This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Understanding and Improving Visual Prompting: A Label-Mapping Perspective

Aochuan Chen[1], Yuguang Yao[1], Pin-Yu Chen[2], Yihua Zhang[1], Sijia Liu[1,2]
[1]Michigan State University, [2]MIT-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 baselines 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 fine-tuned 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].

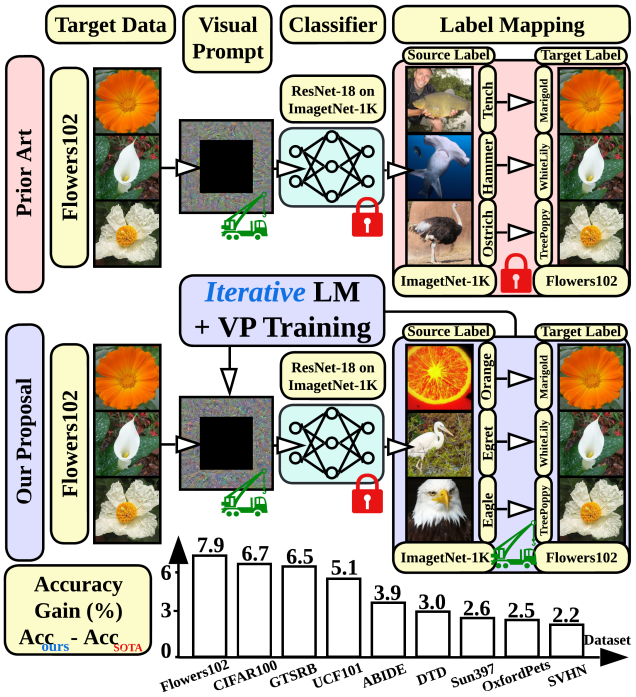


Fig. 1. Overview of VP pipelines (prior art [1,2] and our proposal termed ILM-VP) and accuracy improvement achieved by ILM-VP on target image classification tasks at-a-glance. Generally speaking, VP aims to generate a universal input perturbation template (*i.e.*, 'visual prompt') and leverage a source-target LM (label mapping) in order to drive the fixed source model (*e.g.*, pretrained on ImageNet-1K) to conduct a target task (*e.g.*, Flowers102 image classification). Compared to the prior art, our proposal (ILM-VP) couples the design of LM with VP training. The resulting LM-VP co-design improves target task accuracy across a variety of target image classification tasks using a fixed ImageNet-pretrained source model.

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

$$\mathbf{x}'(\boldsymbol{\delta}) = h(\mathbf{x}_\mathrm{t}, \boldsymbol{\delta}) \in \mathbb{R}^{N_\mathrm{s}}, \ \mathbf{x}_\mathrm{t} \in \mathbb{R}^{N_\mathrm{t}} \tag{1}$$

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

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

$$\underset{\boldsymbol{\delta}}{\text{minimize}} \quad \mathbb{E}_{(\mathbf{x}_\mathrm{t}, y_\mathrm{t}) \in \mathcal{T}_\mathrm{tr}} \left[\ell_{\mathrm{VP}}(f_{\boldsymbol{\theta}_\mathrm{s}}(\mathbf{x}'(\boldsymbol{\delta})), y_\mathrm{t}) \right], \tag{2}$$

where $\mathcal{T}_\mathrm{tr}$ denotes a supervised training set in $\mathcal{T}$ with feature $\mathbf{x}_\mathrm{t}$ and label y_t for a training sample, and $\ell_{\mathrm{VP}}(\cdot)$ is a visual prompting loss function that we will define later given the prompted input $\mathbf{x}'(\boldsymbol{\delta})$ 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 $\boldsymbol{\delta}$ into test-time target datapoints and call the source model $f_{\boldsymbol{\theta}_\mathrm{s}}$ for downstream prediction in $\mathcal{T}$ (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 $\mathbf{x}_\mathrm{t}$ to the source dimension-aligned datapoint $\mathbf{x}'$ 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 $\mathcal{Y}_\mathrm{s}$ with K_s classes) to the target task's data label (in the target label space $\mathcal{Y}_\mathrm{t}$ with K_t classes). In the 'pre-training + finetuning' paradigm, we typically have $K_\mathrm{t} \leq K_\mathrm{s}$. Therefore, the problem of LM (label mapping) arises:

> (**LM problem**) Given the source model $f_{\boldsymbol{\theta}_\mathrm{s}}$, how to build a mapping from the source label space $\mathcal{Y}_\mathrm{s}$ to the target label space $\mathcal{Y}_\mathrm{t}$ 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 $f_{\boldsymbol{\theta}_\mathrm{s}}(\mathbf{x}'(\boldsymbol{\delta}))$ 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.

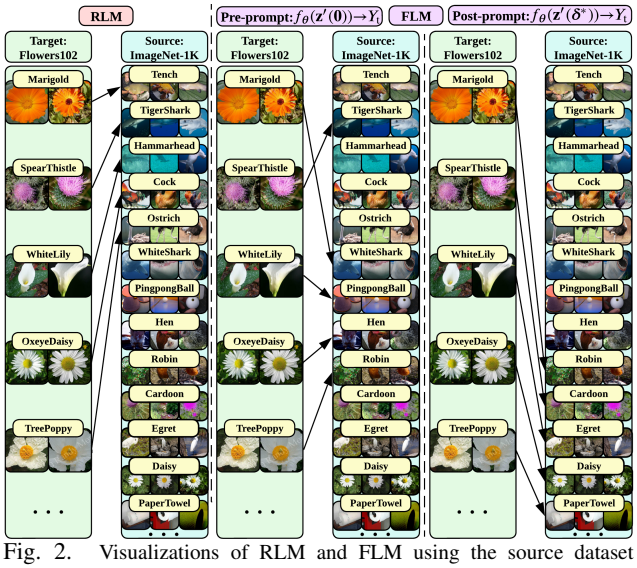


Fig. 2. Visualizations of RLM and FLM using the source dataset ImageNet-1K and the pretrained ResNet-18, as well as the target dataset Flowers102. In FLM, the pre-prompt label mapping using (3) selects source labels different from FLM. Yet, the post-prompt label mapping using (3) at $\boldsymbol{\delta}^*$ in (4) shows many newly-selected source labels, indicating (1) the dynamics of LM in the source domain, and (2) the pre-promp LM is sub-optimal (*i.e.*, mis-selecting the best-matching label) after VP training.

② **Frequency-based label mapping (FLM)**: FLM matches target labels to source labels based on the source model's prediction frequencies on zero-padded target datapoints, *i.e.*, $f_{\boldsymbol{\theta}_s}(\mathbf{x}'(\boldsymbol{\delta}))$ with $\boldsymbol{\delta} = \mathbf{0}$. Here recall from (1) that $\mathbf{x}'(\mathbf{0}) = h(\mathbf{x}_t, \mathbf{0})$. More concretely, a target label y_t is mapped to the source label y_s^* following

$$y_s^*(y_t) = \arg\max_{y_s} \Pr\{\text{Top-1 prediction of } f_{\boldsymbol{\theta}_s}(h(\mathbf{x}_t, \mathbf{0})) \text{ is } y_s \,|\, \forall \mathbf{x}_t \in \mathcal{T}_{y_t}\} \quad (3)$$

where $y_s^*(y_t)$ explicitly expresses the dependence of the mapped source label on the target label, $\mathcal{T}_{y_t}$ denotes the target data set in the class y_t, and $\Pr\{\cdot\}$ is the probability of the event that the top-1 prediction of $f_{\boldsymbol{\theta}_s}$ is the source class y_s under the zero-padded target data points in $\mathcal{T}_{y_t}$.

As shown in **Fig. 2**, FLM results in a mapping scheme different from that of RLM. However, it is still difficult to interpret the obtained LM results, and remains elusive how the quality of LM impacts the performance of VP. In the rest of the work, we will shed light on how to improve VP by carefully designing the LM scheme and when and why LM matters at different source and downstream tasks.

4. Method: Iterative Label Mapping-based VP

In this section, we uncover the *hidden dynamics of LM* existing in the source task domain of VP, which was neglected by the prior art. This finding then motivates us to develop a novel VP framework, which we call Iterative LM-based VP (**ILM-VP**). Compared to existing VP methods, ILM-VP closes the loop between LM and prompt generation (2), and improves VP's explanation and target task accuracy simultaneously.

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 $\mathcal{Y}_s$ to the target label set $\mathcal{Y}_t$), and **(A3)** prompt generation (2). The prior art follows the pipeline (A1)→(A2)→(A3) to generate the desired prompt $\boldsymbol{\delta}^*$, which drives the source model to accomplish target tasks. However, in the viewpoint of the *source* domain, the prompt updating from $\boldsymbol{\delta} = \mathbf{0}$ to $\boldsymbol{\delta}^*$ induces the prediction dynamics of the source model $f_{\boldsymbol{\theta}_s}$. That is,

$$f_{\boldsymbol{\theta}_s}(\mathbf{x}'(\mathbf{0})) \to f_{\boldsymbol{\theta}_s}(\mathbf{x}'(\boldsymbol{\delta}^*)), \quad (4)$$

where $\mathbf{x}'(\boldsymbol{\delta})$ has been defined in (1), which refers to the $\boldsymbol{\delta}$-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 $f_{\boldsymbol{\theta}_s}(\mathbf{x}'(\mathbf{0})) \overset{\text{FLM}}{\to} \mathcal{Y}_t$. Similarly, after generating the prompt $\boldsymbol{\delta}^*$ following (A3), we can obtain the *post-prompt* target-source mapping, $f_{\boldsymbol{\theta}_s}(\mathbf{x}'(\boldsymbol{\delta}^*)) \overset{\text{FLM}}{\to} \mathcal{Y}_t$, 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.

ILM-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)→(A2)→(A3). To improve it, **we propose** to take the LM dynamics into the prompt learning process. This modifies the conventional VP pipeline to (A1)→(A2)⇄(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

$$\underbrace{\underset{\boldsymbol{\delta}}{\text{minimize}} \ \mathbb{E}_{(\mathbf{x}_\text{t},y_\text{t})\in\mathcal{T}_\text{tr}}\left[\ell(f_{\boldsymbol{\theta}_\text{s}}(\mathbf{x}'(\boldsymbol{\delta})),y_\text{s}^*(y_\text{t}))\right]}_{\text{Upper-level prompt optimization}}$$
$$\text{subject to } \underbrace{y_\text{s}^*(y_\text{t}) \text{ is obtained by (3) at (non-zero) prompt } \boldsymbol{\delta}}_{\text{Lower-level LM design at current prompt } \boldsymbol{\delta} \text{ for every target label } y_\text{t}} \tag{5}$$

where the visual prompt $\boldsymbol{\delta}$ denotes the upper-level variable, ℓ is the cross-entropy loss, and the mapped source label y_s is a lower-level variable for each given target label y_t at the current prompt $\boldsymbol{\delta}$. 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 $\boldsymbol{\delta}$ and LM y_s^* (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 $\mathcal{T}_\text{tr}$, pre-trained model $f_{\boldsymbol{\theta}_\text{s}}$, prompt pattern initialization $\boldsymbol{\delta}_0$, and upper-level learning rate λ for SGD
2: **for** Epoch $n = 0, 1, \ldots,$ **do**
3: **Lower-level label mapping**: Given $\boldsymbol{\delta}_{n-1}$, call LM for each target class y_t in $\mathcal{T}_\text{tr}$
4: **Upper-level prompt learning**: Given LM, call SGD to update prompt $\boldsymbol{\delta}_n \leftarrow \boldsymbol{\delta}_{n-1}$
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

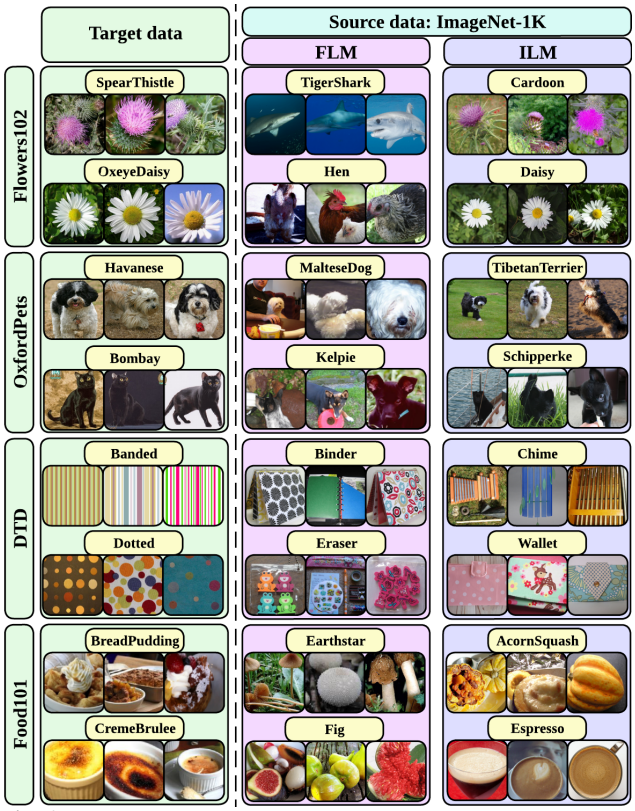


Fig. 3. Interpretation merit of ILM (ours) vs. FLM, visualized by LM results in VP to re-purpose an ImageNet-pretrained source model (ResNet-18) to conduct target image classification tasks on the target datasets Flowers102, OxfordPets, DTD, and Food101. ILM consistently finds more interpretable target-source label mappings than FLM, in terms of colors, scenes, shapes, and textures. See **Fig. A3** for more examples.

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

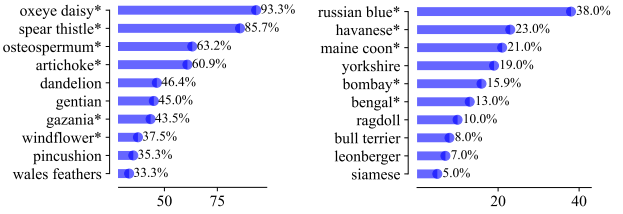
In this section, we empirically demonstrate the effectiveness of our proposed ILM-VP method by comparing it with a variety of baselines across multiple datasets, models, and learning paradigms.

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].

Source Model	ResNet-18 (ImageNet-1K)					ResNet-50 (ImageNet-1K)				ResNeXt-101-32x8d (Instagram)			
Method	Ours ILM-VP	Prompt baseline RLM-VP	Prompt baseline FLM-VP	Finetuning LP	Finetuning FF	Ours ILM-VP	Prompt base. FLM-VP	Finetuning LP	Finetuning FF	Ours ILM-VP	Prompt base. FLM-VP	Finetuning LP	Finetuning FF
Parameter Size	0.05M	0.05M	0.05M	0.51M	11.7M	0.05M	0.05M	0.51M	25.6M	0.05M	0.05M	0.51M	88.8M
Flowers102	**27.9**±0.7	11.0±0.5	20.0±0.3	88.0±0.5	97.1±0.7	**24.6**±0.6	20.3±0.3	90.9±0.4	97.9±0.7	**27.9**±0.3	22.5±0.5	89.1±0.2	99.2±0.5
DTD	**35.3**±0.9	16.3±0.7	32.4±0.5	60.0±0.6	65.5±0.9	**40.5**±0.5	36.9±0.8	67.6±0.3	69.7±0.9	**41.4**±0.7	40.3±0.5	69.7±0.2	69.1±1.0
UCF101	**23.9**±0.5	6.6±0.4	18.9±0.5	63.2±0.8	73.0±0.6	**34.6**±0.2	33.9±0.4	70.8±0.3	78.0±0.8	**43.1**±0.8	41.9±0.6	76.9±0.5	79.1±0.7
Food101	**14.8**±0.2	3.8±0.3	12.8±0.1	50.6±0.3	75.4±0.8	**17.0**±0.3	15.3±0.2	57.6±0.5	80.3±0.9	**23.0**±0.4	20.5±0.5	76.0±0.4	82.5±0.3
GTSRB	**52.0**±1.2	46.1±1.3	45.5±1.0	77.4±1.2	98.0±0.3	**52.5**±1.4	47.6±1.1	77.8±0.7	97.6±1.0	**59.9**±1.0	56.2±0.6	73.5±0.7	97.6±0.9
EuroSAT	**85.2**±0.6	82.4±0.4	83.8±0.2	93.8±0.3	98.8±0.5	83.6±0.7	**84.8**±0.3	95.7±0.2	98.9±0.6	86.2±0.8	**87.8**±0.4	93.4±0.3	98.9±0.7
OxfordPets	**65.4**±0.7	9.3±0.4	62.9±0.1	87.2±0.6	87.8±0.5	76.2±0.6	**76.4**±0.2	90.4±0.3	91.9±0.4	**78.9**±0.8	76.8±0.6	93.6±0.4	90.1±0.9
StanfordCars	**4.5**±0.1	0.9±0.1	2.7±0.1	33.8±0.2	81.0±0.1	**4.7**±0.2	4.2±0.3	40.6±0.1	86.4±0.3	**7.0**±0.2	4.6±0.1	64.7±0.1	92.5±0.2
SUN397	**13.0**±0.2	1.0±0.1	10.4±0.1	46.1±0.2	53.2±0.2	**20.3**±0.2	19.8±0.1	53.5±0.1	59.0±0.1	**23.7**±0.2	21.6±0.3	62.3±0.1	61.0±0.2
CIFAR10	65.5±0.1	63.0±0.1	**65.7**±0.6	85.9±0.5	96.5±0.4	**76.6**±0.3	74.8±0.5	90.1±0.1	96.6±0.2	**81.7**±0.3	80.3±0.3	94.1±0.1	97.1±0.1
CIFAR100	**24.8**±0.1	12.9±0.1	18.1±0.2	63.3±0.8	82.5±1.2	**38.9**±0.3	32.0±0.4	70.7±0.7	83.4±0.9	**45.9**±0.2	39.7±0.2	76.2±0.9	84.6±1.2
SVHN	**75.2**±0.2	73.5±0.3	73.1±0.2	65.0±0.2	96.5±0.3	**75.8**±0.4	75.6±0.2	63.5±0.2	96.9±0.3	**81.4**±0.1	79.0±0.5	51.0±0.2	97.1±0.3
ABIDE	**76.9**±2.1	74.0±2.2	73.1±1.6	65.4±3.8	60.6±4.2	63.5±2.2	**64.4**±3.4	55.8±2.6	70.2±2.5	**67.3**±2.6	65.7±3.4	54.8±3.4	73.1±4.2

Tab. 1. Performance overview of our proposed VP method (ILM-VP), prompt baseline methods (RLM-VP and FLM-VP), and finetuning methods (LP and FF) over 13 target image classification datasets using 3 pretrained source models (ResNet-18 on ImageNet-1K, ResNet-50 on ImageNet-1K, and ResNeXt-101-32x8d on Instagram). In each cell, a±b refers to the mean and standard deviation of target task accuracies (%) over 3 independent trials. The highest accuracy across VP-based methods is marked in **bold**. 'Parameter Size' refers to the number of trainable parameters in the input prompt or model finetuning.



(a) Flowers102 (b) OxfordPets

Fig. 4. ILM-VP's class-wise accuracy improvements over FLM-VP using ImageNet-1K pre-trained ResNet-18 under the target dataset (a) Flowers102, (b) OxfordPets. Target classes with $\star$ refer to those with updated source labels compared to the prompting baseline method FLM-VP.

Baselines and evaluations. In the VP paradigm, our baseline methods include the random LM-based VP (**RLM-VP**) [2, 14], and the frequency LM-based VP (**FLM-VP**) [1, 15]. We highlight that VP is a *finetuning-free* method to drive the source model to conduct target image classification tasks. When implementing VP baselines, we follow their official repository setups. We also refer readers to Appendix D for detailed implementations of ILM-VP and baseline methods. In addition to prompting methods, we also cover finetuning-based methods, including linear probing (**LP**) and end-to-end full finetuning (**FF**). Since finetuning modifies source model parameters, it requires training more parameters and is more computationally intensive.

We evaluate the performance of all the methods by the target task accuracy at the testing time and the efficiency in terms of the parameter size that VP or finetuning needs to handle. We also leverage a post-hoc model explanation method, known as Explanation-by-Example (**EBE**) [54], to assess the quality of visual explanation of different VP methods. The core idea of EBE is to find train-time data points that have the most similar feature representations to that of a queried test datapoint so as to use these identified training samples to explain the model's prediction on this test sample. In the context of VP, EBE can aid us to find the source training samples explainable for model prediction on prompted target test data, like source examples in **Fig. 3**.

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-vision 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-

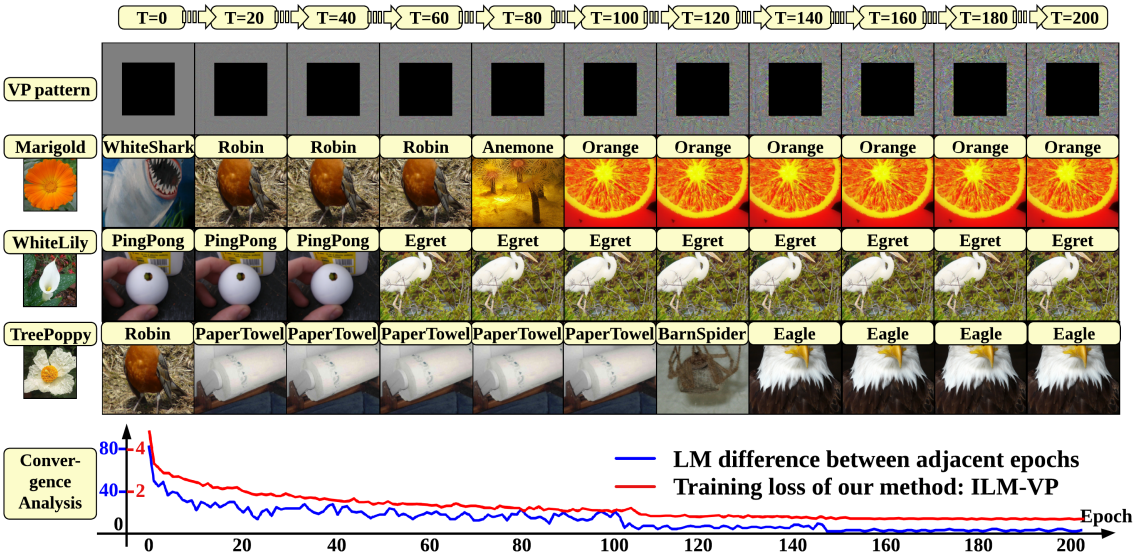


Fig. 5. ILM-VP training dynamics from epoch 0 to 200. Rows show: (1) VP pattern vs. epoch number; (2-4) Learned source label mapping with respect to target label 'Marigold', 'White Lily', and 'Tree Poppy', together with EBE-identified source training examples to explain each re-purposed target label; (5) Convergence of training loss and LM difference between adjacent epochs measured by Hamming distance.

creases 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-

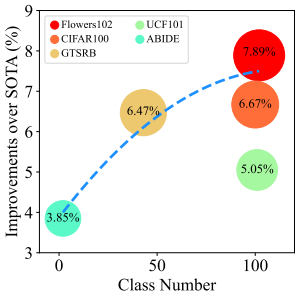


Fig. 6. ILM-VP's improvements over FLM-VP on representative datasets (datasets with improvements over 3%) using ResNet-18. The dashed line is the fitted curve.

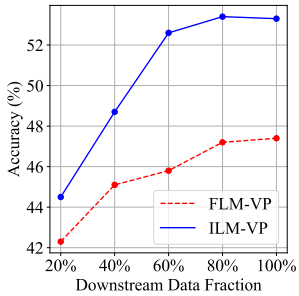


Fig. 7. ILM-VP and FLM-VP performance on different fractions of GTSRB dataset (43 classes and more than 900 training samples per class) using ResNet-18.

tition. **Fig. 7** compares the performance of ILM-VP with FLM-VP vs. target training dataset size (training from 20% to 100% of the entire set). As we can see, ILM-VP consistently outperforms the baseline FLM-VP and the improvement becomes more significant as the data scale grows.

6. Extension: LM in Text Domain for CLIP

In the previous sections, we show that LM could play a vital role in VP when reprogramming a pre-trained vision model to conduct downstream targeted vision tasks. In this section, we shift our focus from the vision source model to the vision-language model, specific to CLIP (contrastive language–image pretraining), which has received increasing attention in the area of VP [2]. We will show that although CLIP does *not* require source-target mapping across *image labels* (due to its multi-modal learning architecture), the proposed idea on iterative LM can be extended to conduct *text prompt selection* to improve target task accuracy.

LM for CLIP. Different from the vision-only model, CLIP can directly take target data labels as its textual inputs so as to mitigate the issue of source-target label mapping; see **Fig. A5** for an illustration. In this setting, LM seems redundant. However, this is only applied to VP in the image domain. We argue that CLIP still needs implicit LM for *text labels*, considering the diversity of text prompts (**TPs**) [26]. That is, CLIP can incorporate a text label into different context prompt templates (81 templates) suggested in [26] to create multiple text label instances. For example, the target label 'dog' can be combined with context prompts 'A photo of a big {label}' and 'A photo of a small {label}'. Thus, given m context prompts and K_{t} target data labels, we can create mK_{t} 'virtual source labels', which should be mapped to K_{t} target labels. Thus, the LM problem arises, and its optimal solution characterizes the optimal prompt selection in the text domain for prompted image data. Similar to the BLO method (5), we can bake iterative LM into VP using the CLIP model by replacing the lower-level image label mapping with the context-fused text label mapping. BLO then gives a unified prompt learning framework that can be easily compatible with CLIP. We refer readers to Appendix E for more implementation details.

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 predefined, 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 {}' instead of 'This is a photo of {}' for the target image with the label 'buttercup'. Another example is that VP for CIFAR10 prefers the text prompt 'a pixelated photo of a {}'. 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**.

Methods	VP+TP Acc(%)	Ours (VP+TP+LM) Acc(%)	Ours (VP+TP+LM) Examples of context prompt template → target label
Flowers102	70.0	**83.7**	a close-up photo of a {} → buttercup
DTD	56.8	**63.9**	graffiti of a {} → blotchy
UCF101	66.0	**70.6**	a {} in a video game → baseball pitch
Food101	78.9	**79.1**	a photo of the dirty {} → crab cake
SVHN	89.9	**91.2**	a photo of a {} → 7
EuroSAT	96.4	**96.9**	a pixelated photo of a {} → river
StanfordCars	57.2	**57.6**	the toy {} → 2011 audi s6 sedan
SUN397	60.5	**61.2**	a photo of a large {} → archive
CIFAR10	93.9	**94.4**	a pixelated photo of a {} → ship
ImageNet-R	67.5	**68.6**	a rendition of a {} → gold fish
ImageNet-Sketch	38.5	**39.7**	a sketch of a {} → eagle

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 {}") 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

[1] Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. *arXiv preprint arXiv:2007.08714*, 2020. 1, 2, 3, 4, 5, 6, 15

[2] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 1(3):4, 2022. 1, 2, 3, 6, 8, 16

[3] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2):199–210, 2010. 1

[4] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 1

[5] Muhammad Ghifary, W Bastiaan Kleijn, and Mengjie Zhang. Domain adaptive neural networks for object recognition. In *Pacific Rim international conference on artificial intelligence*, pages 898–904. Springer, 2014. 1

[6] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 1

[7] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009. 1

[8] Durjoy Sen Maitra, Ujjwal Bhattacharya, and Swapan K Parui. Cnn based common approach to handwritten character recognition of multiple scripts. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1021–1025. IEEE, 2015. 1

[9] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Self-taught clustering. In *Proceedings of the 25th international conference on Machine learning*, pages 200–207, 2008. 1

[10] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766, 2007. 1

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1

[12] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 1

[13] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 1

[14] Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. *arXiv preprint arXiv:1806.11146*, 2018. 1, 2, 3, 6, 7

[15] Lingwei Chen, Yujie Fan, and Yanfang Ye. Adversarial reprogramming of pretrained neural networks for fraud detection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2935–2939, 2021. 1, 2, 3, 6

[16] Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov, and Farinaz Koushanfar. Adversarial reprogramming of text classification neural networks. *arXiv preprint arXiv:1809.01829*, 2018. 1, 2, 3

[17] Paarth Neekhara, Shehzeen Hussain, Jinglong Du, Shlomo Dubnov, Farinaz Koushanfar, and Julian McAuley. Cross-modal adversarial reprogramming. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2427–2435, 2022. 1, 2, 3

[18] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A Efros. Visual prompting via image inpainting. *arXiv preprint arXiv:2209.00647*, 2022. 2

[19] Kihyuk Sohn, Yuan Hao, José Lezama, Luisa Polania, Huiwen Chang, Han Zhang, Irfan Essa, and Lu Jiang. Visual prompt tuning for generative transfer learning. *arXiv preprint arXiv:2210.00990*, 2022. 2, 3

[20] Yunhe Gao, Xingjian Shi, Yi Zhu, Hao Wang, Zhiqiang Tang, Xiong Zhou, Mu Li, and Dimitris N Metaxas. Visual prompt tuning for test-time domain adaptation. *arXiv preprint arXiv:2210.04831*, 2022. 2, 3

[21] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, and Yanning Zhang. Class-aware visual prompt tuning for vision-language pre-trained model. *arXiv preprint arXiv:2208.08340*, 2022. 2, 3

[22] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022. 2

[23] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. *arXiv preprint arXiv:2210.03117*, 2022. 2

[24] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[25] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 2

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 8

[27] Hao Yen, Pin-Jui Ku, Chao-Han Huck Yang, Hu Hu, Sabato Marco Siniscalchi, Pin-Yu Chen, and Yu Tsao. A study of low-resource speech commands recognition based on adversarial reprogramming. *arXiv preprint arXiv:2110.03894*, 2021. 2, 3

[28] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 2

[29] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 2

[30] Pin-Yu Chen. Model reprogramming: Resource-efficient cross-domain machine learning. *arXiv preprint arXiv:2202.10629*, 2022. 2

[31] Yang Zheng, Xiaoyi Feng, Zhaoqiang Xia, Xiaoyue Jiang, Ambra Demontis, Maura Pintor, Battista Biggio, and Fabio Roli. Why adversarial reprogramming works, when it fails, and how to tell the difference. *arXiv preprint arXiv:2108.11673*, 2021. 2

[32] Guanhua Zhang, Yihua Zhang, Yang Zhang, Wenqi Fan, Qing Li, Sijia Liu, and Shiyu Chang. Fairness reprogramming. *arXiv preprint arXiv:2209.10222*, 2022. 2, 3

[33] Aochuan Chen, Peter Lorenz, Yuguang Yao, Pin-Yu Chen, and Sijia Liu. Visual prompting for adversarial robustness. *arXiv preprint arXiv:2210.06284*, 2022. 2, 3

[34] Hadi Salman, Andrew Ilyas, Logan Engstrom, Sai Vemprala, Aleksander Madry, and Ashish Kapoor. Unadversarial examples: Designing objects for robust vision. *Advances in Neural Information Processing Systems*, 34:15270–15284, 2021. 3

[35] Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jinfeng Yi, Mingyi Hong, Shiyu Chang, and Sijia Liu. How to robustify black-box ml models? a zeroth-order optimization perspective. *arXiv preprint arXiv:2203.14195*, 2022. 3

[36] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 3

[37] Yimeng Zhang, Xin Chen, Jinghan Jia, Sijia Liu, and Ke Ding. Text-visual prompting for efficient 2d temporal video grounding. *arXiv preprint arXiv:2303.04995*, 2023. 3

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[40] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 5

[41] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018. 5

[42] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 5

[43] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 5

[44] 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. 5

[45] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014. 5

[46] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, 2013. 5

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

[48] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 5

[49] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5

[50] 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

[51] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 5

[52] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *cs.utoronto.ca*, 2009. 5

[53] Cameron Craddock, Yassine Benhajali, Carlton Chu, Francois Chouinard, Alan Evans, András Jakab, Budhachandra Singh Khundrakpam, John David Lewis, Qingyang Li, Michael Milham, et al. The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives. *Frontiers in Neuroinformatics*, 7:27, 2013. 5

[54] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. *Advances in Neural Information Processing Systems*, 2020. 6

[55] Junyang Wu, Xianhang Li, Chen Wei, Huiyu Wang, Alan Yuille, Yuyin Zhou, and Cihang Xie. Unleashing the power of visual prompting at the pixel level. *arXiv preprint arXiv:2212.10556*, 2022. 7

[56] Saachi Jain, Hadi Salman, Alaa Khaddaj, Eric Wong, Sung Min Park, and Aleksander Madry. A data-based perspective on transfer learning. *arXiv preprint arXiv:2207.05739*, 2022. 14