Meta-Prompting for Automating Zero-shot Visual Recognition with LLMs

M. Jehanzeb Mirza[†1,2] **Leonid Karlinsky**[3] **Wei Lin**[4]
Sivan Doveh[5,6] **Jakub Micorek**[1] **Mateusz Kozinski**[1]
Hilde Kuhene[3,7] **Horst Possegger**[1,2]

[1]ICG, TU Graz, Austria. [2]CDL-EML. [3]MIT-IBM Watson AI Lab, USA.
[4]JKU, Austria. [5]IBM Research, Israel. [6]Weizmann Institute of Science, Israel.
[7]University of Bonn, Germany.

Code & Data: https://github.com/jmiemirza/Meta-Prompting

Abstract. Prompt ensembling of Large Language Model (LLM) generated category-specific prompts has emerged as an effective method to enhance zero-shot recognition ability of Vision-Language Models (VLMs). To obtain these category-specific prompts, the present methods rely on hand-crafting the prompts to the LLMs for generating VLM prompts for the downstream tasks. However, this requires manually composing these task-specific prompts and still, they might not cover the diverse set of visual concepts and task-specific styles associated with the categories of interest. To effectively take humans out of the loop and completely automate the prompt generation process for zero-shot recognition, we propose **M**eta-**P**rompting for **V**isual **R**ecognition (MPVR). Taking as input only minimal information about the target task, in the form of its short natural language description, and a list of associated class labels, MPVR automatically produces a diverse set of category-specific prompts resulting in a strong zero-shot classifier. MPVR generalizes effectively across various popular zero-shot image recognition benchmarks belonging to widely different domains when tested with multiple LLMs and VLMs. For example, MPVR obtains a zero-shot recognition improvement over CLIP by up to 19.8% and 18.2% (5.0% and 4.5% on average over 20 datasets) leveraging GPT and Mixtral LLMs, respectively.

1 Introduction

Dual encoder Vision-Language Models (VLMs) [35, 47] attain unprecedented performance in zero-shot image classification. They comprise a text encoder and an image encoder trained to map text and images to a shared embedding space. Zero-shot classification with dual encoder VLMs consists in evaluating the cosine similarity between the embedding of a test image and the embeddings of texts representing candidate classes.

The composition of the class-representing text has a significant impact on the accuracy of zero-shot classification. Already the authors of CLIP [35], the first large-scale vision-language model, highlighted its importance and reported

[†]Correspondence: mirza@tugraz.at

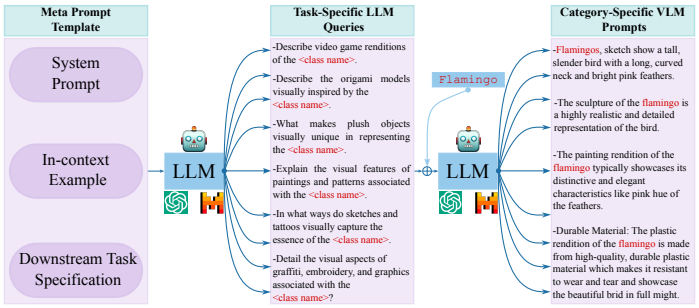

Fig. 1: Our MPVR utilizes a Meta Prompt, comprising a system prompt (instruction), in-context example demonstrations (fixed throughout), and metadata (name and description) for a downstream task of interest. The Meta Prompt instructs an LLM to generate diverse task-specific LLM queries, which are used to obtain category-specific VLM prompts (visual text descriptions) by again querying the LLM after specifying the `<class name>`. These category-specific VLM prompts are then ensembled into a zero-shot classifier for recognizing the downstream task categories.

that embedding class names in a *prompt* of the form `'A photo of a <class name>'` resulted in considerable performance growth over using raw class names. Moreover, specializing the prompt to the data set by adding high-level concepts, for example, embedding the class name in the sentence `'A photo of a <class name>, a type of flower'` for fine-grained flower recognition, brought further improvement. Finally, a substantial performance boost was achieved by ensembling multiple different prompts, tailored towards the downstream task (dataset). Since ensembling a larger number of dataset- and class-specific prompts is beneficial, and manually designing a large number of class-specific prompts is prohibitively time-consuming, several authors delegated prompt generation to a Large Language Model (LLM) [28,34,37]. These approaches consist in asking an LLM to generate class descriptions [34], or class attributes [28], and mix them with manually defined prompt templates [37]. They enable generating large sets of prompts adapted to the downstream task, which would be prohibitively time-consuming when performed manually. However, they still require hand-crafting prompts to the LLM [34] or dataset-specific LLM prompt templates [37], or rely on the assumption that class attributes are discriminative [28, 37]. In other words, they do not eliminate the manual effort completely, but shift some of it from manually designing prompts for the VLMs (as in [35]) to manually designing LLM prompts. This raises the following question: Does the manual design of the LLM prompts bias the resulting VLM prompts, possibly affecting performance? In this work, we answer this question affirmatively: we minimize manual interventions in the prompt generation process and show that this significantly boosts zero-shot recognition accuracy. [1]

[1] To avoid confusion between the 'prompts' used to query the LLMs and the 'prompts' used to compute the text embedding by the VLMs, in the remaining part of this manuscript, we call the first one 'LLM query' and the second one 'VLM prompt'.

The gist of our approach lies in automating the prompt generation process. To that end, we draw inspiration from methods for reducing the prompt engineering effort in natural language processing [13,40] and propose to meta-prompt the LLM to produce LLM query templates tailored to the downstream task. We call our method Meta-Prompting for Visual Recognition (MPVR). Its overview is presented in Figure 1. MPVR comprises a 'system prompt' that describes the meta-prompting task for the LLM, a description of the downstream task, and an in-context example. The in-context example contains a description (metadata) of another task and its corresponding 'LLM queries', and serves to bootstrap the LLM with examples of expected results. They are kept the same across different downstream tasks and for all our experiments. MPVR extracts the LLM's knowledge of the visual world gradually, in two steps. The first query to the LLM contains the system prompt, in-context example, and the downstream task (dataset) description, and produces a diverse set of LLM *query templates*, containing a `<class name>` placeholder. These templates are infused (by the LLM) with information on visual styles specific to the downstream task of interest, but they are still category-agnostic. In the second step, for each class, we populate its label into all the task-specific LLM query templates generated in the first step and use them to query the LLM to generate (category-specific) VLM prompts describing the category in visually diverse ways and also containing task-specific visual styles infused by the LLM in the first step. We use the resulting VLM prompts to create an ensemble of zero-shot classifiers. In section 4, we show that MPVR's two-step process results in state-of-the-art zero-shot classification.

Our meta-prompting strategy does not take any parameters specific to the dataset, other than the dataset description, which can be easily obtained through public APIs or from its webpage. Yet, we show that prompts generated by MPVR cover diverse visual concepts and styles specific to the downstream task. As a result, MPVR yields significant performance gains on a range of zero-shot benchmarks. Our contributions can be summarized as follows:

- We propose MPVR: a general, automated framework requiring minimal human involvement for tapping into the visual world knowledge of LLMs through meta-prompting for zero-shot classification.
- MPVR generalizes beyond closed models (like GPT [2]). We are the first to show that category-specific descriptions generated from open-source models (like Mixtral [16]) can also enhance the zero-shot recognition abilities of state-of-the-art VLMs.
- We open-source a dataset of $\sim$ 2.5M unique class descriptions harnessed from GPT and Mixtral with our meta-prompting framework. This is the first large-scale dataset encompassing the breadth of LLM knowledge of the visual world.

2 Related Work

We first provide an overview of the zero-shot vision-language foundation models, then touch upon approaches that propose to improve these models by requiring

visual data (relying on additional training), and later discuss different methods that follow our line of work, *i.e.,* improving zero-shot models in a training-free manner by generating textual data through LLMs and finally provide an overview of the prompt engineering literature.

Large Scale Vision-Language Foundation Models: VLMs have shown impressive performance for many vision-language understanding tasks, *e.g.,* zero-shot recognition, visual question-answering (VQA), image captioning, *etc*. The present-day VLMs can be placed in two distinct groups in a broader categorization. One group of methods relies on dual-encoders (vision and text encoder) and usually trains the encoders with a contrastive objective by using a large corpus of paired image-text data scraped from the web. The most common among these methods are CLIP [35], ALIGN [15], OpenCLIP [38], and the very recent MetaCLIP [47]. The zero-shot classification is performed by measuring the similarity between the image embeddings and encoded text features, usually obtained by using the default template `'a photo of a <class name>'`. The other group of methods aligns the visual modality with a frozen LLM. BLIP-2 [22] bridges the modality gap between a pre-trained visual encoder and an LLM by using a querying transformer. Instruct-BLIP [8] proposes to improve [22] by employing instruction tuning. MiniGPT [53] aligns a vision encoder with a frozen LLM (Vicuna [6]) by only using a (trainable) linear projection layer between the two. MiniGPT-V2 [3] replaces the LLM with Llama-2 [41] and also proposes to unfreeze it during the training/finetuning phases. Llava [26] also aligns an LLM with a pre-trained visual encoder and also proposes Visual Instruction Tuning, by carefully curating instruction-response pairs, to enhance the performance. Furthermore, the performance of LLaVA is also enhanced with better data curation [24] and slight architectural changes [25]. In our work, we focus on the contrastively pre-trained zero-shot models widely used for object recognition (*e.g.,* CLIP [35]), and improve the recognition abilities of these models by generating the text embeddings from a variety of descriptions (instead of the default templates) harnessed through our proposed meta-prompting technique. Furthermore, we show that MPVR-enhanced CLIP [35] outperforms even the leading LLM-decoder-based methods (*e.g.,* [25]) in visual recognition tasks.

Training-based Approaches for Improving VLMs: Different approaches propose to improve the zero-shot recognition performance of the contrastively pre-trained models through parameter-efficient fine-tuning. CoOp [52] proposed to learn randomly initialized text prompts in a few-shot manner. CoCoOp [51] further conditions the learnable text prompts on the visual inputs to enhance the performance. Maple [18] proposes to learn both the visual and text prompts in conjunction. Contrary to relying on few-shot labeled visual samples, UPL [14] proposes to learn the text prompts on unlabeled image data and LaFTer [31] learns visual prompts by leveraging the cross-modal transfer capabilities of CLIP. While these approaches propose to adapt the VLM on image data, MAXI [23] proposes to fine-tune CLIP in an unsupervised manner for video inputs. In contrast to the methods proposed to improve the zero-shot recognition abilities of CLIP, our work does not rely on visual inputs and gradient-based updates of network pa-

rameters. Instead, it improves the zero-shot recognition performance by harnessing fine-grained textual concepts generated through our MPVR, thus supporting the capability to scale zero-shot recognition performance improvements to visual domains where *no visual data* might be available for training.

Zero-shot Recognition with Additional Textual Data from LLMs: It was initially highlighted in CLIP [35] that generating the text embeddings through an ensemble of (dataset specific) hand-crafted prompts[2] improved the zero-shot recognition performance on the downstream datasets, hinting towards the sensitivity of CLIP's text encoder towards fine-grained textual concepts. Following up on this idea, DCLIP [28] enhances visual recognition by generating category-specific descriptors through an LLM (GPT-3 [2]). On the other hand, CUPL [34] proposes to obtain the category-level text embeddings from the prompts generated with the dataset-specific hand-crafted queries fed to the LLM. Waffle [37] hints towards the potential *bag-of-words* behavior of the CLIP text encoder and performs zero-shot classification by adding random descriptors to broad concepts and DCLIP-generated attributes. Our work also takes inspiration from the prompt ensembling in [28, 34, 35, 37] and performs zero-shot classification by generating category-level prompts through an LLM. However, contrary to these approaches, MPVR proposes a more general prompting framework to alleviate the human effort spent for handcrafting the LLM queries (CUPL [34]), dataset-specific concepts (Waffle [37]), or reduce reliance on individually recognizable visual attributes (DCLIP [28]). By effectively incorporating general downstream task information (description) into the first phase of MPVR (*i.e.,* meta-prompting), we automatically produce task-tailored LLM query templates ready to be populated by task categories and used to query an LLM for a diverse spectrum of category-level VLM prompts comprising an enhanced set of visual details for recognizing those categories. The performance gains by using MPVR with both closed and open-source LLMs (GPT [2] and Mixtral [16]) on 20 different datasets when compared to relevant baselines highlight the generalization capabilities and benefits of our approach.

Prompt Engineering: Manually manipulating the text inputs (prompts) to the LLMs for enhancing performance for various natural language processing (NLP) tasks has been an active field of research, which is formalized as prompt engineering. In this context, providing demonstrations to the LLM for solving related downstream tasks has been referred to in the NLP literature as in-context learning (ICL) [4, 29, 44, 49]. Orthogonal to the idea of in-context learning, some approaches rely on breaking down a complex task into a series of events. To this end, Chain-of-Thought (CoT) [45] achieved impressive performance gains by prompting the model to perform intermediate reasoning steps. Other approaches following this line of work include [19, 48]. Our MPVR also employs ICL and manipulates the input prompts to the LLMs, but effectively alleviates the need for human involvement for this manipulation by probing an LLM for more diverse concepts (LLM query templates – incorporating general information about the task), which are then populated with specific task categories and fed again to

[2] https://github.com/openai/CLIP/blob/main/data/prompts.md

the LLM for generating VLM prompts - both task- and category-specific text descriptions of visual concepts. To the best of our knowledge, such a two-stage (meta-) prompting strategy for tapping into the visual world knowledge of LLMs does not exist in literature.

3 MPVR: Meta-Prompting for Visual Recognition

Zero-shot classification with a dual encoder VLM consists in projecting a test image and each candidate class to the common embedding space, and evaluating the cosine similarity between the embeddings. The image embedding is produced by the VLM's vision encoder ϕ. The embedding of a class is obtained by passing a textual description of the class, called a VLM prompt, through the VLM's text encoder ψ. The simplest technique of constructing a VLM prompt is to complete a prompt template, for example, '`A photo of a <class name>`', with class label [35]. The authors of CLIP [35], the first large-scale VLM, highlighted that prompt composition is vital to the performance of the zero-shot classifier. To boost the performance, they proposed VLM prompt ensembling, which represents the class as a mean embedding of multiple diverse prompts. To formalize this approach, we denote the test image by x, the set of candidate classes by C, and the set of prompt templates by P. By $p(c)$ we denote a prompt obtained by completing template $p \in P$ with the label of class $c \in C$. We define the zero-shot likelihood of class $\hat{c}$ as

$$l_{\hat{c}}(x) = \frac{e^{\cos(\psi_{\hat{c}}, \phi(x))/\tau}}{\sum_{c \in C} e^{\cos(\psi_c, \phi(x))/\tau}}, \quad \text{where} \quad \psi_c = \frac{1}{|P|} \sum_{p \in P} \psi(p(c)), \tag{1}$$

and τ denotes the temperature constant. This approach forms the point of departure for our method.

Ensembling a larger number of class-specific VLM prompts improves the performance of the zero-shot classifier, but generating these prompts manually would be prohibitively time-consuming. Several methods [28, 30, 34, 37] address this problem by generating the VLM prompts with a large language model (LLM), for example GPT [2]. They enhance the performance of the zero-shot classifiers, but still require manual construction of the LLM queries, which scales poorly: A prohibitively large human effort might be needed to creatively design prompts that cover the diverse ways the visual aspects of a certain class can be described in text. Moreover, manually specified queries can be influenced by the subjective bias of the person who composes them, which could affect zero-shot recognition performance.

To improve the scaling of VLM prompt generation and eliminate subjectivity from the process, we design Meta Prompting for Visual Recognition (MPVR), an approach to VLM prompt generation that reduces human input to the necessary minimum. MPVR taps into the visual world knowledge possessed by the VLM and extracts it in two steps. In the first step, MPVR meta-prompts the LLM with generic instructions and coarse information about the downstream task to generate diverse task-specific LLM query templates. These LLM query templates encode elements of the LLM's knowledge about the visual styles characteristic

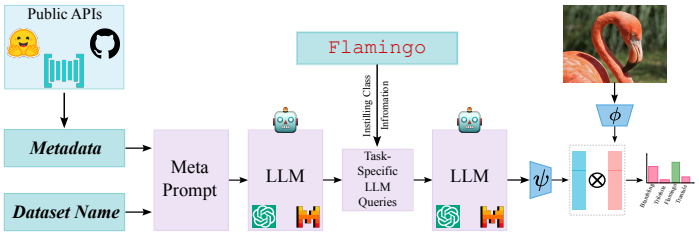

Fig. 2: MPVR framework. In the first stage, a meta-prompt comprising of a *system prompt*, *in-context examples*, and metadata consisting of *downstream task specification* is queried to the LLM instructing it to generate multiple diverse task-specific LLM queries, which are populated with the category of interest and again queried to the LLM to obtain the category-level prompts for assembling a zero-shot classifier.

of the downstream task but are still class-agnostic. In the second step, the LLM query templates are populated with names of candidate classes and fed to the LLM to obtain VLM prompts. The resulting VLM prompts are both task- and class-specific. Each prompt carries LLM's diverse visual knowledge about the possible appearance of objects representing the class in the style defined by the downstream task.

For ease of assimilation, we divide our MPVR into two distinct stages and provide an overview in Figure 2. In Section 3.1, we describe how to effectively meta-prompt LLMs to generate diverse, task-specific LLM query templates (stage 1). Later in Section 3.2 we describe how to use these task-specific LLM query templates to obtain category-specific VLM prompts (stage 2).

3.1 Meta-Prompting a Large Language Model

Aligning with the true motivation of our MPVR, the goal of meta-prompting is to extract the abundant visual world knowledge possessed by the LLMs by querying it to generate multiple diverse LLM query templates with minimal human intervention. To that end, we compose a meta-prompt of three parts: the *system prompt*, an *in-context example*, and the *downstream task specification*. We illustrate the meta-prompt in Figure 3.

System prompt is a generic set of instructions that describe the elements of the meta-prompt and specify the expected output of the LLM. It instructs the LLM to generate a variety of query templates for the downstream dataset and conveniently format them to be employed in a `Python` script.

In-context example serves to bootstrap the LLM to the type of output that is expected. It comprises a description of an example downstream task and a list of the corresponding LLM query templates. Since we expect the output from the LLM to be suitable for use in a `Python` script thus, it contains the prompts listed as `Python code` (*c.f.,* Figure 3, middle left & right).

Downstream task specification is the only part of the meta-prompt that is specific to the downstream task. It is scraped from a public API or the webpage of the

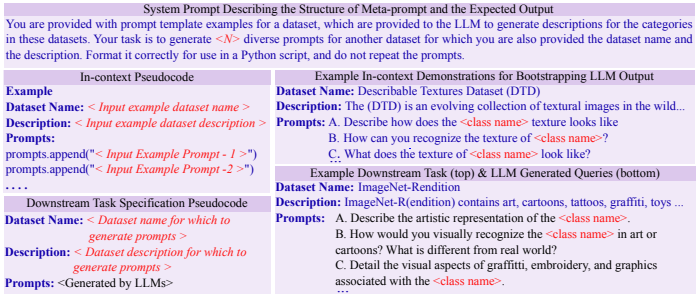

Fig. 3: Our meta-prompt comprises 3 parts: A *system prompt* provides an overview of what is included in the overall prompt and what is expected from the LLM as a response (top). An *in-context example* consisting of metadata, dataset name, and hand-crafted prompts for the dataset (middle left). The *downstream task metadata* for which a diverse set of prompts are requested from the LLM (bottom left). For completeness, we also provide the in-context demonstrations (middle right) we use throughout, and the diverse LLM-generated queries for the example ImageNet-R dataset (bottom right).

dataset associated with the task and contains a general description of the task data (*c.f.,* Figure 3, bottom left & right). This coarse information about the downstream task of interest is critical for the LLM to generate task-specific LLM queries, which are employed in stage 2 of MPVR.

Note that the *system prompt* and the *in-context example* demonstrations are generic and are kept fixed across different tasks in all of our experiments. The *downstream task specification* is the only part of the meta-prompt that is specific to the downstream task. Our experiments highlight that all the individual parts of the meta-prompt are extremely vital for our MPVR to obtain effective category-specific VLM prompts and are extensively ablated in Table 5.

The three elements of the meta-prompt are embedded in the template presented in Figure 1 (left). The resulting meta-prompt is then fed to the LLM (GPT [2] or Mixtral [16]) to generate N diverse LLM query templates that are infused with the LLM's knowledge of visual styles expected in the dataset, but are still class-agnostic. Instead of the downstream `<class name>` of interest, they contain a generic `<class name>` placeholder. To obtain category-specific VLM prompts, we transition to stage 2 of our MPVR explained next.

3.2 Category-Specific VLM prompts

The LLM response to the meta-prompt in stage 1 is a diverse set of LLM query templates, which contain task-specific knowledge about the downstream task of interest, but are still generic. To instill the category information, for obtaining the category-specific VLM prompts, we replace the generic `<class name>` placeholders in the LLM query templates with the actual class of interest. These diverse category-specific queries constitute our second call to the LLM, which generates category-specific VLM prompts. They carry the LLM's knowledge of the appearance of objects of the queried classes in the context of the downstream task and are ready to be plugged into Eq. (1). We repeat this procedure for each class from 20 different datasets (used for evaluations) with both the GPT [2] and

Mixtral [16] LLMs and obtain a huge corpus of $\sim$ 2.5M VLM prompts. In section 4, we show that the ensemble of these VLM prompts results in a zero-shot classifier that outperforms previous methods by a significant margin.

The VLM prompts can be thought of as visually diverse descriptions of the queried classes in the context of the downstream tasks, and their corpus represents a chunk of the LLM's knowledge about our visual world. This diversity stems from our proposed two-stage approach[3]. The first stage can already provide diverse LLM query templates, which resemble the dataset-specific templates for prompt ensembling [2] (but more diverse and automatically generated with our MPVR). Interestingly, even by generating the ensemble of zero-shot classifiers by populating these generic query templates from stage 1 with category information, we can already achieve enhanced zero-shot recognition, as reported in an ablation in Table 6. To conclude, after the second call to the LLM, the VLM prompts constitute fine-grained details about the specific category, reflecting the true diversity of the visual LLM knowledge and resulting in a huge category-specific text corpus, already incorporated in our codebase attached as the supplementary material and will be open-sourced upon acceptance.

4 Experimental Evaluation

In this section, we first briefly describe the datasets and the baselines we use to evaluate and compare our MPVR, then explain our implementation details and finally provide a detailed discussion of the results.

4.1 Evaluation Settings

Datasets: We extensively evaluate our MPVR on 20 object recognition datasets belonging to widely different domains. These domains can be narrowed down to datasets containing commonly occurring natural categories: ImageNet [9], ImageNet-V2 [36], CIFAR-10/100 [21], Caltech-101 [10]. Fine-grained classification datasets containing different task-specific images: Flowers [32], Standford Cars [20], CUBS-200 [42], Oxford Pets [33], Describable Textures dataset (DTD) [7], Food-101 [1], FGVC-Aircraft [27]. Datasets used for scene classification: Places365 [50] and SUN397 [46], action recognition datasets: UCF101 [39] and Kinetics400 [17]. Datasets consisting of out-of-distribution images: ImageNet-(R)endition [12] and ImageNet-(S)ketch [43] and also datasets which contain images taken from a satellite or an aerial view: EuroSAT [11] and RESISC45 [5].

Baselines: We compare to the following baselines and state-of-the-art methods:

- **CLIP** [35] denotes the zero-shot classification scores obtained by using the simple '`{a photo of a <class name>}`' template (S-TEMP) and dataset-specific templates (DS-TEMP[2]).
- **CUPL** [34] proposes to generate category-level descriptions from an LLM with hand-crafted prompts for each dataset.

[3] We also experimented with generating category-specific VLM prompts in a single step with meta-prompting, but it performs worse than our 2-stage framework. These results are provided in the ablations Table 6.

	ImageNet	ImageNetv2	C10	C100	Caltech101	Flowers	Stanford Cars	Cubs	Pets	DTD
CLIP (S-TEMP)	61.9	54.8	88.3	64.4	91.4	64.0	**60.2**	51.6	85.0	40.2
CLIP (DS-TEMP)	63.3	56.0	89.2	65.1	89.9	66.7	60.0	53.0	87.4	42.4
CUPL	64.3	56.9	89.0	65.3	92.1	68.8	60.0	51.9	87.2	48.9
DCLIP	63.1	55.8	86.7	64.2	92.5	64.6	57.9	52.6	83.5	44.3
Waffle	63.4	56.3	89.4	65.2	90.8	67.8	59.9	52.8	87.7	40.4
Waffle+Con	63.4	56.3	89.4	65.2	89.7	65.2	59.5	52.1	86.8	41.7
Waffle+Con+GPT	63.4	56.3	89.4	65.2	91.9	68.2	59.6	52.6	87.9	41.8
MPVR (MIXTRAL)	63.8	56.5	89.5	65.5	92.8	**75.2**	58.3	55.5	88.0	50.2
MPVR (GPT)	**65.0**	**57.3**	**89.9**	**66.3**	**92.9**	73.9	59.5	**55.9**	**88.1**	**50.8**

	Food101	Aircraft	Places365	SUN397	UCF101	K400	IN-R	IN-S	EuroSAT	Resisc45
CLIP (S-TEMP)	77.6	18.1	39.4	62.1	60.4	39.7	66.3	41.1	35.9	54.1
CLIP (DS-TEMP)	79.2	19.5	40.0	63.0	62.4	42.1	69.3	42.7	45.8	57.8
CUPL	81.0	20.4	_	66.5	65.2	41.7	_	_	_	61.9
DCLIP	79.7	19.8	40.9	63.1	62.6	39.1	66.0	42.3	48.9	56.9
Waffle	**81.6**	20.1	41.1	63.3	62.7	40.4	68.8	43.4	42.7	61.4
Waffle+Con	81.1	19.0	39.3	60.7	62.2	39.1	68.1	42.5	44.8	58.6
Waffle+Con+GPT	81.2	19.8	41.5	64.0	63.4	40.4	68.5	43.7	47.0	62.0
MPVR (MIXTRAL)	81.3	**22.4**	42.1	66.5	66.0	42.2	**70.2**	43.6	54.0	**64.6**
MPVR (GPT)	81.0	21.5	**42.2**	**67.0**	**67.9**	**43.9**	**70.2**	**44.2**	**55.6**	64.0

Table 1: Top-1 accuracy (%) for 20 datasets obtained by employing the ViT-B/32 backbone from OpenAI CLIP [35]. *S-TEMP* refer to the results obtained by using the default template (`a photo of a <class name>`), while *DS-TEMP* refer to the results obtained by using the ensemble of dataset specific prompts. An empty placeholder for CUPL [34] indicates that the respective baseline did not provide the handcrafted prompts for the dataset. For Waffle [37], mean results from 7 random runs are reported, following the original publication.

- **DCLIP** [28] proposes to obtain a zero-shot classifier with category-specific descriptors (from an LLM) consisting of usual visual attributes.
- **Waffle** [37] employs hand-crafted task-specific broad concepts and adds random descriptors to the prompts for zero-shot classification. Following their evaluation setting, we compare with different variants: (i) Waffle (prompt + random descriptors), (ii) WaffleCon (Waffle + high-level concepts), and (iii) WaffleConGPT (WaffleCon + DCLIP descriptors).

Implementation Details: To report the results for each dataset we use the test splits provided by [52] and further build upon their framework for all our evaluations on datasets that are not present in their framework. All the baselines are also implemented in the same framework. To generate the diverse set of task-specific LLM queries for our MPVR in the first stage, we use the public web API of ChatGPT[4] and the Hugging Face API for Mixtral-7B (8x)[5]. To obtain the category-level VLM prompts after querying an LLM in the second stage of MPVR, we use GPT-3.5 [2] and the open source weights of Mixtral-7B (8x) [16], accessed through Hugging Face. In the first stage, we instruct the LLM to generate 30 diverse task-specific queries for each dataset, and to obtain the category-level VLM prompts, we append the category of interest and prompt the LLM to generate 10 prompts for each LLM query respectively, where we limit each generated prompt by 50 tokens. The in-context dataset is (arbitrarily) chosen to be DTD [7] for all experiments, however, to avoid information contamination, we switch the in-context dataset to EuroSat [11] when DTD is the

[4] https://chat.openai.com/
[5] https://huggingface.co/chat/

	OpenAI CLIP		MetaCLIP 400m		
	B/16	L/14	B/32	B/16	L/14
S-TEMP	61.9	69.2	62.4	65.9	71.0
DS-TEMP	63.8	71.2	64.0	67.3	72.8
D-CLIP	64.4	70.7	62.8	66.4	72.2
Waffle	64.0	70.7	62.8	66.5	72.4
Waffle+Con	62.7	69.1	61.7	65.7	71.7
Waffle+Con+GPT	64.6	71.0	63.2	66.9	72.7
MPVR (Mixtral)	66.4	72.5	65.6	**68.7**	73.9
MPVR (GPT)	**66.7**	**73.4**	**65.8**	**68.7**	**74.3**

Table 2: Mean top-1 accuracy (%) over 20 datasets for different backbones from OpenAI [35] and MetaCLIP-400m [47].

target dataset. We ablate the choice of DTD for in-context example and provide the complete meta prompts in the supplementary. Unless otherwise specified, we obtain the zero-shot classifier as the mean of the class embeddings obtained from the category-specific VLM prompts (from stage 2 of MPVR) using Eq. (1).

4.2 Results

We test our MPVR extensively on 20 diverse datasets and report the results (with ViT-B/32 from CLIP [35]) in Table 1. We consistently outperform the CLIP zero-shot baseline while using the category-level prompts generated both from GPT and Mixtral. While comparing to the CLIP baseline, using the default template, on some datasets like EuroSAT, the improvement is up to 19.8% and 18.2%, and on average our MPVR improves upon CLIP by 5.0% and 4.5% while averaging the results on 20 datasets, with GPT and Mixtral LLMs respectively. Similarly, while compared to the more expressive CLIP zero-shot baseline, which uses the hand-crafted dataset-specific templates[2], we still observe considerable average gains of 3.1% and 2.7% with the two LLMs.

Our MPVR also shows strong gains when compared to CUPL [34], which obtains category-level prompts by hand-crafting the LLM queries for each downstream task of interest. Our MPVR not only alleviates this extensive human effort spent while generating the category-level prompts (as in CUPL [34]) but also out-performs CUPL on most of the datasets we compare to. For example, obtaining up to 5.1% and 6.3% performance gains on Flowers-102 [32] dataset with GPT and Mixtral LLMs.

Furthermore, we also observe that while comparing with the baselines which do not generate (descriptive) VLM prompts but rely on other cues like category-level (attribute) descriptors, our MPVR also performs favorably. For example, we outperform DCLIP [28] on all the 20 datasets with performance gains up to 5.3% and 3.3% on UCF-101 with GPT and Mixtral. These results indicate that the generic attributes generated for a category by DCLIP for classification might not capture fine-grained task-specific details required to enhance the classification of categories in domain-specific benchmarks (*e.g.,* action recognition in UCF-101). Finally, from Table 1 we also observe that our MPVR (on average) also outperforms all the variants proposed by Waffle [37], which also highlights that the CLIP text encoder responds favorably to semantically rich text descriptions (prompts), instead of randomly generated descriptors as in Waffle [37]. In

		GPT	Mixtral	Temp	GPT+Temp	Mixtral+Temp	GPT+Mixtral	GPT+Mixtral+Temp
Embedding Average	ViT-B/32	62.9	62.4	59.7	57.0	56.1	63.0	57.7
	ViT-B/16	66.7	66.4	63.8	60.5	59.6	67.0	61.5
	ViT-L/14	73.4	72.5	71.2	68.6	67.3	73.4	69.2
Softmax Average	ViT-B/32	_	_	59.8	62.8	62.3	62.4	62.4
	ViT-B/16	_	_	63.8	66.7	66.4	66.4	66.3
	ViT-L/14	_	_	71.1	73.3	72.4	72.6	72.6

Table 3: Comparison of mean top-1 accuracy (%) for MPVR over 20 datasets while constructing the zero-shot classifier by ensembling with the mean of the embeddings from different text sources (top) and mean of softmax (bottom). For GPT and Mixtral, we only report the results with the mean of the embeddings, since ensembling the softmax of individual descriptions is prohibitively expensive (also noted in [35]). For datasets with fewer classes, we performed softmax ensembling but did not find any major deviation in results. These results are provided in the supplementary.

	EuroSAT	DTD	Caltech	CIFAR-100	Resisc	Mean
CLIP (ViT-B/32)	35.9	40.2	91.4	64.4	54.1	57.2
LLAVA-1.6 (7B)	41.3	16.2	33.0	25.7	33.8	30.0
MPVR (ViT-B/32)	55.6	50.8	92.9	66.3	64.0	65.5

Table 4: Comparison of top-1 accuracy (%) with LLAVA-1.6-Vicuna7b model [24].

summary, our MPVR demonstrates better performance across the board, outperforming all baselines on 18 out of 20 datasets. On the Food-101 [1] dataset, our MPVR comes in second, trailing by a narrow margin of 0.3%. Similarly, on Standford Cars [20], our results indicate that even the dataset-specific prompt ensembling proposed by CLIP fails to enhance performance, underscoring the unique challenges posed by this particular dataset.

To test the generalization ability of our MPVR beyond different LLMs, we also evaluate it with different backbones from CLIP [35] and also employ MetaCLIP [47], which is trained with a different training recipe than CLIP. These results are listed in Table 2. We observe that even while testing with more expressive backbones, like MetaCLIP ViT-L/14, our visually diverse text descriptions (prompts) help to improve the zero-shot accuracy from 71.0% $\rightarrow$ 74.3% (for GPT descriptions) while averaging over the 20 datasets. Due to space limitations, we defer the individual dataset results for these backbones to the supplementary.

4.3 Ablations

Here, we study the significance of different components that constitute our MPVR. Specifically, we first examine the effect of combining multiple text sources, and then motivate our choice of using dual encoder models like CLIP [35] instead of multi-modal language models (MMLMs) by evaluating them for image classification. Later we extensively ablate our prompting strategy and finally conclude with ablations on robustness of the obtained results and scaling analysis.

Ensembling Text Sources. From Tables 1 & 2 we gather that in addition to the enhanced zero-shot classification with GPT and Mixtral generated VLM prompts with our MPVR, the dataset-specific templates[2] from CLIP can also show improvement in results, in comparison to only using the default templates. To evaluate the combined performance of these text sources, we ensemble the 3

dataset name	dataset metadata	in-context (prompts)	class names	Top-1
✗	✓	✓	✗	46.7
✓	✗	✓	✗	42.0
✓	✓	✗	✗	
✓	✓	✓	✓	53.5
✓	✓	✓	✗	55.6

Table 5: Top-1 accuracy (%) for EuroSAT [11] with GPT as LLM and the ViT-B/32 backbone [35] while ablating the different parts of our Meta Prompt. The last row represents the results obtained by our MPVR.

	CLIP (S-TEMP)	CLIP (DS-TEMP)	Prompts-Only	1-Step	MPVR
EuroSAT	35.9	45.8	47.2	51.2	55.6

Table 6: Comparison of top-1 accuracy (%) from the zero-shot classifier obtained with the prompts generated in the first stage and generating category-level descriptions directly from stage-1 of MPVR.

different sources and provide the results in Table 3. We observe that when the category-specific VLM prompts and templates are ensembled over the embedding space, the resulting classifier is weaker than the classifier obtained from only the LLM-generated VLM prompts. However, the mean of the embeddings from both GPT and Mixtral prompts performs the best. These results hint that the prompts from both the LLMs are clustered closely in the CLIP latent space suggesting that these sources describe the categories of interest in a similar (more detailed) way, yet differently from the 'more mechanical' CLIP dataset-specific prompts that do not provide much detail. We also test ensembling the probability spaces from both sources and find that the degradation in performance as a consequence of mixing the descriptions and templates is alleviated.

MMLMs for Zero-shot Classification. Recently, multi-modal language models such as LLaVA [24, 26] have emerged as the preferred choice for various vision-language tasks. Here, we extended their evaluation to zero-shot classification, and the findings are summarized in Table 4. Notably, our results indicate that, for the specific task of object recognition, CLIP [35] outperforms LLaVA by a substantial margin, reinforcing our decision to employ CLIP for the discriminative task, which is the focus of our study. We ablate and detail the sensitivity of MMLMs to different prompting strategies in the supplementary, here we report only its best prompting strategy result.

Meta Prompt. In Table 5 we ablate different components of our meta-prompt (outlined in Figure 3) and report the results on the EuroSAT dataset. We see that all the major components have a strong effect on the downstream performance. For example, if we do not populate the meta-prompt with the in-context demonstrations of example LLM queries for a dataset, the LLM fails to generate the task-specific queries from the first stage. Similarly, removing the metadata (description of datasets) from the in-context example and the resulting dataset of interest also results in a huge performance drop 55.6% $\rightarrow$ 42.0%. We also noticed that interestingly, providing the category names for the datasets in the meta prompt (for stage 1) as extra information did not improve the results, potentially hinting that LLM prefers more simple and succinct instructions.

	accuracy(%)	std
ViT-B/32	62.8	±0.05
ViT-B/16	66.7	±0.04
ViT-L/14	73.3	±0.03

Table 7: Top-1 mean accuracy (%) for CLIP and standard deviation for 10 random runs, for all datasets.

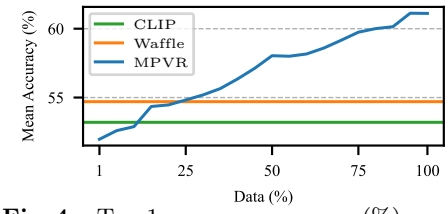

Fig. 4: Top-1 mean accuracy (%) over DTD, EuroSat, Flowers, Resisc45, subsampling the VLM prompts sets.

Altering Meta Prompting Stages. In Table 6 we report the results by altering our meta-prompting strategy in two distinct ways: By generating the category-level VLM prompts directly in one step, by incorporating the class name already in stage 1 of our MPVR, and populating the `<class names>` in the generated task-specific LLM queries from stage 1 (which resembles the prompt ensembling performed by CLIP [35]). The results indicate that our 2-stage approach performs better than altering it to a single stage, and even our generated prompts from stage 1 can offer a more robust zero-shot classifier than templates ensembling[2], highlighting the visual diversity of our generated task-specific queries, which later effectively translates to the VLM prompts as well.

Results Robustness and Scaling Analysis: In Table 7 we study the robustness of MPVR results by reporting the mean and variance with randomly sampling MPVR-generated VLM prompts 10 times for all 20 datasets. We observe that the variances are negligible w.r.t. the obtained gains (in Table 1). In Figure 4 we show the scaling potential by sampling more VLM category- and task-specific prompts. The results highlight that sampling an increasing number of generated VLM prompts significantly boosts performance showing promising scaling potential.

5 Conclusion

We have presented meta-prompting for enhancing zero-shot visual recognition with LLMs, which essentially alleviates any human involvement in VLM prompt design for new tasks. Our MPVR generates task-specific category-level VLM prompts by only requiring minimal information about the downstream task of interest. MPVR first queries the LLM to generate different high-level queries letting it discover the diverse ways of querying itself to generate visually diverse category-level prompts. These prompts are ensembled to construct a robust zero-shot classifier, that achieves enhanced zero-shot classification on a diverse set of 20 datasets belonging to widely different domains. Furthermore, we also open-source the 2.5M category-level text descriptions dataset, harnessed from GPT and Mixtral, covering the breadth of the LLM knowledge of our visual world. This large-scale dataset can be employed in many exciting future work directions, *e.g.,* fine-tuning multi-modal language models for enhanced fine-grained visual classification, or constructing large-scale synthetic datasets via generative text-to-image models for VLM pre-training.

Acknowledgment

This work was partially supported by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association, and the Austrian Research Promotion Agency (FFG) under the project SAFER (894164)

References

1. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – Mining Discriminative Components with Random Forests. In: Proc. ECCV (2014) 9, 12
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language Models are Few-Shot Learners. arXiv:2005.14165 (2020) 3, 5, 6, 8, 10
3. Chen, J., Li, D.Z.X.S.X., Zhang, Z.L.P., Xiong, R.K.V.C.Y., Elhoseiny, M.: MiniGPT-v2: Large Language Model as a Unified Interface for Vision-Language Multi-task Learning. arXiv preprint arXiv:2310.09478 (2023) 4
4. Chen, M., Du, J., Pasunuru, R., Mihaylov, T., Iyer, S., Stoyanov, V., Kozareva, Z.: Improving In-Context Few-Shot Learning via Self-Supervised Training. arXiv preprint arXiv:2205.01703 (2022) 5
5. Cheng, G., Han, J., Lu, X.: Remote Sensing Image Scene Classification: Benchmark and State of the Art. Proceedings of the IEEE **105**(10), 1865–1883 (2017) 9
6. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality (March 2023), https://lmsys.org/blog/2023-03-30-vicuna/ 4
7. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , Vedaldi, A.: Describing Textures in the Wild. In: Proc. CVPR (2014) 9, 10
8. Dai, W., Li, J., Li, D., Tiong, A., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. In: NeurIPS (2023) 4
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proc. CVPR (2009) 9
10. Fei-Fei, L., Fergus, R., Perona, P.: Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In: Proc. CVPR (2004) 9
11. Helber, P., Bischke, B., Dengel, A., Borth, D.: EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. In: Proc. IGARSS (2018) 9, 10, 13
12. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In: Proc. ICCV (2021) 9
13. Hou, Y., Dong, H., Wang, X., Li, B., Che, W.: MetaPrompting: Learning to Learn Better Prompts. arXiv preprint arXiv:2209.11486 (2022) 3

14. Huang, T., Chu, J., Wei, F.: Unsupervised Prompt Learning for Vision-Language Models. arXiv:2204.03649 (2022) 4
15. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T.: Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In: Proc. ICML (2021) 4
16. Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., Casas, D.d.l., Hanna, E.B., Bressand, F., et al.: Mixtral of Experts. arXiv preprint arXiv:2401.04088 (2024) 3, 5, 8, 9, 10
17. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The Kinetics Human Action Video Dataset (2017) 9
18. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: MaPLe: Multi-Modal Prompt Learning. In: Proc. CVPR (2023) 4
19. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large Language Models are Zero-Shot Reasoners. NeurIPS **35**, 22199–22213 (2022) 5
20. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D Object Representations for Fine-Grained Categorization. In: Proc. ICCVW (2013) 9, 12
21. Krizhevsky, A., Hinton, G.: Learning Multiple Layers of Features from Tiny Images. Tech. rep., Department of Computer Science, University of Toronto (2009) 9
22. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. arXiv preprint arXiv:2301.12597 (2023) 4
23. Lin, W., Karlinsky, L., Shvetsova, N., Possegger, H., Kozinski, M., Panda, R., Feris, R., Kuehne, H., Bischof, H.: MAtch, eXpand and Improve: Unsupervised Finetuning for Zero-Shot Action Recognition with Language Knowledge. In: Proc. ICCV (2023) 4
24. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved Baselines with Visual Instruction Tuning. arXiv:2310.03744 (2023) 4, 12, 13
25. Liu, H., Li, C., Li, Y., Lee, Y.J.: LLaVA-Next (LLaVA 1.6). arXiv:2310.03744 (2023), https://llava-vl.github.io/blog/2024-01-30-llava-next/ 4
26. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual Instruction Tuning. In: NeurIPS (2023) 4, 13
27. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-Grained Visual Classification of Aircraft. arXiv preprint arXiv:1306.5151 (2013) 9
28. Menon, S., Vondrick, C.: Visual Classification via Description from Large Language Models. Proc. ICLR (2023) 2, 5, 6, 10, 11
29. Min, S., Lewis, M., Zettlemoyer, L., Hajishirzi, H.: MetaICL: Learning to Learn In Context. arXiv preprint arXiv:2110.15943 (2021) 5
30. Mirza, M.J., Karlinsky, L., Lin, W., Possegger, H., Feris, R., Bischof, H.: TAP: Targeted Prompting for Task Adaptive Generation of Textual Training Instances for Visual Classification. arXiv preprint arXiv:2309.06809 (2023) 6
31. Mirza, M.J., Karlinsky, L., Lin, W., Possegger, H., Kozinski, M., Feris, R., Bischof, H.: LaFTer: Label-Free Tuning of Zero-shot Classifier using Language and Unlabeled Image Collections. In: NeurIPS (2023) 4
32. Nilsback, M.E., Zisserman, A.: Automated Flower Classification Over a Large Number of Classes. In: Proc. ICVGIP (2008) 9, 11
33. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: Proc. CVPR. pp. 3498–3505 (2012). https://doi.org/10.1109/CVPR.2012.6248092 9
34. Pratt, S., Liu, R., Farhadi, A.: What does a platypus look like? Generating customized prompts for zero-shot image classification. arXiv:2209.03320 (2022) 2, 5, 6, 9, 10, 11

35. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning Transferable Visual Models from Natural Language Supervision. In: Proc. ICML (2021) 1, 2, 4, 5, 6, 9, 10, 11, 12, 13, 14
36. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do ImageNet Classifiers Generalize to ImageNet? In: Proc. ICML. pp. 5389–5400. PMLR (2019) 9
37. Roth, K., Kim, J.M., Koepke, A., Vinyals, O., Schmid, C., Akata, Z.: Waffling around for Performance: Visual Classification with Random Words and Broad Concepts. arXiv preprint arXiv:2306.07282 (2023) 2, 5, 6, 10, 11
38. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C.W., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S.R., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: LAION-5b: An open large-scale dataset for training next generation image-text models. In: NeurIPS (2022) 4
39. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. arXiv:1212.0402 (2012) 9
40. Suzgun, M., Kalai, A.T.: Meta-Prompting: Enhancing Language Models with Task-Agnostic Scaffolding. arXiv preprint arXiv:2401.12954 (2024) 3
41. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv preprint arXiv:2307.09288 (2023) 4
42. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. rep., California Institute of Technology (2011) 9
43. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning Robust Global Representations by Penalizing Local Predictive Power. In: NeurIPS (2019) 9
44. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned Language Models Are Zero-Shot Learners. arXiv preprint arXiv:2109.01652 (2021) 5
45. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS **35**, 24824–24837 (2022) 5
46. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: SUN Database: Large-scale Scene Recognition from Abbey to Zoo. In: Proc. CVPR (2010) 9
47. Xu, H., Xie, S., Tan, X.E., Huang, P.Y., Howes, R., Sharma, V., Li, S.W., Ghosh, G., Zettlemoyer, L., Feichtenhofer, C.: Demystifying CLIP Data. In: Proc. ICLR (2023) 1, 4, 11, 12
48. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., Narasimhan, K.: Tree of Thoughts: Deliberate Problem Solving with Large Language Models. NeurIPS **36** (2023) 5
49. Zhao, Z., Wallace, E., Feng, S., Klein, D., Singh, S.: Calibrate Before Use: Improving Few-Shot Performance of Language Models. In: Proc. ICML. pp. 12697–12706. PMLR (2021) 5
50. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million Image Database for Scene Recognition. IEEE TPAMI **40**(6), 1452–1464 (2017) 9
51. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional Prompt Learning for Vision-Language Models. In: Proc. CVPR (2022) 4
52. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to Prompt for Vision-Language Models. IJCV (2022) 4, 10
53. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. arXiv preprint arXiv:2304.10592 (2023) 4