Reproducibility Report for Counterfactual Prompt Learning

Chris Lin, Bhargavi Paranjape, Deekshita Doli
{clin25, bparan, deekdoli}@uw.edu

Reproducibility Summary

Scope of Reproducibility

This report describes a reproducibility study to the main claims made by He et al. (2022) about the performance of the proposed Counterfactual Prompt Learning (CPL) method in comparison to the state-of-the-art prompt tuning method Conditional Context Optimization (CoCoOp) (Zhou et al., 2022) on downstream image classification, image-text retrieval, and visual question answering. Besides reproducing the primary claims made by He et al. (2022), this report describes additional ablation experiments to further explore CPL performance. These include varying CPL prompt length, varying the weight of the CPL contrastive loss, using a different text similarity score for negative sampling, and using a larger vision encoder.

Methodology

The proposed method, Counterfactual Prompt Learning (CPL), aims to learn a task-agnostic prompt for the CLIP model (Radford et al., 2021). CPL generates a counterfactual image representation and uses it for contrastive learning to encourage the prompt to align with features present in the original image representation but absent in the counterfactual one. Text-based negative sampling is used to obtain the counterfactual representation, and a discriminator is trained to classify image representations. For the main claims from the original paper, the hyperparameters used in our experiments are consistent with those specified in the original paper. The code from the CPL public repository has been adapted and bugs have been fixed. The computational requirements for training CoCoOp and CPL for different datasets and tasks are estimated.

Results

For image classification, CPL outperforms CoCoOp on both seen and unseen classes, although the improvement of CPL over CoCoOp is not as significant as reported in the original paper. In the case of image-text retrieval, CPL performs better than CoCoOp, when trained with only 0.5% of the data. Finally, even in a low-resource setting, CPL outperforms CoCoOp by 24.32% relative accuracy in visual question answering. Additionally, by varying the learnable task-agnostic prompt length (L) and the weight of the CPL contrastive loss (λ), we found that the values of $L = 4$ and $\lambda = 0$ tend to produce the best results. We replaced the BERTScore similarity score for text-based negative sampling with BLUERT and observed that the choice of score does not significantly affect the performance of CPL. Lastly, we observed that using a larger CLIP image encoder yields better results.

What was Easy

Owing to the extensive documentation of the Dassl PyTorch toolbox and the CoCoOp code repository, processing the data, installing the dependencies, and creating the environment to run the baseline CoCoOp were easy.

What was Difficult

It was challenging to find missing files and consistently save model outputs in the original CPL code. Due to the lack of clear documentation, it was necessary to examine CPL's Dassl package to identify correct input arguments for scripts.

Communication with Original Authors

We opened two GitHub issues on the original author's code repository. One of them was answered in two weeks, while the other remains unanswered.

1 Introduction

When applying self-supervised vision-and-language models such as CLIP (Radford et al., 2021) for downstream tasks (e.g., image classification), performance depends on the prompt used to embed the task for the model. One approach to improve downstream performance is prompt tuning to optimize for task-specific predictive accuracy. However, He et al. (2022) argue that such optimization procedures can lead to prompts with spurious correlations with downstream classes and samples in the training set, limiting the generalization of these prompts to unseen classes and samples.

Counterfactual Prompt Learning (CPL) is a recent prompt tuning method aimed to improve the generalization of learned prompts to downstream unseen classes and samples (He et al., 2022). Overall, CPL seeks to tune a task-agnostic prompt using minimal counterfactual image representations, obtained through negative image sampling and counterfactual generation based on negative samples. The contributions of He et al. (2022) are (i) a text-based sampling approach for finding negative image samples; (ii) an optimization framework for identifying counterfactual image representations having small differences with the corresponding positive image representations; and (iii) combining (i) and (ii) into CPL and comprehensively evaluating CPL on image classification, image-text retrieval, and visual question answering.

2 Scope of Reproducibility

Addressed Claims from the Original Paper. In He et al. (2022), it is shown that CPL performs better than the previous state-of-the-art prompt tuning method Conditional Context Optimization (CoCoOp) (Zhou et al., 2022) in downstream image classification, image-text retrieval, and visual question answering. Particularly, He et al. (2022) highlight the superior generalization performance of CPL to unseen classes and samples. Therefore, in this report, we aim to reproduce the following three claims about the generalization performance of CPL in comparison to CoCoOp.

1. On image classification, CPL achieves a 3.22% average relative accuracy improvement over CoCoOp on unseen classes in SUN397, Caltech101, OxfordPets, StanfordCars, Flowers102, and Food101.

2. On image-text retrieval, when trained with 0.5% of the training data, CPL achieves a 3.93% relative recall@1 improvement over CoCoOp on unseen test samples in Flickr30k. When trained with 1% of the training data, CPL achieves a 2.49% relative recall@1 improvement over CoCoOp on unseen test samples in Flickr30k.

3. On visual question answering, when trained with 0.02% of the training data, CPL achieves a 19.34% relative accuracy improvement over CoCoOp on unseen test samples in VQAv2.

We note some differences between the above claims and the results reported in He et al. (2022), due to our limited computational capacity. First, the above claims exclude large datasets such as ImageNet for image classification and MSCOCO for image-text retrieval. Second, training with 3% Flickr30k training data is infeasible given our compute, so only claims with 0.5% and 1% training data are tested. Third, training with more than 0.5% VQAv2 training data exceeds our GPU memory, so we opt to test whether the relative CPL performance (with 0.5% VQAv2 training data) reported in He et al. (2022) holds with only 0.02% VQAv2 training data.

Additional Experiments Not in the Original Paper. To better understand CPL performance under different scenarios, we also perform the following ablation experiments.

1. (Varying CPL prompt length) In He et al. (2022), the task-agnostic prompt length of CPL is set to $L = 4$. We additionally evaluate CPL performance with $L = 8, 16, 32$ on Caltech101, Flowers102, and Food101.

2. (Varying the weight of the CPL contrastive loss for image-text retrieval) In He et al. (2022), the weight of the CPL contrastive loss is set to $\lambda = 1$ for image-text retrieval experiments. We additionally evaluate CPL performance on the image-text retrieval dataset Flickr30k with $\lambda = 0, 5, 10$.

3. (Using a different text similarity score for negative sampling) CPL uses BERTScore (Zhang et al., 2019) to compute text similarity for negative sampling. We additionally evaluate CPL performance on OxfordPets and StanfordCars when replacing BERTScore with BLEURT (Sellam et al., 2020), a BERT-based metric that models human judgements.

4. (Using a larger CLIP vision encoder) In He et al. (2022), the CLIP vision encoder ViT-B/16 is used in CPL. We additionally evaluate CPL performance on OxfordPets and StanfordCars when replacing ViT-B/16 (ViT-Base with 16 patches) with the larger CLIP vision encoder large ViT-L/16 (ViT-Large with 16 patches).

3 Methodology

3.1 Model Descriptions

Problem Formulation. The goal of CPL is to learn a task-agnostic prompt $\boldsymbol{p} \in \mathbb{R}_{+}^{L \times V}$, where L is the prompt length and V the vocabulary size. Given a downstream task and its class set $\{1, ..., C\}$, the final prompt for class $c \in \{1, ..., C\}$ is $\boldsymbol{t}_c = [\boldsymbol{p}, \boldsymbol{h}_c]$, where $\boldsymbol{h}_c \in \{0, 1\}^{* \times V}$ is a task- and class-specific prompt. Here, $*$ denotes the variable dimensionality of $\boldsymbol{h}_c$.

Given a CLIP text encoder G mapping to $\mathbb{R}^d$, a CLIP image encoder F also mapping to $\mathbb{R}^d$, where $d > 0$ is the embedding size, and an image $\boldsymbol{x}$, the predicted probability of the i-th class for $\boldsymbol{x}$ is computed as

$$p(\boldsymbol{t}_i|\boldsymbol{x}) = \frac{\exp\left(\frac{\langle G(\boldsymbol{t}_i), F(\boldsymbol{x})\rangle}{\tau}\right)}{\sum_{j=1}^{C} \exp\left(\frac{\langle G(\boldsymbol{t}_j), F(\boldsymbol{x})\rangle}{\tau}\right)}, \tag{1}$$

where τ is the temperature parameter optimized for the given CLIP model, and $\langle \cdot, \cdot \rangle$ denotes the cosine similarity. To achieve downstream performance, the parameters in $\boldsymbol{p}$ are then trained to minimize the cross entropy loss

$$\mathcal{L}_{CE}(\boldsymbol{p}) = -\sum_{c=1}^{C} \boldsymbol{y}_c \log p(\boldsymbol{t}_c|\boldsymbol{x}), \tag{2}$$

for a particular training image $\boldsymbol{x}$ with one-hot label $\boldsymbol{y} \in \{0, 1\}^C$. The parameters in F, G (from CLIP) remain frozen.

Counterfactual Image Representation Generation. Assuming that $\boldsymbol{v} \equiv F(\boldsymbol{x})$ is the cause of the label $\boldsymbol{y}$ for an image $\boldsymbol{x}$, CPL aims to train the task-agnostic prompt using only the minimal features in $\boldsymbol{v}$ such that the label remains unchanged. Suppose we have an image representation $\boldsymbol{v}^-$, where its corresponding label is different from $\boldsymbol{y}$. Then identifying the minimal feature set can be cast as finding a mask $\boldsymbol{u} \in [0, 1]^d$ such that the counterfactual image representation

$$\boldsymbol{v}' = (\boldsymbol{1} - \boldsymbol{u}) \odot \boldsymbol{v} + \boldsymbol{u} \odot \boldsymbol{v}^- \tag{3}$$

has a label different from $\boldsymbol{y}$, and the size of $\boldsymbol{u}$ is small. Here, $\odot$ denotes element-wise multiplication. To determine whether the label of $\boldsymbol{v}'$ is the same as the label of $\boldsymbol{v}$, a discriminator $D : \mathbb{R}^d \to [0, 1]^C$ is trained to classify each image representation. Overall, $\boldsymbol{u}$ can be obtained through the following optimization:

$$\min_{\boldsymbol{u}^*} \ \|\boldsymbol{u}^*\|_1 \tag{4}$$

$$\text{s.t. } \boldsymbol{u}^* = \arg\max_{\boldsymbol{u}'} D_{c^-}\Big((\boldsymbol{1} - \boldsymbol{u}') \odot \boldsymbol{v} + \boldsymbol{u}' \odot \boldsymbol{v}^-\Big), \tag{5}$$

where c^- denotes the aggregated negative class compared to the ground truth class in $\boldsymbol{y}$. The discriminator is a fully connected neural network with an input dimension d, a hidden layer with dimension $\lfloor d/16 \rfloor$, and an output dimension C, with the ReLU non-linearity (Agarap, 2018). Here, $\lfloor x \rfloor$ denotes the largest integer $\leq x$, for all $x \in \mathbb{R}$.

To encourage the prompt $\boldsymbol{t}_{c:\boldsymbol{y}_c=1}$ to align with features present in $\boldsymbol{v}$ but absent in $\boldsymbol{v}'$, the InfoNCE loss is used in CPL for contrastive learning:

$$\mathcal{L}_{CL}(\boldsymbol{p}, \boldsymbol{u}^*) = -\log\left(\frac{\exp\Big(\langle G(\boldsymbol{t}_{c:\boldsymbol{y}_c=1}), \boldsymbol{v}\rangle/\tau\Big)}{\exp\Big(\langle G(\boldsymbol{t}_{c:\boldsymbol{y}_c=1}), \boldsymbol{v}\rangle/\tau\Big) + \exp\Big(\langle G(\boldsymbol{t}_{c:\boldsymbol{y}_c=1}), \boldsymbol{v}'\rangle/\tau\Big)}\right). \tag{6}$$

Text-Based Negative Sampling. To obtain the negative image representation $\boldsymbol{v}'$, the task-specific prompt $\boldsymbol{h}'$ is first found through

$$\boldsymbol{h}' = \arg\max_{\boldsymbol{h}_k : k \in \{1,...,C\} \setminus q} \text{SIM}(\boldsymbol{h}_q, \boldsymbol{h}_k), \tag{7}$$

where q is the class of the training sample (i.e., $\boldsymbol{y}_q = 1$), and SIM is the BERTScore (Zhang et al., 2019). Then $\boldsymbol{v}'$ is an image sampled from the class corresponding to $\boldsymbol{h}'$.

Joint Optimization. Finally, all the objectives are combined together for joint optimization in the following optimization objective:

$$\min_{\boldsymbol{p}, \boldsymbol{u}^*} \ \mathcal{L}_{CE}(\boldsymbol{p}) + \lambda \cdot \mathcal{L}_{CL}(\boldsymbol{p}, \boldsymbol{u}^*) + \|\boldsymbol{u}^*\|_1 \tag{8}$$

$$\text{s.t. } \boldsymbol{u}^* = \arg\max_{\boldsymbol{u}'} D_{c^-}\Big((\boldsymbol{1} - \boldsymbol{u}') \odot \boldsymbol{v} + \boldsymbol{u}' \odot \boldsymbol{v}^-\Big), \tag{9}$$

where λ is a hyperparameter. The learnable parameters in CPL across training samples are $\boldsymbol{p} \in \mathbb{R}_{+}^{L \times V}$ and the parameters in the discriminator D, totaling $L \cdot V + (d+1) \cdot \lfloor d/16 \rfloor + (\lfloor d/16 \rfloor + 1) \cdot C$ parameters. The learnable parameter in CPL for each training sample is $\boldsymbol{u}^* \in [0,1]^d$, totaling d parameters.

3.2 Datasets

Table 1 summarizes datasets from He et al. (2022) included in this report. The code repository in He et al. (2022) includes instructions on how to process these datasets and split them into training and test sets.

Image classification datasets SUN397, Caltech101, OxfordPets, StanfordCars, Flowers102, and Food101 (Zhou et al., 2022) are evaluated in a 16-shot setting, with half of the classes included in the training set and 16 samples for each of these training classes. To construct the corresponding test sets with seen classes, the official validation or test samples (depending on official test set availability) from the same classes are used. For test sets with unseen classes, the remaining half of the classes are used instead.

Image-text retrieval (Flickr30K; (Plummer et al., 2015)) and visual question answering (VQAv2; (Antol et al., 2015)) are evaluated by reducing the training data to 0.5%, 1% and 3%, while another set of 1,000 samples are reserved for testing.

Table 1: Datasets from He et al. (2022) used in this report. Official training, validation, and test set sizes are shown.

Name	Training	Validation	Test
Image Classification			
SUN397	15,880	3,970	19,850
Caltech101	4,128	1,649	2,465
OxfordPets	2,944	736	3,669
StanfordCars	6,509	1,635	8,041
Flowers102	4,093	1,633	2,463
Food101	50,500	20,200	30,300
Image-Text Retrieval			
Flickr30k	29,000	1,000	1,000
Visual Question Answering			
VQAv2	96,536	40,000	96,536

3.3 Hyperparameters

For the addressed claims from the original paper in Section 2, all hyperparameters in our experiments follow those specified in He et al. (2022). That is, CoCoOp and CPL were trained with a prompt length of $L = 4$ using the SGD (stochastic gradient descent) optimizer with one warm-up epoch having learning rate $= 1 \times 10^{-5}$ followed by cosine annealing (Loshchilov & Hutter, 2016) starting at learning rate $= 0.002$. The weight of the CPL contrastive loss was set to $\lambda = 1$. The trained CLIP text encoder was BERT (Devlin et al., 2018), and the trained CLIP vision encoder was ViT-B/16 (Dosovitskiy et al., 2020). The corresponding CLIP temperature parameter τ was inherited from the CLIP model[1]. Besides the setting where random seed $= 1$ in He et al. (2022), we additionally ran model training with random seed $= 2, 3$. For image classification and image-text retrieval, CoCoOp and CPL were trained with 10 epochs, as in He et al. (2022). For visual question answering, due to limited computational time, CoCoOp and CPL were trained for 5 epochs instead of 10 epochs.

3.4 Implementation

We adapted and fixed bugs in the code from the CPL repository[2], where CoCoOp and CPL are implemented in PyTorch and the Dassl toolbox[3]. See `https://github.com/chris522229197/CPL` for our version of the adapted and bug-fixed code. Instructions for running the experiments proposed in Section 2 and for analyzing the experimental results are documented in our `README` file.

[1] `https://github.com/openai/CLIP`

[2] `https://github.com/eric-ai-lab/CPL`

[3] `https://github.com/KaiyangZhou/Dassl.pytorch`

3.5 Computational Requirements

Before carrying out all the experiments, we estimated that one trial (corresponding to one random seed) for each experiment would be sufficient for reproducing the results in He et al. (2022), because the original paper reported experiment values with only one trial. However, because the exact numeric values could not be reproduced with the same random seed, we ran each experiment with three trials (i.e., three random seeds). For all tasks, we estimated that 10 epochs could be run as in He et al. (2022). We were able to train CoCoOp and CPL for 10 epochs for image classification and image-text retrieval, but due to limited computational time, CoCoOp and CPL were trained for only 5 epochs for the visual question answering dataset VQAv2.

To estimate runtime for image classification and image-text retrieval, a pilot experiment with CoCoOp was run for Caltech101, which took 0.70 GPU (graphics processing unit) minutes per epoch on an NVIDIA GeForce RTX 2080 Ti GPU. The runtimes of CoCoOp and CPL were then estimated by linearly scaling the CoCoOp runtime on Caltech101 based on training set sizes. To estimate runtime for VQAv2, a pilot experiment with CoCoOp was run, taking 124.53 GPU minutes per epoch on an NIVIDIA A100 GPU. This runtime was directly taken to estimate the CPL runtime on VQAv2.

We can see from Table 2 that the CPL runtimes were more underestimated compared to CPL. This suggests that the BERTScore-based negative sampling and the generation of counterfactual image representations, which are the primary differences between CPL and CoCoOp, likely present computational bottleneck during CPL training.

Table 2: Estimated and actual runtime (in GPU minutes) and hardware for each experiment. Average (standard deviation) of actual runtimes across three runs are shown. NIVIDIA GPU model names (memory capacities in gigabytes) are shown. For Flickr30k, the runtime and hardware when CoCoOp and CPL are trained with 1% training data are shown. For VQAv2, the runtime and hardware when CoCoOp and CPL are trained with 0.02% training data are shown.

		SUN397	**Caltech101**	**OxfordPets**	**StanfordCars**	**Flowers102**	**Food101**	**Flickr30k**	**VQAv2**
Estimated									
Epoch runtime	CoCoOp	2.79	0.70	0.25	1.37	0.71	0.71	0.25	124.53
	CPL	2.79	0.70	0.25	1.37	0.71	0.71	0.25	124.53
Total runtime	CoCoOp	27.86	7.00	2.52	13.72	7.14	7.07	2.54	622.67
	CPL	27.86	7.00	2.52	13.72	7.14	7.07	2.54	622.67
Hardware	CoCoOp	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	A100 (80GB)
	CPL	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	A100 (80GB)
Actual									
Epoch runtime	CoCoOp	8.80 (0.16)	0.70 (0.01)	0.77 (0.05)	6.83 (0.12)	0.71 (0.00)	2.28 (0.08)	1.02 (0.01)	125.40 (4.00)
	CPL	142.34 (3.52)	35.78 (1.09)	29.19 (0.17)	65.09 (0.53)	38.80 (2.14)	41.30 (1.09)	11.47 (0.66)	436.40 (26.10)
Total runtime	CoCoOp	87.96 (1.62)	7.03 (0.14)	7.65 (0.50)	68.26 (1.21)	7.05 (0.04)	22.83 (0.79)	10.23 (0.05)	627.01 (19.99)
	CPL	1423.38 (35.18)	357.82 (10.86)	291.89 (1.70)	650.95 (5.26)	388.03 (21.38)	413.02 (10.85)	114.65 (6.56)	2181.98 (130.52)
Hardware	CoCoOp	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	A40 (48GB)	A40 (48GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	A100 (80GB)
	CPL	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	A40 (48GB)	A40 (48GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	RTX 2080 Ti (11GB)	A100 (80GB)

4 Results

This section provides a comprehensive overview of the various reproducibility experiments conducted to evaluate the performance of CPL. Section 4.1 focuses on testing the main claims in the original paper that CPL outperforms CoCoOp. Section 4.2 reports the results of additional experiments aimed at investigating the sensitivity of CPL to different hyperparameters and model components.

4.1 Addressed Claims from the Original Paper

Image classification. Training CoCoOp and CPL with the same hyperparameters and random seed did not produce exactly the same accuracy results, for both seen and unseen classes, as in He et al. (2022). Hence, we trained CoCoOp and CPL with three random seeds to compare the reported results in He et al. (2022) and our reproduced results.

As shown in Table 3, for seen classes, our reproduced CPL performs better than our reproduced CoCoOp in four out of the six image classification datasets, whereas the reported CPL performs better than the reported CoCoOp in five out of the six datasets. The relative average accuracy improvement of CPL compared to CoCoOp on seen classes is +0.86% in this report, which is smaller than the +1.73% reported in He et al. (2022).

For unseen classes, our reproduced CPL performs better than our reproduced CoCoOp in four out of the six datasets, whereas the reported CPL performs better than the reported CoCoOp in all six datasets. Finally, in this reproducibility report, CPL achieves a relative average accuracy improvement of +1.27% over CoCoOp on unseen classes. This relative improvement is smaller than the +3.22% relative average improvement reported in He et al. (2022). Our results reproduce the claim that CPL, when average over the six image classification datasets, achieves better accuracy than CoCoOp. However, the relative average improvement of CPL over CoCoOp is only reproduced to a lesser extent.

Table 3: Image classification test accuracy (in %) of CoCoOp and CPL reported by He et al. (2022) and reproduced in this report. For CPL, the relative accuracy differences (in %) with the corresponding CoCoOp accuracy results (based on classes and reported vs. reproduced) are denoted as relative Δ_{CoCoOp}. For the reproduced CoCoOp and CPL accuracy results, relative differences (in %) with the corresponding reported accuracy results (based on classes and methods) are denoted as relative Δ_{reported}.

Classes	Method	SUN397	Caltech101	OxfordPets	StanfordCars	Flowers102	Food101	Average
Reported in He et al. (2022)								
Seen	CoCoOp	79.08	97.66	95.18	70.91	94.65	90.67	88.02
	CPL [relative Δ_{CoCoOp}]	81.05 [+2.49]	97.70 [+0.04]	96.69 [+1.59]	75.51 [+6.49]	93.31 [-1.42]	93.01 [+2.58]	89.55 [+1.73]
Unseen	CoCoOp	76.83	93.92	97.78	73.09	69.24	91.53	83.73
	CPL [relative Δ_{CoCoOp}]	80.19 [+4.37]	94.94 [+1.09]	98.81 [+1.05]	78.90 [+7.95]	72.30 [+4.42]	93.44 [+2.09]	86.43 [+3.22]
Reproduced (average across three random seeds)								
Seen	CoCoOp	79.32	97.74	94.64	70.78	94.62	90.60	87.95
	CoCoOp relative Δ_{reported}	+0.30	+0.08	-0.57	-0.18	-0.03	-0.08	-0.09
	CPL [relative Δ_{CoCoOp}]	79.50 [+0.23]	97.93 [+0.19]	94.56 [-0.08]	74.67 [+5.50]	95.00 [+0.40]	90.56 [-0.04]	88.70 [+0.86]
	CPL relative Δ_{reported}	-1.91	+0.24	-2.20	-1.11	+1.81	-2.63	-0.94
Uneen	CoCoOp	76.90	92.61	97.28	73.06	71.94	91.20	83.83
	CoCoOp relative Δ_{reported}	+0.09	-1.39	-0.51	-0.04	+3.90	-0.36	+0.12
	CPL [relative Δ_{CoCoOp}]	76.75 [-0.20]	93.67 [+1.14]	97.93 [+0.67]	77.79 [+6.47]	71.89 [-0.07]	91.37 [+0.19]	84.90 [+1.27]
	CPL relative Δ_{reported}	-4.29	-1.34	-0.89	-1.41	-0.57	-2.22	-1.77

Image-text retrieval. Although we used identical hyperparameters and random seed, the recall@1 results obtained for CoCoOp and CPL were not precisely the same as reported in He et al. (2022). Therefore, we conducted additional training runs for CoCoOp and CPL, each with three different random seeds, to compare the reported outcomes with our replicated outcomes.

According to Table 4, our reproduced results show that CPL outperforms CoCoOp with 0.5% of the training data, but not with 1% of the training data. However, the results reported by He et al. (2022) indicate that CPL performs better than CoCoOp with both 0.5% and 1% of the training data.

We observe that CPL achieves a relative recall@1 improvement of +0.50% over CoCoOp when training with 0.5% of the data, but we also note a relative decrease of -0.18% when training with 1% of the data.

Visual question answering. We first attempted training CoCoOp and CPL with the same hyperparameters and random seed used in He et al. (2022). To simulate the few-shot setting for VQAv2, He et al. (2022) use 0.5%, 1% and, 3% of the training data. VQAv2 is cast into a classification problem that involves classifying each (question, image) pair into the appropriate answer category. A 0.5% subset of the VQAv2 training data corresponds to ≈19,000 questions, and consequently the same number of classes. CoCoOp and CPL both involve computing similarity of class prompts with image embeddings, for all C classes (see Equation (1)). We found that computing text embeddings for all ≈19,000 questions and answers require approximately 5 times the GPU space we had (2 A100 GPUs with ≈80GB each) even for 0.5% of the data.

We hence reduced the amount of training data to 0.02%, which is ≈4,000 training and test examples, and we report results for CoCoOp and CPL averaged over 3 runs in Table 5. Absolute performance for 0.02% of the training data is significantly lower than for 0.5% of the training data, which is expected. However, even in this low-resource setting, CPL performance is higher than CoCoOp performance (by 24.32%), verifying the claim about the better performance of CPL than CoCoOp for visual question answering with limited training data.

4.2 Additional Experiments Not in the Original Paper

Varying CPL prompt length. In He et al. (2022), the learnable task-agnostic prompt length is set to $L = 4$. We additionally trained CPL with $L = 8, 16, 32$ to see how the test performance of CPL varies with L for the datasets Caltech101, Flowers102, and Food101. From Figure 1, we can see that, for Caltech101 and Food101, the test accuracy results of CPL on seen and unseen classes have absolute differences within 1.5% across different prompt lengths. For Flowers102, the test accuracy results on seen classes have absolute differences within 4%, whereas the test accuracy

Table 4: Image-text retrieval test recall@1 (in %) of CoCoOp and CPL reported by He et al. (2022) and reproduced in this report. For CPL, the relative recall@1 differences (in %) with the corresponding CoCoOp recall@1 results (based on amount of training data used and reported vs. reproduced) are denoted as relative Δ_{CoCoOp}. For the reproduced CoCoOp and CPL recall@1 results, relative differences (in %) with the corresponding reported recall@1 results (based on amount of training data used and methods) are denoted as relative Δ_{reported}.

Training Data Used	**Method**	**Flickr30k**
Reported in He et al. (2022)		
0.5%	CoCoOp	82.40
	CPL [Δ_{CoCoOp}]	85.64 [+3.93]
1%	CoCoOp	84.80
	CPL [Δ_{CoCoOp}]	86.91 [+2.49]
Reproduced (average acorss three random seeds)		
0.5%	CoCoOp	82.77
	CoCoOp relative Δ_{reported}	+0.45
	CPL [Δ_{CoCoOp}]	83.12 [+0.42]
	CPL relative Δ_{reported}	-2.94
1%	CoCoOp	84.30
	CoCoOp relative Δ_{reported}	+2.31
	CPL [Δ_{CoCoOp}]	84.17 [-0.15]
	CPL relative Δ_{reported}	-3.15

Table 5: Visual question answering accuracy (in %) of CoCoOp and CPL reported by He et al. (2022) and reproduced in this report. For CPL, the relative accuracy differences (in %) with the corresponding CoCoOp accuracy results (based on amount of training data used and reported vs. reproduced) are denoted as relative Δ_{CoCoOp}. Since we were unable to generate results for 0.5% of the training data used, direct comparisons with reported results in He et al. (2022) are not made.

Training data used	**Method**	**VQAv2**
Reported in He et al. (2022)		
0.5%	CoCoOp	27.98
	CPL [relative Δ_{CoCoOp}]	33.39 [+19.34]
Reproduced		
0.02%	CoCoOp	13.78
	CPL [relative Δ_{CoCoOp}]	17.13 [+24.31]

results on unseen classes can differ as much as approximately 5%, with the default $L = 4$ having the best performance. Overall, these results suggest that CPL performance on image classification can vary with different prompt lengths, but using the default $L = 4$ tends to result in good or optimal performance.

Varying the weight of CPL contrastive loss. He et al. (2022) utilizes a CPL contrastive loss weight of $\lambda = 1$. To further investigate the effect of varying λ on the test performance of CPL, we conducted additional experiments on the Flickr30k dataset by training CPL with $\lambda = 1, 5, 10$. As illustrated in Figure 2, our results indicate that the recall@1 outcomes of CPL display differences of no more than 0.6% across different λ values when trained with 0.5% of the training data, and up to a 2% difference with 1% of the training data. Hence, our findings suggest that the performance of CPL in image retrieval can be influenced by the contrastive loss weight, and surprisingly using the weight $\lambda = 0$ yielded the best results.

Varying the sentence similarity score. He et al. (2022) use BERTScore Zhang et al. (2019) for text-based negative sampling—finding a negative image representation by sampling images from a class that is most similar to the class of the reference image. BERTScore between text prompts of reference class and all other classes is computed and the most similar class is chosen. In order to measure sensitivity of CPL to the choice of textual similarity measure, we replaced BERTScore with BlUERT Sellam et al. (2020), a learned evaluation metric based on BERT that can model human judgments with superior performance on evaluating summarization performance. We ran CPL with BLUERT with all other hyperparameters unchanged. While there is a small increase in performance, the gains are not significant

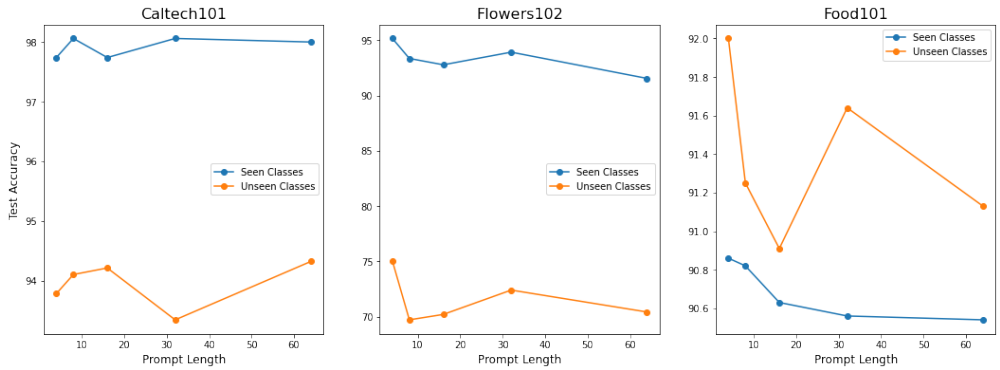

Figure 1: Test accuracy of CPL on seen and unseen classes for the datasets Caltech101, Flowers102, and Food101, with the learnable task-agnostic prompt length $L = 4, 8, 16, 32$. Results with random seed $= 1$ are shown.

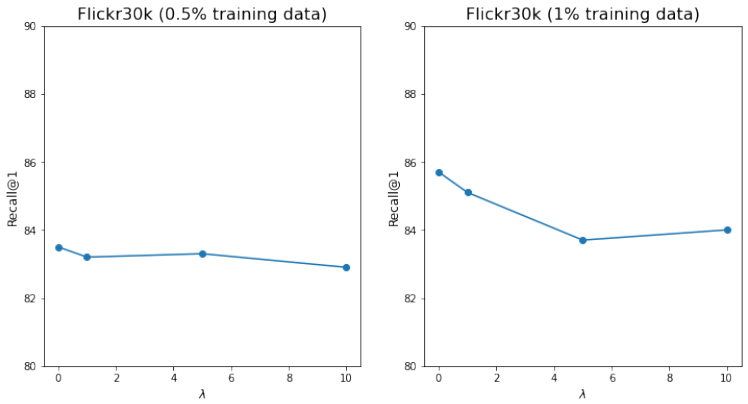

Figure 2: Recall@1 of CPL on unseen classes for the Flickr30k dataset, with the contrastive loss weight $\lambda = 0, 1, 5, 10$. Results with random seed $= 1$ are shown.

Table 6: CPL test accuracy (in %) on unseen classes for OxfordPets and StanfordCars, comparing performance with BERTScore (original) vs. BLEURT as the sampling score for identifying negative images, and performance with ViT-B/16 vs. ViT-L/16 as the CLIP image encoder. For CPL with BLEURT and ViT-L/16, relative accuracy differences (in %) with the original CPL are denoted by relative Δ_{original}.

	OxfordPets	StanfordCars
Varying sentence similarity score in CPL		
BERTScore (original)	94.56	74.67
BLEURT [relative Δ_{original}]	94.92 [+0.38]	74.94 [+0.36]
Varying CLIP image encoder in CPL		
ViT-B/16 (original)	94.56	74.67
ViT-L/16 [relative Δ_{original}]	95.68 [+1.18]	76.11 [+1.93]

(Table 6). Upon further analysis, we find that the most similar class does not change often enough to make a significant difference. We conclude that CPL is not sensitive to the choice of similar text similarity scores.

Varying the image encoder size. To test the sensitivity of CPL to the CLIP image encoder size, we replaced the original CLIP ViT-B/16 image encoder with the larger ViT-L/16. We ran CPL with ViT-L/16 with all other hyperparameters unchanged. We find that performance increased on both the OxfordPets and StanfordCars datasets, and performance gains are significant (Table 6). We conclude that CPL performance improves with a larger image encoder.

5 Discussion

Reproducing main claims from the original paper. For unseen classes in image classification, our reproduced CPL achieves a relative average accuracy improvement of +1.27% over our reproduced CoCoOp. This relative improvement is smaller than the +3.22% relative average improvement reported in He et al. (2022). In this report, for image-text retrieval, CPL displays a relative accuracy improvement of +0.50% over CoCoOp on 0.5% of the training data and a decrease of -0.18% on 1% of the training data. Compared to the reported relative performance gains of +4.76% and +2.94% achieved by CPL over CoCoOp with 0.5% and 1% data respectively in He et al. (2022), the observed improvement with CPL is smaller. For visual question answering, we had to reduce the amount of training data from 0.5% used in He et al. (2022) to 0.02% due to GPU constraints, resulting in significantly reduced absolute performance for CoCoOp and CPL. However, CPL performs better than CoCoOp performance (by 24.32% relative improvement) in this low-resource setting.

Additional Experiments Not in the Original Paper. CPL performance can vary with the task-agnostic prompt length, but using the default $L = 4$ tends to result in good or optimal performance on image classification. The contrastive loss term weight can impact the performance of CPL for image-text retrieval. Surprisingly, setting $\lambda = 0$ gives the highest recall@1 for Flickr30k. CPL is not sensitive to the choice of similar text similarity scores—replacing BERTScore with BLEURT does not lead to significant changes in CPL image classification performance. Finally, CPL performance improves with a larger CLIP image encoder.

5.1 What was Easy

Data processing. He et al. (2022) did not have accurate documentation for processing most of the image classification datasets. However, their code is based on the CoCoOp codebase[4], which has extensive documentation for dataset sources, training-vs-test-set splitting, and processing scripts. Therefore, downloading, splitting, and processing data became straightforward.

Code environment and documentation for CoCoOp. He et al. (2022) built their code repository on top of the Dassl PyTorch toolbox [5] and the CoCoOp codebase, both of which are well documented. Installing dependencies and creating the environment to run CoCoOp were straightforward. Consequently, running the CoCoOp baselines was easy.

5.2 What was Difficult

Bugs in CPL code. Multiple bugs in the CPL code by He et al. (2022) had to be fixed, totaling 165 lines of code changes before the reproducibility experiments for CPL could be run. These bugs include (i) syntactic errors, (ii) missing data and model configuration files, and (iii) the reuse of output directories, such that model output files are overwritten between experiments.

Lacking instructions for image-text retrieval and visual question answering. The documentation by He et al. (2022) does not include the instructions for evaluating image-text retrieval and visual question answering, requiring us to look into their modified Dassl package to figure out the appropriate input arguments.

5.3 Recommendations for Reproducibility

Configuration files, training bash scripts, and evaluation bash scripts should not be ignored in git updates and should be uploaded to the public repository. All commands in the codebase should be tested before publishing the public code repository. Computational devices and runtimes should be included in the original paper in order for others to assess computational feasibility.

Communication with Original Authors

We opened an GitHub issue to request missing model configuration files in the original authors' code repository[6]. The original authors responded in about two weeks. We opened another issue asking about a bug in their code[7], which has not been addressed since the passing of one week.

[4] https://github.com/KaiyangZhou/CoOp
[5] https://github.com/KaiyangZhou/Dassl.pytorch
[6] https://github.com/eric-ai-lab/CPL/issues/4
[7] https://github.com/eric-ai-lab/CPL/issues/6

References

Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Xuehai He, Diji Yang, Weixi Feng, Tsu-Jui Fu, Arjun Akula, Varun Jampani, Pradyumna Narayana, Sugato Basu, William Yang Wang, and Xin Eric Wang. CPL: Counterfactual prompt learning for vision and language models. *arXiv preprint arXiv:2210.10362*, 2022.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

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*, pp. 8748–8763. PMLR, 2021.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*, 2020.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16816–16825, 2022.