP model, which leads to better performance. Furthermore, unadver-
sarial learning [34] also enjoys the similar idea to VP, while it focuses on generating class-wise prompts with the goal of improving the out-of-distribution generalization ability of a pre-trained model.

VP is gaining increasing attention. In [1], it is applied to re-purpose black-box source models [35] and achieves state-of-the-art (SOTA) performance on different target datasets. Besides, in data-scarce regimes like the biochemical domain, it is shown in [15, 17, 27] that VP can enable effective cross-domain transfer learning. Other than transfer learning, VP is also used in in-domain settings to improve different metrics like adversarial robustness [33] and fairness [32]. Although input prompting is the most commonly-used prompt learning method in the vision domain, generalization to learning prompting parameters at intermediate layers of a source model is also developed in [19–21, 36]. The resulting technique is called visual prompt tuning and is typically restricted to vision transformers.

3. Problem Statement

In this section, we begin by providing some background information on VP. Based on that, we will then present the problem of our interest—LM (label mapping)—which defines how a visual prompt maps a source model prediction label to a target data class. This is the first question encountered in VP across domains but was typically overlooked in the literature. By reviewing the commonly-used LM methods, we will point out several open questions raised by LM.

Preliminaries on visual prompting. The technology of VP addresses the problem of how to adapt a pre-trained source model (e.g., the ImageNet-1K-pre-trained ResNet-18) to a target downstream task (e.g., flower classification over the Flowers102 dataset) without any task-specific model modification (e.g., finetuning). Throughout the paper, we focus on input-based VP (also known as model reprogramming) [1, 2, 14–17, 27, 37], which incorporates a carefully-designed universal perturbation pattern to the raw target images so as to enforce the transferability of the source model to the target domain. We refer readers to Fig. 1 for the schematic overview.

To be concrete, let $S$ and $T$ denote the source dataset and the target dataset, respectively. And let $f_{\theta_{s}}$ denote a source model with pre-trained parameters $\theta_{s}$. Suppose $f_{\theta_{s}}$ is a supervised classifier, then it defines a mapping from the input data $x \in \mathbb{R}^{N_{s}}$ to the source label space $\mathcal{Y}_{s} \subseteq \mathbb{R}^{K_{s}}$, i.e., $f_{\theta_{s}}(x) = y_{s} \in \mathcal{Y}_{s}$, where $N_{s}$ is the dimension of a source datapoint, $K_{s}$ is the number of source data classes, and $y_{s}$ is the source class label. We have $f_{\theta_{s}}$ trained based on $S$, e.g., via empirical risk minimization. The goal of VP is to reprogram the source model $f_{\theta_{s}}$ to accomplish the target task defined in $T$, without making task-specific finetuning over $f_{\theta_{s}}$. To this end, VP modifies the target data $x_{t}$ (of $N_{t}$ dimensions) by injecting a task-designated input perturbation pattern $\delta$. This leads to the input prompting operation with the generic form:

$$x'(\delta) = h(x_{t}, \delta) \in \mathbb{R}^{N_{t}}, x_{t} \in \mathbb{R}^{N_{t}}$$

where $x_{t}$ is the target datapoint, and $h(\cdot, \cdot)$ is an input transformation that integrates $x_{t}$ with the input perturbation $\delta$ and produces a modified datapoint $x'(\delta)$ with the source data dimension $N_{s}$. It was shown in [1] that $h$ can be specified as an additive perturbation model that pads $\delta$ outside the target data sample (see Fig. 1 for an example as well).

Given the input prompting model (1), VP then seeks the optimal $\delta$ to improve the target task accuracy when using the pre-trained source model $f_{\theta_{s}}$. This raises a prompt generation problem, which is typically cast as

$$\min_{\delta} \mathbb{E}_{(x_{t}, y_{t}) \in T_{tt}}[\ell_{VP}(f_{\theta_{s}}(x'(\delta)), y_{t})],$$

where $T_{tt}$ denotes a supervised training set in $T$ with feature $x_{t}$ and label $y_{t}$ for a training sample, and $\ell_{VP}(\cdot)$ is a visual prompting loss function that we will define later given the prompted input $x'(\delta)$ and the ground-truth target label $y_{t}$. To solve problem (2), the standard stochastic gradient descent (SGD) method can be used. At inference, we will integrate the designed $\delta$ into test-time target datapoints and call the source model $f_{\theta_{s}}$ for downstream prediction in $T$ (see Fig. 1 and a more detailed description in Fig. A1).

Label mapping: Existing methods and questions. Although the input prompting operation (1) converts the original $x_{t}$ to the source dimension-aligned datapoint $x'$ that the source model can use, the successful realization of VP (3) needs to map the source model’s prediction (in the source label space $\mathcal{Y}_{s}$ with $K_{s}$ classes) to the target task’s data label (in the target label space $\mathcal{Y}_{t}$ with $K_{t}$ classes). In the ‘pre-training + finetuning’ paradigm, we typically have $K_{t} \leq K_{s}$. Therefore, the problem of LM (label mapping) arises:

(LM problem) Given the source model $f_{\theta_{s}}$, how to build a mapping from the source label space $\mathcal{Y}_{s}$ to the target label space $\mathcal{Y}_{t}$ so that the model’s prediction directs to the correct target label?

Clearly, the desired prompt generation (2) heavily relies on the LM scheme, which defines the one-to-one correspondence between the source model’s prediction $f_{\theta_{s}}(x'(\delta))$ and the target data class $y_{t}$. Yet, nearly all the existing work neglects its influence on the prompt generation and adopts either (1) the simplest random mapping [2,14] or (2) a pre-defined, one-shot frequency-based mapping [1,15]. We elaborate on the above two schemes below.

(1) Random label mapping (RLM): RLM does not use any prior knowledge or source model information to guide the LM process. The mapped source labels (to the target domain) could be even random. For example, in the case of ‘ImageNet (source) + CIFAR-10 (target)’ [2,14], existing VP methods coded CIFAR-10 labels using the top 10 ImageNet indices, i.e., ImageNet label $i \rightarrow$ CIFAR-10 label $i$, despite the lack of interpretation.
The ‘missing’ dynamics of LM in the source domain. As shown in Sec. 3, a prompt learning pipeline mainly involves three steps: (A1) input prompt modeling (1), (A2) LM (from the source label set \( Y_s \) to the target label set \( Y_t \)), and (A3) prompt generation (2). The prior art follows the pipeline (A1) \( \rightarrow \) (A2) \( \rightarrow \) (A3) to generate the desired prompt \( \delta^* \), which drives the source model to accomplish target tasks. However, in the viewpoint of the source domain, the prompt updating from \( \delta = 0 \) to \( \delta^* \) induces the prediction dynamics of the source model \( f_{\theta_s} \). That is,

\[
f_{\theta_s}(x'(0)) \rightarrow f_{\theta_s}(x' (\delta^*)),
\]

where \( x'(\delta) \) has been defined in (1), which refers to the \( \delta \)-perturbed target data with the same dimension as the source datapoint. As will be evident later, it is important to understand the dynamics (4) as it reflects the stability of the selected source labels when mapping to the target labels.

Fig. 2 instantiates the dynamics of (4) in the scenario of ‘ImageNet (source) + Flowers102 (target)’ when the FLM-oriented VP approach (3) is used [1]. Prior to prompt generation, the Flowers102 target labels are first mapped to the ImageNet source labels using the FLM method (3), corresponding to step (A2) in prompt learning. This yields the pre-prompt target-source mapping, denoted by \( f_{\theta_s}(x'(0)) \rightarrow Y_t \). Similarly, after generating the prompt \( \delta^* \) following (A3), we can obtain the post-prompt target-source mapping, \( f_{\theta_s}(x'(\delta^*)) \rightarrow Y_t \), using the FLM method. Fig. 2 shows that there exists a significant discrepancy between the pre-prompt LM and the post-prompt LM, evidenced by the newly-selected source labels (‘Cardoon’, ‘Egret’, ‘Daisy’, ‘Paper Towel’) in the post-prompting phase. This justifies the dynamics of (4) in LM. However, it also raises a new concern that the pre-prompt target-source LM is sub-optimal for prompt generation (2) given the existing dynamics of LM in the source domain.

**IIM-VP: A bi-level optimization viewpoint of VP.** The dynamics of LM inspire us to re-think the optimality of the current VP pipeline: (A1) \( \rightarrow \) (A2) \( \rightarrow \) (A3). To improve it, we propose to take the LM dynamics into the prompt learning process. This modifies the conventional VP pipeline to (A1) \( \rightarrow \) (A2) \( \Rightarrow \) (A3), where LM and prompt generation are in a closed loop. Since the design of LM will interact with the design of the prompt iteratively, we call the proposed new design ILM-VP.

Next, we formally present ILM-VP through the lens of bi-level optimization (BLO). Generally speaking, BLO provides a hierarchical learning framework involving two levels (i.e., upper and lower levels) of optimization tasks, where one task is nested inside the other (i.e., the objective and variables of an upper-level problem depend on the optimizer of the lower-level problem). In the context of ILM-VP, we regard the prompt generation problem (2) as the upper-level optimization task and the LM problem (3)
as the lower-level problem. This yields

$$\begin{align*}
\text{minimize} & \quad E_{(x, y)}(\mathcal{T}_t) \left[ \ell(f_{\theta}(x'(\delta)), y^s_t(y_t)) \right] \\
\text{subject to} & \quad y^s_t(y_t) \text{ is obtained by (3) at (non-zero) prompt } \delta \\
\end{align*}$$

(5)

where the visual prompt $\delta$ denotes the upper-level variable, $\ell$ is the cross-entropy loss, and the mapped source label $y^s_t$ is a lower-level variable for each given target label $y_t$ at the current prompt $\delta$. We also note that there exists a lower-level constraint in (5) to ensure that if a source class has been mapped to a target class, it will then be excluded when mapping to a new target class. Further, it is clear from (5) that the design of visual prompt $\delta$ and LM $y^s_t$ (vs. $y_t$) are intertwined with each other.

To solve problem (5), we employ the alternating optimization (AO) method, which alternatively executes the upper-level prompt generation and the lower-level LM. We summarize the algorithm details in Algorithm 1 and provide a schematic overview in Fig. A2.

Algorithm 1 The proposed ILM-VP algorithm

1: Initialize: Given target training set $\mathcal{T}_t$, pre-trained model $f_{\theta}$, prompt pattern initialization $\delta_0$, and upper-level learning rate $\lambda$ for SGD
2: for Epoch $n = 0, 1, \ldots$, do
3: \hspace{1em} Lower-level mapping: Given $\delta_{n-1}$, call LM for each target class $y_t$ in $\mathcal{T}_t$
4: \hspace{1em} Upper-level prompt learning: Given LM, call SGD to update prompt $\delta_n \leftarrow \delta_{n-1}$
5: end for

An interpretation merit of ILM-VP. In the literature, it is quite difficult to interpret why VP can reprogram a source model to conduct target tasks. The main hurdle of interpreting VP lies in the LM phase: It remains elusive why the semantics-irrelevant source labels should be mapped to target labels. However, we find that ILM-VP can alleviate this interpretation difficulty to a large extent. We show the explanation merit of ILM-VP through an empirical study in Fig. 3, where the target dataset is instantiated by Flowers102 and the source dataset is ImageNet-1K. We list the target labels, the mapped source labels using the baseline FLM method [1], and the identified source labels using ILM-VP, together with image examples under each label. As we can see, an interpretable target-source mapping is found by ILM-VP, even if the target label and the source label describe different subjects. For example, target images in the label ‘Spear Thistle’ share a similar color and object shape with the source images in the label ‘Cardoon’. The same observations can also be drawn from other target-source label mappings together with their data instances. This finding is quite encouraging and is in sharp contrast to FLM. As will be evident in Sec. 5, the BLO-oriented ILM-VP (5) would enforce a convergence of LM as the alternating optimization proceeds. As a result, source labels and target labels, which share the most similar concepts (like colors, scenes, shapes, and materials), will be identified. We also show that the improved interpretation of LM consistently enhances the target task accuracy of the VP.

5. Experiments

In this section, we empirically demonstrate the effectiveness of our proposed ILM-VP method by comparing it with a variety of baselines across multiple datasets, models, and learning paradigms.

5.1. Experiment setups

Datasets and models. In the source domain, we will consider the source models ResNet-18 and ResNet-50 [38] pre-trained on ImageNet-1K [39], and the source model ResNeXt-101-32x8d [40] pre-trained on Instagram [41]. In the target domain, we will evaluate the performance of ILM-VP over 13 target datasets: Flowers102 [42], DTD [43], UCF101 [44], Food101 [45], GTSRB [46], SVHN [47], EuroSAT [48], OxfordPets [49], StanfordCars [50], SUN397 [51], CIFAR10/100 [52], ABIDE [53].
5.2. Experiment results

Overall performance of ILM-VP. Tab. 1 shows the effectiveness of our proposed ILM-VP method vs. VP baselines (RLM-VP and FLM-VP) on diverse source models and target datasets. For comparison, we also present the model finetuning performance on target datasets using LP or FF. It is worth noting that FLM-VP typically outperforms RLM-VP as the latter only uses a random label mapping to guide the learning of prompts [1]. Thus, we only show the results of RLM-VP when using ResNet-18.

As shown in Tab. 1, our proposed method (ILM-VP) consistently outperforms other VP baselines by a large margin in nearly all the data-model setups, e.g., 7.9%, 6.7% and 6.5% accuracy improvement over FLM-VP in the target dataset Flowers102, CIFAR100, GTSRB, respectively. In addition, we note that model finetuning is typically more effective in transfer learning than prompting methods, consistent with existing work [2]. This is not surprising as source models are allowed for modification, and the trainable parameter size increases (as evidenced by ‘Parameter Size’ in Tab. 1). As will be evident in Sec. 6, the accuracy of VP can be further improved if a language-translation source model is used. Nonetheless, in the target dataset ABIDE, prompting methods can outperform the full model finetuning method (FF). Compared to other standard transfer learning tasks for image classification, ABIDE was a newly-proposed medical dataset in [1], which converts the original 1D numerical medical input sequences to image-alike data formats (i.e., brain regional correlation graphs). The size of this dataset is extremely small due to the high cost of collecting data in the medical area, which restricts the performance of LP and FF. In contrast, VP is uniquely suited for this setting. Lastly, in the model finetuning paradigm, a source model with larger capacity typically yields a better target task accuracy, e.g., the finetuning results of ResNet-50 vs. ResNet-18. However, this belief might not hold in the VP paradigm. As we can see, the prompting-induced target accuracy de-
increases under ResNet-50 in the target datasets Flowers102, EuroSAT, CIFAR100, and ABIDE.

Additionally, **Tab. A2** in Appendix shows that ILM-VP takes a bit more run time than FLM-VP and LP, but is faster than FF. This is not surprising since the former adopts alternating optimization with a bit higher computation complexity than ordinary single-level minimization. Recently, the concurrent work [55] shows that properly re-sizing images before integrating with a VP could further boost the performance on a downstream task. We also find the same benefit of image re-sizing to VP on CIFAR10/100, GTSRB, and SVHN datasets (e.g., up-scaling the original image size to 128 × 128). However, for ease of comparison with existing VP baselines (RLM-VP [14]), our experiments do not apply the image re-sizing trick to VP.

**LM is key to improving the accuracy of VP.** Next, we peer into the influence of LM on target prediction accuracy per class when using ILM-VP. In **Fig. 4**, we demonstrate the testing accuracy improvements (over the FLM-VP baseline) of prompt-injected datapoints, belonging to 10 classes with the highest improvements selected from the target datasets Flowers102 and OxfordPets, respectively. Note that OxfordPets shares the most similar label space with ImageNet (e.g., they both have beagles, boxers, bassets, etc.). We use * in **Fig. 4** to mark target data classes whose source labels are remapped during ILM-VP, and list non-* marked target data classes whose source labels retain the same as FLM-VP. We observe that target classes with large accuracy improvements typically require ILM. This justifies the benefit of target-source label re-mapping during prompt learning. In addition, we note that the source labels of target classes (e.g., ‘yorkshire’ in OxfordPets) are not re-mapped, but ILM-VP can still bring in accuracy improvements.

This implies that LM has a coupling effect on all classes and the BLO framework (5) enables us to improve LM as well as prompt learning in an interactive manner.

Further, **Fig. 5** shows the training dynamics of ILM-VP vs. training epoch number and its convergence to the stable, high-explainable, and high-accurate visual prompt. As we can see, the mapped source label for a target class is updated at the early training epochs of ILM-VP, but tends to converge at the later training phase. A similar trend holds for the convergence of LM difference between two adjacent epochs and the VP training loss. In addition, we can see that the VP pattern and the LM are updated jointly. Furthermore, the explainability of mapped source labels grows as the training proceeds. For example, the target label ‘Marigold’ shares a similarity with the source label ‘Orange’ in color and shape, as visualized by EBE-identified examples. It is worth mentioning that EBE facilitates us to directly link the source dataset and the target dataset, and thus helps us to better understand the rationale behind VP. We refer readers to Appendix C for more EBE results.

**How target dataset scale affects VP?** Through our experiments over a large number of target datasets, we find that ILM-VP becomes more powerful when it comes to tasks with a larger target label space. For example, **Fig. 6** shows the target datasets with at least 3% accuracy improvement using ILM-VP compared with FLM-VP on ResNet-18. As we can see, target datasets with the highest number of target classes correspond to the most significant accuracy improvement brought by ILM-VP. Next, we fix the target dataset and study how VP behaves at different downstream training dataset sizes. Here we choose GTSRB as the target task since GTSRB contains a sufficient amount of training data and thus facilitates us to conduct training dataset par-
LM improves VP’s accuracy using CLIP. In Tab. 2, we demonstrate the performance of the VP-driven CLIP model on several challenging target tasks shown in Tab. 1, e.g., Flowers102 and DTD. We term our method ‘VP+TP+LM’, where the BLO-enabled LM method is called to map ‘virtual source labels’ (i.e., the combination of context prompt template and target label) to realistic target image labels. For comparison, we also present the performance of the baseline method termed ‘VP + TP’ [2], which uses a pre-defined, fixed context prompt template ‘This is a photo of a {label}’ when generating a visual prompt for CLIP. As we can see, our proposal consistently outperforms the baseline by a substantial margin. For example, we obtain 13.7% and 7.1% accuracy gain in Flowers102 and DTD respectively.

In addition, we find that LM brings in the interpretability merit: Our selected context prompt templates have better semantic meaning than the one used by the baseline. For example, VP for Flowers102 selects the text prompt ‘a close-up photo of a {label}’ instead of ‘This is a photo of a {label}’ for the target image with the label ‘buttercup’. Another example is that VP for CIFAR10 prefers the text prompt ‘a pixelated photo of a {label}’. In particular, we observe that in domain shift datasets (ImageNet-R and ImageNet-Sketch), the selected prompts can exhibit the domain information. More explainable results can be found in Fig. A6.

<table>
<thead>
<tr>
<th>Methods</th>
<th>VP+TP</th>
<th>VP+TP+LM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flowers102</td>
<td>70.0</td>
<td>83.7</td>
</tr>
<tr>
<td>DTD</td>
<td>56.8</td>
<td>63.9</td>
</tr>
<tr>
<td>UCF101</td>
<td>66.0</td>
<td>70.6</td>
</tr>
<tr>
<td>Food101</td>
<td>78.9</td>
<td>79.1</td>
</tr>
<tr>
<td>SVHN</td>
<td>89.9</td>
<td>91.2</td>
</tr>
<tr>
<td>EuroSAT</td>
<td>96.4</td>
<td>96.9</td>
</tr>
<tr>
<td>StanfordCars</td>
<td>57.2</td>
<td>57.6</td>
</tr>
<tr>
<td>SUN397</td>
<td>60.5</td>
<td>61.2</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>93.9</td>
<td>94.4</td>
</tr>
<tr>
<td>ImageNet-R</td>
<td>67.5</td>
<td>68.6</td>
</tr>
<tr>
<td>ImageNet-Sketch</td>
<td>38.5</td>
<td>39.7</td>
</tr>
</tbody>
</table>

Tab. 2. Results of our CLIP-based prompt learning ‘VP+TP+LM’ and the baseline ‘VP+TP’ [2] (restricted to using text prompt template “This is a photo of a {label}”) over 11 target datasets. In each cell, the target task accuracy (%) is shown along with examples of LM in the text domain. Our method with higher accuracy than SOTA is marked in bold.

7. Conclusion

This paper unveils LM’s importance in the VP framework. Inspired by the prediction dynamics in optimizing VP, we formalize the VP problem through the lens of BLO (bi-level optimization). Upon our formalization, we propose a novel ILM-VP algorithm to jointly optimize the input pattern training and the LM function. Across 13 datasets, we show our method’s significant accuracy improvement over the SOTA VP baselines with graceful interpretability. Further, we extend our method to CLIP to improve its downstream task performance.

8. Acknowledgement

The work of A. Chen, Y. Yao, Y. Zhang, and S. Liu was supported by the DARPA RED program and National Science Foundation (NSF) Grant IIS-2207052.
References


Proceedings of the IEEE conference on computer vision and pattern recognition 2017.

5


5


5


5


5


5


5


5


5

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2013.

5


5


5

