CPL: Counterfactual Prompt Learning for Vision and Language Models

Xuehai He1  Diji Yang1  Weixi Feng2  Tsu-Jui Fu2  Arjun Akula3  Varun Jampani3  Pradyumna Narayana3  Sugato Basu3  William Yang Wang2  Xin Eric Wang1

1UC Santa Cruz, 2UC Santa Barbara, 3Google
{xhe89,dyang39,xwang366}@ucsc.edu
{weixifeng,tsu-juifu,william}@ucsb.edu
{arjunakula,varunjampani,pradyn,sugato}@google.com

Abstract

Prompt tuning is a new few-shot transfer learning technique that only tunes the learnable prompt for pre-trained vision and language models such as CLIP. However, existing prompt tuning methods tend to learn spurious or entangled representations, which leads to poor generalization to unseen concepts. Towards non-spurious and efficient prompt learning from limited examples, this paper presents a novel Counterfactual Prompt Learning (CPL) method for vision and language models, which simultaneously employs counterfactual generation and contrastive learning in a joint optimization framework. Particularly, CPL constructs counterfactual by identifying minimal non-spurious feature change between semantically-similar positive and negative samples that causes concept change, and learns more generalizable prompt representation from both factual and counterfactual examples via contrastive learning. Extensive experiments demonstrate that CPL can obtain superior few-shot performance on different vision and language tasks than previous prompt tuning methods on CLIP. On image classification, we achieve 3.55% average relative improvement on unseen classes across seven datasets; on image-text retrieval and visual question answering, we gain up to 4.09% and 25.08% relative improvements across three few-shot scenarios on unseen test sets respectively.

1 Introduction

Pre-trained vision and language foundation models (Radford et al., 2021; Jia et al., 2021) have shown encouraging results toward open-domain visual-concept matching. Benefiting from prompt engineering (Song et al., 2022a; Liu et al., 2022), where free-form text prompts are designed for specific task goals, those foundation models can be easily transferred to a wide array of tasks under zero-shot and few-shot scenarios, including image classification (Deng et al., 2009), visual question answering (Shen et al., 2021), image-text retrieval (Jia et al., 2021), etc. But manually constructing prompts for vision and language models such as CLIP is a tedious, time-consuming process, which usually requires prior domain knowledge and leads to suboptimal solutions.

Prompt tuning (Lester et al., 2021), on the other hand, liberates us from manual prompt engineering and automates this process. Prompt tuning methods (Ju et al., 2021; Lin et al., 2014; Zhou et al., 2022) are proposed to effectively transfer CLIP to image recognition tasks after tuning a learnable prompt with a few examples of the classes. However, those methods purely conduct empirical risk minimization (ERM) and optimize for predictive accuracy, which often produces spurious, inefficient, or entangled representations (Wang and Jordan, 2021). Therefore, the generalization ability of existing prompt tuning methods for vision and language models is limited, and they often fail to transfer well to unseen classes or concepts. For example, the image classification performance of
the SOTA method CoCoOp (Zhou et al., 2022) is similar or even degrades on unseen classes when compared with zero-shot CLIP.

Learning non-spurious representation for better generalization requires disentangling features that causally determine the prompts. One solution is counterfactual reasoning. Counterfactual (“counter to the facts”) is a concept that describes the human capacity to learn from limited prior experiences by imagining the outcome of an alternative action that could have been taken. So we can do counterfactual intervention by asking “what if ...” questions in prompt learning. For example, as shown in Figure 1, a change in the visual feature of the barn would cause the label to change (if we view the two prompts as two labels).

Therefore, we introduce a new causality-based approach, Counterfactual Prompt Learning (CPL), for non-spurious and efficient prompt learning. First, we introduce a text-based negative sampling strategy to discover the most semantically-similar negative sample based on text similarity. Then we generate a counterfactual example by identifying minimal non-spurious feature change between semantically-similar positive and negative samples that causally causes prompt change. Finally, we adopt contrastive learning in the joint optimization framework (with counterfactual construction) to tune the learnable prompts using both factual and counterfactual examples. The causally fine-tuned prompts will eventually guide vision-and-language foundation models to distinguish images from unseen concepts, thereby improving the generalization ability of prompt learning.

We extensively evaluate CPL using seven standard datasets for image classification, two for image-text-retrieval, and one for visual question answering (VQA). We show that CPL outperforms the baseline on all three tasks: on image classification, our method achieves 3.55% average relative improvement on unseen classes across the seven datasets in terms of accuracy; on image-text retrieval, our method improves the most (4.09% relative improvement in terms of Recall@1) when using 0.5% of total training instances on MSCOCO (Lin et al., 2014) and Flickr30K (Plummer et al., 2015); on VQA, we gain up to 25.08% relative improvement in terms of accuracy on the VQAv2 (Goyal et al., 2017a) dataset.

Our main contributions are summarized below:

- We introduce Counterfactual Prompt Learning (CPL), a task-agnostic causality-based prompt learning method to effectively transfer CLIP to unseen concepts for different downstream tasks.
- We propose a text-based negative sampling strategy, where we compute BERTScore (Zhang et al., 2019) between text prompts, based on which we sample the most semantically-similar negative images.
- We introduce a joint optimization framework that simultaneously constructs counterfactuals by identifying minimal non-spurious feature change, and learns the generalized prompt representation from both factual and counterfactual examples.
- We conduct extensive experiments on image classification, image-text retrieval, and visual question answering, and validate the superiority of CPL to existing prompt tuning methods in transferring effectiveness on both seen and unseen concepts.

2 Related Work

Vision-and-Language Models. Vision-and-Language models pre-trained on large-scale image-text pairs have demonstrated great potential in multimodal representation learning (Jia et al., 2021; Yao et al., 2021; Yuan et al., 2021). Among them, the representative CLIP (Radford et al., 2021) benefits from 400M curated data and defines various prompt templates to carry out zero-shot image classification. However, those prompts still require hand-crafted designs. In this work, we automatically learn task-agnostic and task-relevant prompts without human priors. In addition, by considering the counterfactual examples, we can further improve various vision-and-language tasks, including visual question answering and image-text retrieval in a few-shot scenario.

Prompt Tuning. Many works focus on learning from discrete natural language prompts, e.g., Auto-Prompt (Shin et al., 2020) elicits knowledge from language models with automatically generated discrete prompts. Lately, many other works (Zhou et al., 2021, 2022) directly tune prompts in continuous vector forms. Guo et al. (2021) introduces Q-Learning to optimize the soft prompt. P-Tuning v2 (Liu et al., 2021) shows that continuous
prompt tuning achieves the same performance as fine-tuning in various settings. Prompt tuning also receives great interest in the computer vision domain. For example, CoOp proposes a continuous prompt optimization strategy to avoid prompt design. CoCoOp (Zhou et al., 2022) extends CoOp by further learning an instance-conditional network to generate an input-conditional token for each image. However, these methods trained with empirical risk minimization (ERM) may learn to rely on correlations between class labels and spurious attributes by minimizing average training error (Zhang et al., 2022). They usually learn spurious, inefficient, and entangled representation, lacking generalization ability to unseen scenarios.

**Counterfactual Reasoning.** A number of recent works have investigated generating counterfactual images (Besserve et al., 2020), or counterfactual text in specific language domains (e.g., court view (Wu et al., 2020), dialogue generation (Zhu et al., 2020), Natural Language Inference (Kaushik et al., 2019; Gokhale et al., 2021), named entity recognition (Zeng et al., 2020)); On the vision end, Zhang et al. (2021) proposes to add intervention over the changed domain on images during the data-generation process and steer the generative model to produce counterfactual features to augment the training process. Agarwal et al. (2020) uses automated semantic image manipulations to generate synthetic data to make models more robust against spurious correlations; On the vision and language end, Chen et al. (2020) proposes to generate counterfactual VQA samples by masking critical objects in images or words in questions to augment the training data and gain a huge improvement on the VQAv2 dataset. Gokhale et al. (2020) proposes template-based counterfactual image augmentation methods. Fu et al. (2020) proposes a novel training strategy for visual language navigation that dynamically generates counterfactuals to account for unseen scenarios. To our best knowledge, CPL is the first to apply counterfactual generation to prompt-based few-shot learning for vision and language models.

**Few-shot Learning.** Recently, several few-shot learners on vision-and-language tasks were proposed including CLIP. GPT (Brown et al., 2020), as a strong few-shot learner, is capable of performing a new language task by learning from only a few training instances. Frozen (Tsipoukelli et al., 2021) is developed based on GPT and made into a multimodal few-shot learner by expanding the soft prompting to include a collection of images and text. Their method demonstrates strong few-shot capabilities on visual question answering and image classification tasks. Similarly, CoCa (Yu et al., 2022) is pre-trained from scratch and end-to-end using both web-scale data and annotated images by considering all labels as text, therefore unifying supervision for learning representations through natural language. It can achieve state-of-the-art performance with few-shot transfer or by minimal task-specific adaptation on a wide range of downstream vision-and-language tasks, including visual recognition, multimodal understanding, crossmodal retrieval, and image captioning. SimVLM (Wang et al., 2021) is pre-trained with prefix language modeling on datasets with weak supervision. It exhibits its efficacy on few-shot captioning tasks. Even though all these models mentioned above can already achieve improvement on some few-shot tasks, how to exploit their few-shot reasoning ability using limited training examples still deserves the effort. In this work, we study this direction via the lens of prompt learning utilizing CLIP as a starting point.

### 3 Counterfactual Prompt Learning

#### 3.1 Problem Formulation

Our goal is to learn generalizable prompt representation with limited data. The prompt in CLIP is divided into two parts: task-agnostic prompt \( p \) and task-relevant prompt \( h \). Task-agnostic prompt \( p \) is learned end-to-end automatically. The set of task-relevant prompts \( \mathbb{H} = \{h_0, h_1, \ldots, h_C\} \) is mapped from the label space \( \mathbb{Y} \) with some predefined rules hinging on the task type, where \( C \) is the total number of classes. The final prompt \( t_c \) is the concatenation of the task-agnostic prompt and the task-relevant prompt fed into CLIP’s text encoder: \( t_c = [p, h_c] \).

Existing works to this problem (Zhou et al., 2021, 2022) propose to first extract visual feature \( v \) of each input image by feeding it into CLIP’s vision encoder \( F \); and text embeddings are generated by feeding \( \{t_c\}_{c=1}^C \) into the CLIP’s text encoder \( G \). The probability of \( i \)-th class is computed as

\[
p(t_i \mid x) = \frac{e^{<G(t_i), v>}}{\sum_{c=1}^{C} e^{<G(t_c), v>}},
\]  

(1)
where $\tau$ is the temperature parameter, $\langle \cdot , \cdot \rangle$ denotes the cosine similarity. Cross-entropy loss is then minimized and the gradients can be back-propagated via the text encoder $G$ to update the learnable prompt representation $p$. During training, the weights of CLIP always remain frozen. During inference, Eq. 1 is used to compute the probability for each class.

3.2 Method Overview

An overview of the Counterfactual Prompt Learning (CPL) framework is shown in Figure 2. For pre-processing, we construct task-relevant prompts for all training samples. The goal is to optimize the task-agnostic prompt $p$. During training, given a positive image-prompt pair, we first perform text-based negative sampling to find the most semantically-similar negative sample based on text similarity scores. Then we adopt a controllable counterfactual generation strategy to construct the counterfactual from the positive and negative samples in the visual feature space. Finally, we perform contrastive learning using both generated counterfactual image features and factual image features in a joint optimization framework to fine-tune the task-agnostic prompt $p$, allowing the model to understand non-spurious semantic information and learn generalized prompt representations.

3.3 Controllable Counterfactual Generation

By viewing image feature $v$ as a potential cause of the label, a non-spurious feature shall be a sufficient cause of the label. So we would like to generate counterfactuals by identifying minimal non-spurious feature change that causes the label change. The illustration of the counterfactual construction process is shown in Figure 3. Given positive image features $v$ and negative image features $v^-$, we can generate negative counterfactual image features $v'$ as below:

$$v' = (1 - u) \circ v + u \circ v^-,$$  \hspace{1cm} (2)

where $\circ$ is the element-wise multiplication and $u$ is the parameter controlling the amount of negative image feature that replaces the positive image feature. The negative image features are extracted from those images similar to the original image at the semantic level, which we will introduce in Section 3.4.

To capture the non-spuriousness, we would like to construct counterfactuals by replacing essential non-spurious features only. This can be achieved by minimizing the amount of feature change $u^*$ to the original image that can causally incur label
We then discuss how to perform negative sampling \( S \) where \( \text{MI} \) between \( v \) and \( c^\prime \) is predicted as a negative label \( c^- \) by the discriminator \( D \). \( u \) is minimized so a minimal change to the positive image feature \( v \) is captured here to causally change the label.

\[
\text{minimize } ||u^*||_1 \\
\text{s.t. } u^* = \arg \max_u D_{c^-}(v').
\]

(3)

Given the factual and counterfactual features \( v \) and \( v' \), we aim to learn the prompt that can help CLIP better align visual features \( v \) and textual features \( G(t) \) with same semantic meanings. This can be achieved by maximizing the mutual information (MI) between \( v \) and \( G(t) \). Therefore, by minimizing the InfoNCE loss (Hjelm et al., 2018), we can maximize the lower bound on MI(\( v, G(t) \)). To this end, we define the contrastive objective function based on the InfoNCE estimator following Khosla et al. (2020):

\[
\mathcal{L}_{CL}(p, u^*) = - \log \left( \frac{e^{S(v, G(t))}}{e^{S(v, G(t))} + e^{S(v', G(t))}} \right),
\]

(4)

where \( S (\cdot, \cdot) \) is normally the cosine similarity function and \( \tau \) is the temperature value.

### 3.4 Text-based Negative Sampling

We then discuss how to perform negative sampling for constructing counterfactual features. As suggested in Robinson et al. (2020), good negative samples have different labels and are difficult to be distinguished from an anchor point, while their semantic representations are close (Suresh and Ong, 2021). Since not all negative samples can serve as useful negatives (Chuang et al., 2020), indiscriminate leverage of these data may harm model robustness and algorithm efficiency. Therefore, during training, in each batch, we only utilize the most semantically-similar one to generate counterfactual image features. Other image samples are filtered out.

Semantic concepts may be highly complex in the visual representations, and thus it is hard to directly measure semantic similarity in the visual space. While language is more expressive and naturally preserves semantic meanings. Therefore, we propose a text-based negative sampling method. We first measure the text similarity between prompts with BERTScore (Zhang et al., 2019), which computes pairwise cosine similarity between reference sentences and candidate sentences using BERT contextual embedding (Devlin et al., 2019). We compute a similarity matrix with the value of each element being:

\[
sim(i, j) = \text{BERTScore}(h_i, h_j).
\]

(5)

Denote \( \mathcal{B} \) as the collection of sampled instances. During training, each prompt \( h_c \in \mathcal{B} \) \((1 \leq c \leq C)\), where \( C \) is the size of sampled instances) can be treated as a query. Given a query prompt \( h_q \), its most semantically similar prompt (the one with the highest BERTScore) \( h_{k_q} \) is searched from \( \mathcal{B} \). Then we use the CLIP vision encoder to obtain the features of the corresponding positive and negative images \( v \) and \( v^- \).

3.5 Joint Optimization

In addition to the contrastive learning loss as introduced in Eq. 4, we also adopt the standard cross-entropy loss for training:

\[
\mathcal{L}_{CE}(p) = - \sum_c y_c \log p(t_c \mid x),
\]

(6)

where \( y_c \) denotes the one-hot ground-truth annotation of the label. We treat all downstream tasks in this work as classification tasks, where the model predicts if the image and text prompt pair is matched or not.

Then the task-agnostic prompt \( p \) is learned by minimizing the weighted combination of contrastive learning loss and cross-entropy loss:

\[
\mathcal{L}(p) = \mathcal{L}_{CE}(p) + \lambda \cdot \mathcal{L}_{CL}(p, u^*),
\]

(7)

where \( \lambda \) determines the weight of \( \mathcal{L}_{CL} \).

In fact, we can seek to put Eq. 3 and Eq. 7 in a single-stage optimization framework. The intuition is that we generate counterfactual image
Algorithm 1 Counterfactual Prompt Learning

1: \( X \): image space
2: \( Y \): label space
3: \( h_i \): task-relevant prompt for the \( i \)-th class
4: \( \mathbb{H} \): the set of task-relevant prompts
5: \( p \): the task-agnostic prompt
6: \( v \): image features
7: \( v^- \): negative image features
8: \( u \): parameter controls the generation of counterfactual image features
9: function \( \text{CPL}(X, Y) \)
10: \( \mathbb{H} \leftarrow \emptyset \)
11: for each \( i \) do
12: \( \mathbb{H} \leftarrow [p, h_i] \)
13: \( \mathbf{t}_i \leftarrow [p, h_i] \)
14: for each \( i, j \) do
15: \( v \leftarrow v_i \)
16: Find the index \( j \) that maximize \( \text{sim}(i, j) \) with the given index \( i \)
17: \( v' \leftarrow v_j \)
18: Generate counterfactual image features \( \triangleright \) Eq. 2
19: \( L_{CE} \leftarrow \text{cross-entropy loss} \) \( \triangleright \) Eq. 5
20: \( L_{CL} \leftarrow \text{contrastive loss} \) \( \triangleright \) Eq. 6
21: Update \( p \) and \( u \) with the joint optimization loss \( \triangleright \) Eq. 7
22: end for
23: end for
24: end function

Algorithm 1 presents the learning algorithm of CPL. In summary, given few input training samples \( \{ (x_1, y_1), \ldots, (x_n, y_n) \} \), CPL consists of three main steps: (1) compute the similarity matrix between different text prompts within the sampled batch; (2) generate counterfactual image features; (3) optimize \( p \) and \( u \) with contrastive learning loss and cross-entropy loss.

### 3.6 Task-relevant Prompt Construction

We construct task-relevant prompts \( \mathbb{H} \) for image classification, image-text retrieval, and visual question answering, respectively. For image classification, the prompts are class labels for each task; for image-text retrieval, captions for each image are adopted as prompts; for visual question answering, we first use a pre-trained generative T5 model (Raffel et al., 2019) to convert the question-answer pairs into declarative sentences referring to the VQA prompt generation method proposed in Song et al. (2022b). Then, motivated by Wei et al. (2022), we add additional category information into the prompt generated from templates based on the question type to help the model perform intermediate reasoning steps. Specifically, we add “The question is asking about yes or no” and “The question is asking about numbers” are added for Yes/No and Number questions.

### 4 Experiments

#### 4.1 Tasks and Datasets

**Image Classification.** We employ seven publicly available image classification datasets used in CLIP: SUN397 (Xiao et al., 2010), Caltech101 (Griffin et al., 2007), ImageNet (Deng et al., 2009), OxfordPets (Parkhi et al., 2012), StanfordCars (Krause et al., 2013), Flowers102 (Nilsback and Zisserman, 2008), and Food101 (Bossard et al., 2014). These datasets constitute a comprehensive benchmark, which covers a diverse set of vision tasks including the classification of generic objects, fine-grained image recognition, action classification, etc. To evaluate the generalization ability of methods, we split those datasets into seen and unseen classes. Only images in the seen classes will be used for training. The setting follows the few-shot evaluation protocol in CLIP, where we use 16 shots for training and full test sets for testing.

**Image-Text Retrieval.** We consider two datasets for image-text retrieval: MSCOCO (Lin et al., 2014) and Flickr30K (Plummer et al., 2015). We adopt the widely used Karpathy split (Karpathy and Fei-Fei, 2015) for both the MSCOCO and Flickr30K datasets, where MSCOCO contains 113/5K/5K for train/validation/test. Flickr30K contains 29K/1K/1K images for train/validation/test. We construct few-shot setting subsets for both CoCoOp and CPL by taking 0.5%, 1%, and 3% of training instances. We train the model with the sub-
Counterfactual Prompt Learning.

We mainly compare CPL with CoCoOp (Zhou et al., 2022), one of the earliest prompt tuning methods proposed for vision-and-language pre-trained models. CoCoOp considers each input image and injects the learnable instance-aware tokens into the context vectors as the final prompt. For a fair comparison, both CPL and CoCoOp adopt CLIP (Radford et al., 2021) as the pre-trained vision-and-language backbone and are compared with respect to their relative improvements over zero-shot CLIP.

**Prompt Tuning.** The task-agnostic prompt is randomly initialized from a zero-mean Gaussian distribution with the standard deviation 0.02, where we set length \( L = 4 \) by default. For vision and language tasks, in contrast to image classification, where an image is labeled by a category, the task-relevant prompts comprise more fine-grained details, usually a sentence. We here similarly tokenize the whole sentence using the CLIP word.

The questions are first converted into a masked template using the pre-trained T5 model and predefined rules. The infilled template along with the question will be turned into prompts that naturally connect questions and answers. The model will predict whether the given prompt and image pairs are matched. We construct the few-shot setting by taking 0.5\%, 1\%, and 3\% instances for training.

**Visual Question Answering.** VQAv2 (Goyal et al., 2017b) is an extended dataset from the VQA (Antol et al., 2015) dataset. We use the official split VQAv2 version. The questions are categorized into three types: *Number*, *Yes/No*, and *Other*. We set up the experiments following Anderson et al. (2018), which treats visual question answering as a classification problem: for each question, the model picks the corresponding answer from a given set of predefined most frequent candidate answers and matches it with the image. The questions are first converted into a masked template using the pre-trained T5 model and predefined rules. The infilled template along with the questions will be turned into prompts that naturally connect questions and answers. The model will predict whether the given prompt and image pairs are matched. We construct the few-shot setting by taking 0.5\%, 1\%, and 3\% instances for training.

We set up the experiments following Anderson et al. (2018), where an image is labeled by a category, the task-relevant prompts comprise more fine-grained details, usually a sentence. We here similarly tokenize the whole sentence using the CLIP word.

### Table 1: Result comparison between CPL and CoCoOp (Zhou et al., 2022) on seen and unseen classes across seven image classification datasets in terms of accuracy (%) under the few-shot setting. The relative difference (%) compared with CLIP is reported in color.

<table>
<thead>
<tr>
<th>Training data used</th>
<th>Method</th>
<th>Flickr30k</th>
<th>MSCOCO</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CLIP</td>
<td>83.00</td>
<td>53.35</td>
<td>68.18</td>
</tr>
<tr>
<td>0.5%</td>
<td>CoCoOp</td>
<td>82.40 (+0.72)</td>
<td>55.55 (+4.12)</td>
<td>68.98 (+1.17)</td>
</tr>
<tr>
<td></td>
<td>CPL (ours)</td>
<td>85.64 (+3.18)</td>
<td>57.91 (+8.55)</td>
<td>71.78 (+5.28)</td>
</tr>
<tr>
<td>1%</td>
<td>CoCoOp</td>
<td>84.80 (+2.17)</td>
<td>56.62 (+6.13)</td>
<td>70.71 (+3.71)</td>
</tr>
<tr>
<td></td>
<td>CPL (ours)</td>
<td>86.91 (+4.71)</td>
<td>58.43 (+9.52)</td>
<td>72.67 (+6.59)</td>
</tr>
<tr>
<td>3%</td>
<td>CoCoOp</td>
<td>85.90 (+3.49)</td>
<td>58.08 (+8.87)</td>
<td>71.99 (+5.59)</td>
</tr>
<tr>
<td></td>
<td>CPL (ours)</td>
<td>87.74 (+5.71)</td>
<td>59.96 (+12.39)</td>
<td>73.85 (+8.32)</td>
</tr>
</tbody>
</table>

### Table 2: Result comparison between CPL and CoCoOp on two image-text retrieval datasets, Flickr30k (Plummer et al., 2015) and MSCOCO (Lin et al., 2014), on the unseen test sets in terms of Recall@1 (%). Few-shot settings include using 0.5\%, 1\%, and 3\% of instances to train. The relative difference (%) over CLIP is reported in color.

<table>
<thead>
<tr>
<th>Training data used</th>
<th>Method</th>
<th>Flickr30k</th>
<th>MSCOCO</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CLIP</td>
<td>83.00</td>
<td>53.35</td>
<td>68.18</td>
</tr>
<tr>
<td>0.5%</td>
<td>CoCoOp</td>
<td>82.40 (+0.72)</td>
<td>55.55 (+4.12)</td>
<td>68.98 (+1.17)</td>
</tr>
<tr>
<td></td>
<td>CPL (ours)</td>
<td>85.64 (+3.18)</td>
<td>57.91 (+8.55)</td>
<td>71.78 (+5.28)</td>
</tr>
<tr>
<td>1%</td>
<td>CoCoOp</td>
<td>84.80 (+2.17)</td>
<td>56.62 (+6.13)</td>
<td>70.71 (+3.71)</td>
</tr>
<tr>
<td></td>
<td>CPL (ours)</td>
<td>86.91 (+4.71)</td>
<td>58.43 (+9.52)</td>
<td>72.67 (+6.59)</td>
</tr>
<tr>
<td>3%</td>
<td>CoCoOp</td>
<td>85.90 (+3.49)</td>
<td>58.08 (+8.87)</td>
<td>71.99 (+5.59)</td>
</tr>
<tr>
<td></td>
<td>CPL (ours)</td>
<td>87.74 (+5.71)</td>
<td>59.96 (+12.39)</td>
<td>73.85 (+8.32)</td>
</tr>
</tbody>
</table>

### Table 3: Result comparison on the VQA2 dataset (Goyal et al., 2017a) in terms of accuracy (%). Few-shot settings include using 0.5\%, 1\%, and 3\% of instances to train. The relative improvements over CLIP are reported in color. Incorporating category information into task-relevant prompts can further improve the performance.

<table>
<thead>
<tr>
<th>Training data used</th>
<th>Method</th>
<th>VQA2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CLIP</td>
<td>11.83</td>
</tr>
<tr>
<td>0.5%</td>
<td>CoCoOp</td>
<td>27.98 (+136.52)</td>
</tr>
<tr>
<td></td>
<td>CPL w/o. Category Information</td>
<td>31.68 (+167.79)</td>
</tr>
<tr>
<td></td>
<td>CPL</td>
<td>33.39 (+182.25)</td>
</tr>
<tr>
<td>1%</td>
<td>CoCoOp</td>
<td>28.51 (+141.00)</td>
</tr>
<tr>
<td></td>
<td>CPL w/o. Category Information</td>
<td>34.70 (+193.32)</td>
</tr>
<tr>
<td></td>
<td>CPL</td>
<td>35.66 (+204.44)</td>
</tr>
<tr>
<td>3%</td>
<td>CoCoOp</td>
<td>30.18 (+155.11)</td>
</tr>
<tr>
<td></td>
<td>CPL w/o. Category Information</td>
<td>35.41 (+199.32)</td>
</tr>
<tr>
<td></td>
<td>CPL</td>
<td>36.32 (+207.02)</td>
</tr>
</tbody>
</table>
embedding (Radford et al., 2021), and feed the tokenized results to the text encoder with task-agnostic prompt vectors, to generate the language embedding for each prompt. In both the image-text retrieval and visual question answering, all data in the test set can be treated as belonging to unseen classes.

### 4.3 Main Results

#### Image Classification

The experimental results for image classification are shown in Table 1. With better prompts learned from counterfactual examples, our CPL method achieves clear advantages over CoCoOp for both seen and unseen classes across almost all datasets. Particularly on unseen classes, we gain an average relative improvement of 3.55%. Meanwhile, CoCoOp shows its poor generalization ability. Specifically, we found that CoCoOp performs worse than CLIP on StandfordCars on both seen and unseen classes, and on Caltech101 and Flower102 on unseen classes, indicating that it tends to learn and leverage spurious relations and could not generalize well on unseen classes in some cases. We believe all these mentioned above can be sufficient evidence that the main idea of CPL, learning non-spurious prompt representation can aid CLIP adapting at test time, is practical.

#### Image-Text Retrieval

Table 2 reports results on image-text retrieval on the unseen test set. CPL can beat the zero-shot CLIP consistently across the three different settings, demonstrating that CPL can also learn better prompt representation and more effectively exploit the limited amount of data on image-text retrieval. Meanwhile, CoCoOp performs even worse than CLIP on Flickr30k using 0.5% training data, which suggests that a tiny quantity of training data for image-text retrieval can lead to spurious prompt representation if using naive instance-conditional prompt tuning method.

#### Visual Question Answering

For visual question answering, the results are shown in Table 3. As can be seen, CPL surpasses the baseline CoCoOp with a relative improvement of up to 25.08% when using 1% of training instances. This proves the concept that CPL can be effective on more complicated vision-and-language tasks. In fact, visual question answering is more challenging for zero-shot CLIP which is pre-trained for image-text matching. During pre-training, CLIP sees most sentences similar to captions in image-text retrieval and those captions can be directly used as prompts; while for VQA, question-answer pairs have to be adapted into declarative prompts. Therefore, zero-shot CLIP has poor performance on VQA, but few-shot prompt tuning via CPL can help reduce the prompt domain gap significantly. Apart from the vanilla CPL method, we examined another variant of CPL where we do not add additional category information into the prompt (denoted as CPL w/o. Category Information), the results indicate that constructing task-relevant prompts by adding categorical information contributes to the improvement.

### 4.4 Ablation Analysis

#### Negative Sampling

We compare the random sampling vs. BERTScore sampling over ImageNet for image classification, MSCOCO for image-text retrieval, and VQA v2 for visual question answering in Table 4. With more challenging negative examples, BERTScore sampling leads to more effective prompt tuning and overbeats random sampling on all three tasks. The qualitative visualizations of the two sampling strategies are shown in Figure 4, from which it can be seen that BERTScore-sampled images are much more semantically similar to the original images.

#### Non-spurious Feature Visualization

We visualize the heatmap of the learned non-spurious feature weights in the image level in Figure 4. The weights are mainly centralized on the semantically meaningful regions that are aligned to the text.
Table 4: Random sampling vs. BERTScore sampling for CPL over three tasks. On ImageNet, we measure the average accuracy across seen and unseen classes. On MSCOCO and VQAv2, we both use 1% of training instances for few-shot learning.

<table>
<thead>
<tr>
<th>Method</th>
<th>ImageNet</th>
<th>MSCOCO</th>
<th>VQAv2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random sampling</td>
<td>75.28</td>
<td>57.78</td>
<td>33.01</td>
</tr>
<tr>
<td>BERTScore sampling</td>
<td>76.02</td>
<td>58.43</td>
<td>35.66</td>
</tr>
</tbody>
</table>

Figure 5: Accuracy comparison on ImageNet (Deng et al., 2009) unseen classes under three different shots. CPL performs better than CoCoOp consistently and has lower standard errors.

Number of Shots in Image Classification. We then study the effects of the number of shots on CPL for image classification. Following the few-shot evaluation protocol adopted in CLIP, we use 4, 8, and 16 shots for training on ImageNet. From Figure 5, increasing the number of shots keeps improving the performance of both two methods on unseen classes. Meanwhile, CPL outperforms CoCoOp under the three different settings and has lower standard errors.

Contribution of Contrastive Learning. In Section 3, we use the coefficient $\lambda$ to weigh the contrastive learning loss and combine it with the cross-entropy loss. It is observed that the scale of contrastive learning loss is smaller, hence we try to use a larger $\lambda$ to balance the two loss terms. Figure 6 shows the average accuracy result across seen and unseen classes on the SUN397 dataset under four different $\lambda$ values. Note that when $\lambda$ is zero, there is no contribution from the contrastive loss and the method actually learns the prompt using standard cross-entropy loss. From experimental results obtained on the SUN397 dataset, we can observe that using $\lambda = 1$ leads to the best performance.

Figure 6: Ablation of four different $\lambda$ values on the SUN397 dataset in terms of average accuracy (%). The performance of CPL peaks at $\lambda = 1$.

5 Conclusion

In this paper, we propose a Counterfactual Prompt Learning (CPL) framework to avoid time-consuming prompt engineering and learn more generalizable prompt representation for vision and language models. We conduct abundant experiments on seven widely used image classification datasets, two image-text retrieval datasets, and one visual question answering dataset. Our proposed CPL method outperforms the previous prompt tuning baseline and the zero-shot CLIP across the three tasks. In the future, we plan to develop more sophisticated methods based on CPL and extend CPL to other vision and language tasks.

Limitations

There are fairness issues in large pre-trained vision and language models such as CLIP. The proposed prompt learning method in this study automatically learns the prompt and does not address those issues in the pre-trained model. Considering the method is proposed for the few-shot setting, careful inspection and tuning are also needed when testing our method on other biased datasets. The methodologies proposed in Booth et al. (2021) and Wang et al. (2021a) may possibly be paired with CPL to potentially address the issues. Another limitation is the absence of explainability in CPL, which is a common problem with existing soft prompt tuning methods. Backmapping tuned soft prompts representation to natural language is a way for interpretation; however, due to the limited size of vocabulary used by CLIP during the training, prior methods such as searching for the nearest words in the embedding space can not accurately match the vector to natural language. Expanding the dictio-
nary size for CLIP embedding or developing more advanced back-mapping techniques can possibly address the limitation.

References


Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In CVPR.


Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017b. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In CVPR.


Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In International Conference on Machine Learning, pages 4904–4916. PMLR.


