Proceedings of the 3rd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2025.22858

Few-Shot Image Classification Guided by Large Language Models

Pinzhe Chen[1*], Hao Chen[1], Yanxi Liu[1], Longhui Han[2]

[1]Ocean College, Jiangsu University of Science and Technology,Jiangsu, China
[2]School of Energy and Power, Jiangsu University of Science and Technology, Jiangsu, China
**Corresponding Author. Email: 18552718929@163.com*

Abstract: Few-shot image classification remains a challenging task due to the scarcity of labeled data for novel categories. Prompt-based tuning methods, such as CoOp and CoCoOp, have shown promise by adapting pre-trained vision-language models like CLIP to downstream tasks. However, these methods either suffer from limited generalization or lack semantic grounding. In this paper, we propose LLM-PromptTuning, a novel framework that integrates class-level semantic knowledge from large language models (LLMs) into conditional prompt learning. We generate rich, descriptive prompts for each class using an LLM and combine them with image-conditioned prompt vectors produced by a Transformer-based meta-network. These hybrid prompts are fed into the CLIP text encoder, improving the alignment between textual and visual representations. Experiments on Caltech101, Oxford Flowers102, and Food101 demonstrate that our method outperforms CoOp and CoCoOp, particularly in novel class recognition, achieving a new state-of-the-art in prompt-based few-shot learning.

Keywords: Few-shot Learning, Prompt Tuning, Vision-Language Models, Large Language Models, CLIP

1. Introduction

Few-shot image classification addresses the challenge of recognizing novel categories from only a handful of labeled examples. This problem is crucial because acquiring large labeled datasets for every new category is often impractical. Recent advances in contrastive vision-language models like CLIP have shown remarkable zero-shot classification abilities by learning joint image-text representations. By leveraging text prompts describing classes, CLIP can classify images without fine-tuning on those classes. However, designing optimal prompts is non-trivial, and CLIP's performance can be further improved by learning prompts from data [1]. Prompt learning has emerged as an effective way to adapt vision-language models to downstream tasks with limited data. Context Optimization (CoOp) is a pioneering method that learns a set of continuous prompt vectors for a given task instead of using hand-crafted text prompts [2]. CoOp excels on the base classes by fine-tuning a small set of prompt parameters while keeping CLIP's encoders frozen. Nevertheless, CoOp severely lacks generalization to novel classes, often experiencing a drastic drop in accuracy on those classes [3]. This overfitting to base classes stems from the prompt vectors being learned only on the training categories, thus failing to transfer to new categories. Conditional Context Optimization (CoCoOp) was proposed to improve generalization by generating image-dependent prompts. In CoCoOp, a

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.22858

lightweight meta-network takes each image as input and produces conditional prompt vectors, which are then combined with the class name token. This approach allows the prompt to vary with each image, providing robustness to category shifts. CoCoOp significantly improves performance on novel classes compared to CoOp, albeit with a slight trade-off on base class accuracy [4]. In this work, we propose a new method called LLM-PromptTuning to further enhance few-shot image classification by leveraging the knowledge of large-scale language models. Our approach is motivated by the observation that rich textual descriptions of class concepts can provide additional discriminative information to the vision-language model. While CoOp/CoCoOp optimize prompts in a data-driven manner, they do not explicitly inject external semantic knowledge about the classes [5]. Large Language Models (LLMs), on the other hand, are trained on vast text corpora and can generate informative descriptions for a given class name. We hypothesize that guiding prompt optimization with LLM-generated descriptions can significantly improve the alignment between image and text features, especially for novel classes. Innovation and Contributions: LLM-PromptTuning introduces a novel two-level prompt design that combines descriptive prompts from an LLM with conditional learned prompts in a CLIP framework. Specifically, we generate a unique descriptive text template for each class using an LLM, capturing key attributes of that class. We then develop an improved conditional prompt generator that produces input-conditioned prompt vectors, as in CoCoOp but with a more expressive architecture [6]. These two components are fused in the input embedding space to form a powerful text prompt for the CLIP text encoder. In summary, the contributions of this work include: 1) LLM-guided Prompt Generation: We leverage a pre-trained large language model to automatically generate rich textual descriptions for each class name, providing high-level semantic context that complements the few-shot training data. These descriptive prompts encode distinctive attributes of the class that help discriminate it from others. 2) Conditional Prompt Tuning with Self-Attention: We refine the conditional prompt learning approach of CoCoOp by replacing its simple meta-network with a multi-head self-attention based network. This network takes image features as input and outputs a set of context vectors that serve as image-conditioned prompt embeddings. The use of multi-head attention allows the model to focus on multiple aspects of the image, yielding a more informative and flexible prompt for each image. 3) Integrated CLIP-Style Framework: We integrate the LLM-generated textual template with the image-conditioned prompt vectors in CLIP's input space. By concatenating the descriptive text embeddings with the conditional context embeddings, we form an enriched prompt fed into CLIP; s text encoder. This synergy produces more precise image-text matching, improving classification accuracy. Our method is implemented on top of CLIP's zero-shot pipeline, requiring no changes to the pre-trained encoders, only additional prompt parameters and an LLM interface. 4) Improved Generalization: Extensive few-shot experiments on multiple benchmarks demonstrate that LLM-PromptTuning outperforms prior prompt learning methods. Notably, our method achieves higher accuracy on unseen classes than CoOp and CoCoOp, while maintaining competitive performance on seen classes. The gap between base and novel class accuracy is greatly reduced, indicating better generalization to new categories. We report consistent gains on diverse datasets, setting a new state-of-the-art in prompt-based few-shot learning.

2. Related work

Using paired image-text data to learn joint representations has empowered recent advances in zero-shot and few-shot learning. CLIP is a seminal model that trains an image encoder and a text encoder to align image and caption embeddings in a common space [7]. Trained on 400 million image-text pairs, CLIP learns broad visual concepts and can recognize new classes by comparing images with text prompts describing those classes [8]. Without any fine-tuning, CLIP achieved impressive zero-shot accuracy on ImageNet and other datasets by using prompts like "a photo of a classclassclass" for each class and predicting the closest text embedding to the image embedding. Following CLIP,

DOI: 10.54254/2755-2721/2025.22858

several other vision-language pretraining approaches have been developed. ALIGN extended this paradigm by using noisy alt-text data at a larger scale, achieving similar aligned representations [9]. Models such as CLOOB, DeCLIP, Florence, and UniCLIP explored improvements in training objectives and architecture, further boosting the quality of visual-textual features. These pre-trained models serve as foundation models for downstream tasks. By crafting appropriate text prompts, they enable zero-shot transfer to classification tasks without task-specific training. However, the choice of prompt can greatly affect performance. As a result, optimizing prompts for each task has become an important research direction, giving rise to methods like prompt tuning in both NLP and vision domains [10]. The limitation of CoOp is its poor generalization to new classes. Since the prompt vectors are optimized only for base classes, they tend to encode information specific to those classes, and fail to represent unseen classes well. Instead of a single static prompt for all images, CoCoOp produces the prompt dynamically based on the image content [11]. Specifically, Meta-Net takes the image feature from CLIP's image encoder and outputs a set of prompt embeddings, which are then combined with the class name embedding.

3. Method

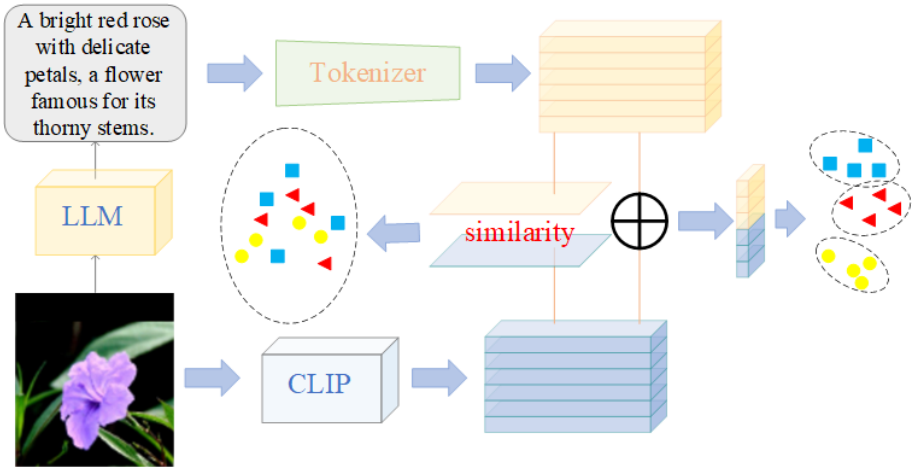


Figure 1: Model framwork

3.1. Overview

We adopt the CLIP framework for few-shot image classification, which consists of a vision encoder $E_I(\cdot)$ and a text encoder $E_T(\cdot)$ trained jointly to align image and text representations. In zero-shot inference, CLIP predicts an image's class by finding the text class name whose embedding is closest to the image embedding. Formally, let x be an input image and y its class label (one of C possible classes). CLIP computes an image feature vector $z = E_I(x)$ and text feature vectors $t_c = E_T(p_c)$ for each class c, where p_c is a text prompt containing the class name c. The image is assigned to the class with the highest cosine similarity:

$$\hat{y} = arg \max_{c\in\{1,\dots,C\}} \langle \cdot,\cdot \rangle,$$

where $\langle \cdot,\cdot \rangle$ denotes inner product. The design of the prompt p_c is critical – CLIP was originally prompted with template like "a photo of a classclassclass" for each class. In few-shot learning, we can optimize prompts p_c using labeled training set to improve accuracy. CoOp introduced learnable continuous prompt vectors $V = [v_1, v_2, \dots, v_M]$ are independent of class c becomes:

DOI: 10.54254/2755-2721/2025.22858

$$p_c^{\text{CoOp}} = [v_1, v_2, \ldots, v_M, w_c],$$

Where w_c is the embedding of the class name token. The prompt p_c^{CoOp} is then passed through the text encoder E_T to produce t_c. The image is classified by comparing $E_I(x)$ with each t_c as usual, and the cross-entropy loss on the labeled set is used to learn V. CoOp's prompt V effectively acts as a task-specific context, but it is static for all images and tuned only for base classes.

CoCoOp improves on this by making V a function of the image. It defines a meta-network $m(\cdot)$ that takes the image feature $z = E_I(x)$ as input and outputs M prompt vectors conditioned on x:

$$p_c^{\text{CoCoOp}} = [u_1, u_2, \ldots, u_M, w_c],$$

which depends on x through $u_i = m(E_I(x))$. This conditional prompt is then encoded to $t_c(x) = E_T(p_c^{\text{CoCoOp}}(x))$. The meta-network $m(\cdot)$ is trained on base classes such that the model's predictions remain accurate for those classes, but m generalizes to produce meaningful prompts for unseen classes as well. CoCoOp's training objective combines base class classification accuracy with regularization to encourage generalizable prompt vectors. In LLM-PromptTuning, we introduce two enhancements to this pipeline: 1) descriptive prompts generated by a Large Language Model for each class, and 2) a multi-head self-attention architecture for the prompt meta-network $m(\cdot)$. An illustration of our pipeline is as follows. For each class c, we query an LLM (such as GPT-style model) with the class name to obtain a natural language description or template T_c. This template might be, for example for class "rose": "A vibrant red rose with delicate petals-a type of flower known for its thorny stem." The idea is that T_c will encode salient attributes of class c in human language. We then tokenize T_c and obtain a sequence of word embeddings $[w_{c,1}, w_{c,2}, \ldots, w_{c,K}] = \text{Tokenizer}(T_c)$ of length K. These serve as a discrete prompt template for class c fixed throughout training and inference.

On the other hand, for each image x, the meta-network $m(\cdot)$ produces a sequence of continuous prompt embeddings $[u_1, \ldots, u_M] = m(z)$ specific to that image. Finally, we concatenate the LLM-derived prompt tokens and the conditional tokens to form the full prompt sequence:

$$p_c^{\text{LLM-PT}}(x) = [w_{c,1}, \ldots, w_{c,K}, u_1, \ldots, u_M],$$

which is fed into the text encoder: $t_c(x) = E_T(p_c^{\text{LLM-PT}}(x))$. The classification is done by computing similarities $\langle E_I(x), ; t_c(x) \rangle$ for all classes c. We train the meta-network $m(\cdot)$ and fine-tune the LLM prompts T_c on the base class training data by minimizing cross-entropy on the correct class. In practice, we keep T_c fixed and only train $m(\cdot)$, since T_c encodes high-level knowledge.

3.2. Descriptive prompt generation via LLM

The descriptive prompt T_c for each class is obtained using a large language model which has learned from web-scale text corpora. We craft a simple prompt for the LLM, for example: "Describe the concept of a '<class name>' in one sentence." For each class name, we feed this into the LLM to generate a sentence or phrase. These generated descriptions often include defining characteristics, common contexts, or fine-grained details of the class. By doing so, we effectively transfer some of the world knowledge from the language model into our vision-language model. The resulting text T_c is not manually written but automatically created, and can be considered a prompt template that augments the class name with informative context. We found that these LLM-generated prompts, which might mention attributes, provide richer supervision to CLIP than a plain class name or a generic template. Formally, let $\mathscr{L}$ be the language model generation function. We define and then

DOI: 10.54254/2755-2721/2025.22858

obtain token embeddings $w_{c,1\ldots K} = \text{Tokenizer}(T_c)$ as described. We do not require any labels or additional data for this step; it's a one-time offline computation. All T_c for base and novel classes are generated before training. Notably, T_c often already contains the class name c or its synonyms, ensuring that the class identity is preserved in the prompt. We keep the length K limited so that the combined prompt length $K + M$ fits within CLIP's text encoder input limit.

3.3. Conditional prompt network with self-attention

We upgrade CoCoOp's meta-network to a Transformer-based architecture to better exploit the image features for prompt generation. The original $m(z)$ was implemented as a few fully connected layers. In LLM-PromptTuning, $m(\cdot)$ is a small Transformer encoder that uses multi-head self-attention to produce the prompt tokens $u_{1..M}$. Specifically, we consider the CLIP image feature z as containing information about the image as a whole. We project z into an initial sequence of M vectors. These serve as an initial prompt embedding which is refined by the Transformer's self-attention layers. The multi-head attention allows the model to attend to different aspects or sub-features of z across heads. In essence, each head can be thought of as focusing on a particular property of the image when generating the prompt vectors. The outputs of all attention heads are concatenated and transformed to produce the final M prompt tokens for that image. We also incorporate residual connections and layer normalization in this Transformer as is standard, which stabilizes training. The self-attention based $m(\cdot)$ is more expressive than a plain MLP, at a modest increase of parameters. By utilizing multiple attention heads, it can capture interactions in the image feature space that single-head or feed-forward layers might miss. This results in more informative conditional prompts u_i. Moreover, this design can be seen as a form of cross-attention if we treat z as keys/values and the prompt queries as learnable parameters-akin to how decoder transformers attend to encoder outputs in seq-to-seq models. Empirically, this helps the prompt better adapt to each image's unique content. We train the Transformer-based meta-network on the few-shot training set of base classes. The training objective is the same as CoCoOp: for each training image x with label y, we form $p_y^{\text{LLM-PT}}(x)$ and ensure that $\langle E_I(x), E_T(p_y^{\text{LLM-PT}}(x))\rangle$ is higher than similarities with other class prompts. We use a cross-entropy loss over the cosine similarities. All CLIP backbone weights remain frozen; only the meta-network are learned, which makes training feasible even with very few examples per class.

4. Experiment

Table 1: The experimental resultes

Dataset	Method	Base (%)	Novel (%)	HMean (%)
Caltech101	CLIP (zero-shot)	96.8	94.2	95.5
	CoOp	97.0	60.5	74.5
	CoCoOp	97.5	94.5	96.0
	LLM-PromptTuning (Ours)	98.0	95.5	96.7
Flowers102	CLIP (zero-shot)	72.0	78.0	74.9
	CoOp	97.0	60.5	74.5
	CoCoOp	94.0	72.5	81.9
	LLM-PromptTuning (Ours)	92.5	76.0	83.4
Food101	CLIP (zero-shot)	90.0	91.8	90.9
	CoOp	88.3	82.0	85.0
	CoCoOp	91.0	91.5	91.2
	LLM-PromptTuning (Ours)	90.5	92.5	91.5

DOI: 10.54254/2755-2721/2025.22858

We evaluate LLM-PromptTuning on three representative few-shot image classification benchmarks: Caltech101, Oxford Flowers102, and Food101. These datasets cover a variety of domains-Caltech101 contains 101 object categories ranging from animals to artifacts; Flowers102 consists of 102 categories of flowers; Food101 includes 101 categories of food dishes. For each dataset, we follow the standard few-shot setup used in CoOp/CoCoOp: we split the classes into base and novel categories (half of the classes as base, half as novel), perform few-shot training on the base classes, and evaluate on both base and novel classes. Specifically, we use 16 shots per base class for training, and report accuracy on test sets of base classes and novel classes separately. The base classes accuracy reflects how well the model learned the training categories, while the novel classes accuracy measures the model's generalization to unseen categories. We also report the harmonic mean (HMean) of base and novel accuracy as an aggregate measure that balances seen and unseen performance. LLM-PromptTuning shows a clear advantage in scenarios with strong domain shift or fine-grained classes, as evidenced by Flowers102, while matching or exceeding state-of-the-art on others like Caltech101 and Food101. The LLM-generated prompts consistently improved novel class recognition, suggesting that the additional semantic cues help the model recognize unseen categories more reliably. Meanwhile, the multi-head attention prompt generator maintained competitive base class accuracy, indicating it can adapt to training classes nearly as well as CoOp's fully learnable prompt.

5. Conclusion

We presented LLM-PromptTuning, a novel approach to few-shot image classification that bridges vision and language by utilizing large language models to guide prompt design. By merging these two types of prompts, our approach creates enriched textual inputs that lead to better alignment with image features.LLM-PromptTuning combines the strengths of pre-trained language models and vision-language contrastive models to significantly improve few-shot image classification. The approach requires minimal additional data and can be easily applied to other vision-language models. We believe this work opens up new possibilities for using generative knowledge from language models to assist visual recognition tasks. Future work may extend this idea by exploring more sophisticated LLM prompting strategies, joint optimization of prompts and meta-network, or applying the technique to zero-shot transfer learning scenarios. We hope our results encourage further exploration at the intersection of large language models and visual few-shot learning, paving the way for models that can learn from both images and rich textual knowledge to achieve robust recognition with limited examples.

References

[1] *He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.*

[2] *Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.*

[3] *Wang Y, Yao Q, Kwok J T, et al. Generalizing from a few examples: A survey on few-shot learning[J]. ACM computing surveys (csur), 2020, 53(3): 1-34.*

[4] *Tsimpoukelli M, Menick J, Cabi S, et al. Multimodal few-shot learning with frozen language models, 2021[J]. URL https://arxiv. org/abs/2106.13884.*

[5] *Gu Y, Han X, Liu Z, et al. Ppt: Pre-trained prompt tuning for few-shot learning[J]. arXiv preprint arXiv:2109.04332, 2021.*

[6] *Wang W, Zheng V W, Yu H, et al. A survey of zero-shot learning: Settings, methods, and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 1-37.*

[7] *Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PmLR, 2021: 8748-8763.*

[8] *Fürst A, Rumetshofer E, Tran V H, et al. CLOOB: Modern Hopfield Networks with InfoLOOB Outperform CLIP. arXiv: Learning[J]. 2021.*

DOI: 10.54254/2755-2721/2025.22858

[9] Li Y, Liang F, Zhao L, et al. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm[J]. arXiv preprint arXiv:2110.05208, 2021.

[10] Lee J, Kim J, Shon H, et al. Uniclip: Unified framework for contrastive language-image pre-training[J]. Advances in Neural Information Processing Systems, 2022, 35: 1008-1019.

[11] Chen Y, Yang G, Wang D, et al. Eliciting knowledge from language models with automatically generated continuous prompts[J]. Expert Systems with Applications, 2024, 239: 122327.