# Language Models as Black-Box Optimizers for Vision-Language Models

Shihong Liu* Samuel Yu* Zhiqiu Lin* Deepak Pathak Deva Ramanan

Carnegie Mellon University

## Abstract

*Vision-language models (VLMs) pre-trained on web-scale datasets have demonstrated remarkable capabilities on downstream tasks when fine-tuned with minimal data. However, many VLMs rely on proprietary data and are not open-source, which restricts the use of white-box approaches for fine-tuning. As such, we aim to develop a black-box approach to optimize VLMs through* **natural language prompts***, thereby avoiding the need to access model parameters, feature embeddings, or even output logits. We propose employing chat-based LLMs to search for the best text prompt for VLMs. Specifically, we adopt an automatic "hill-climbing" procedure that converges to an effective prompt by evaluating the performance of current prompts and asking LLMs to refine them based on textual feedback, all within a conversational process without human-in-the-loop. In a challenging 1-shot image classification setup, our simple approach surpasses the white-box continuous prompting method (CoOp) by an average of* $1.5\%$ *across 11 datasets including ImageNet. Our approach also outperforms both human-engineered and LLM-generated prompts. We highlight the advantage of conversational feedback that incorporates both positive and negative prompts, suggesting that LLMs can utilize the implicit "gradient" direction in textual feedback for a more efficient search. In addition, we find that the text prompts generated through our strategy are not only more interpretable but also transfer well across different VLM architectures in a black-box manner. Lastly, we demonstrate our framework on a state-of-the-art black-box VLM (DALL-E 3) for text-to-image optimization.*

## 1. Introduction

Vision-language models [1, 28, 51, 64] (VLMs) excel at a wide range of classic vision and multimodal [2, 10, 15, 32, 70] tasks, surpassing the performance of their fully-supervised counterparts on downstream tasks even when fine-tuned with minimal data [34, 74]. However, fine-tuning VLMs typically requires transparent *white-box* access to the model weights, such as gradient-based approaches that rely on backpropagation.

**VLMs as black-box services.** Despite community efforts to collect web-scale public datasets [55, 56] and to replicate proprietary VLMs [3, 24], an increasing number of models [1, 4, 13, 45, 64, 71] are not releasing their weights due to privacy and legal concerns [30, 38]. Therefore, one cannot use popular *white-box* fine-tuning strategies (such as LoRA [23] and Adapter [22]) that rely on model weights, feature embeddings, and output logits. Given that contemporary black-box VLMs [45, 47] like DALL-E [4, 52] still offer a language-based user interface and may be accessed through APIs that facilitate input and output in *natural language*, this allows users to customize these models through optimizing textual prompts.

**Manual prompting.** Manual prompt engineering has been proven successful in adapting black-box LLMs to language tasks [25, 66]. Similarly, carefully crafted prompts can enhance the performance of VLMs. For instance, CLIP has demonstrated improved zero-shot recognition performance using specifically tailored prompts, such as `"a photo of a {class}"` for Internet photos and `"a satellite image of a {class}"` for satellite imagery. Despite its effectiveness, manual prompting can be a laborious process, inspiring efforts to explore automated prompt creation and thereby remove the need for human involvement. These strategies typically leverage an LLM as a knowledge base to create rich visual descriptors that augment the prompts for each class [40, 50] in a zero-shot fashion.

**Human-free prompting with conversational LLMs (our approach).** We show how to effectively leverage chat-based LLMs [45] to emulate human-level prompt engineering *without* any human input. We first address an illustrative low-shot image classification task, aiming to find the best class-agnostic prompt (or "template") for image classification with CLIP. We start with a random set of prompts and evaluate the one-shot training accuracy of each. Then, akin to human prompt engineering, our method repeatedly presents ChatGPT with the best and worst prompts, asking it to review the results and suggest an improvement (see Figure 1).

**Learning with implicit "gradients" provided through conversational feedback.** One of our key findings is that

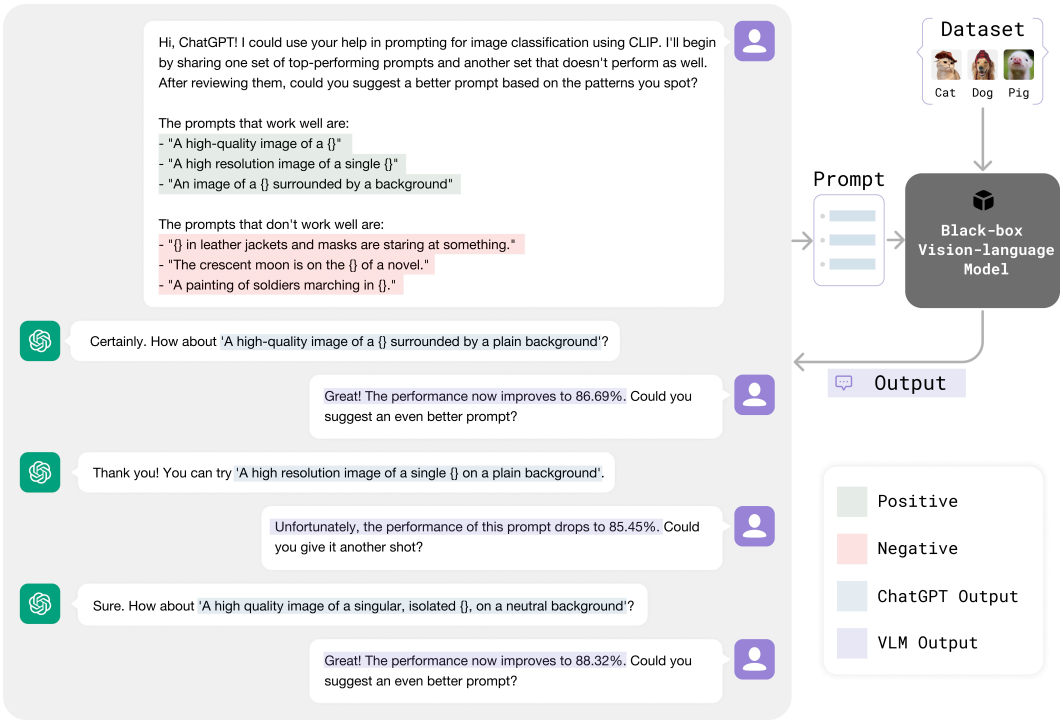


Figure 1. **Prompting VLMs using chat-based LLMs.** Similar to how human prompt engineers iteratively test and refine prompts, we employ ChatGPT [45, 47] to continuously optimize prompts for vision-language models (VLMs). Our iterative approach assesses the performance of ChatGPT-generated prompts on a few-shot dataset (highlighted in blue) and provides feedback (marked in violet) to ChatGPT through simple conversations, as depicted in the illustrative figure. This straightforward method delivers state-of-the-art results for one-shot image classification across 11 datasets using CLIP, operated in a black-box manner without accessing model weights, feature embeddings, or output logits. We show that providing both positive (in green) and negative prompts (in red) enhances efficiency. Remarkably, our approach outperforms both white-box methods such as gradient-based continuous prompting (CoOp [74]) and human-engineered prompts [51] in this extremely low-shot scenario. This figure only shows a typical conversation using ChatGPT's web user interface. Our code implementation follows this pattern using the ChatGPT API. We detail and ablate the prompts in section 8.

LLMs can learn the difference between effective and ineffective prompts, and can use this implicit "gradient" direction provided through language to perform more efficient searches. Compared to previous automatic prompting methods that only use LLMs as a knowledge base [40, 50] or paraphrasing tool [75], we show a novel use of LLMs as an *optimizer* that can utilize the patterns hidden in textual feedback. In our experiments, we find that the inclusion of such feedback greatly improves the efficiency and accuracy of our method, sometimes surpassing existing white-box methods [67, 74] on challenging one-shot scenarios.

**Optimizing text-to-image generation with DALL-E 3.** We further demonstrate our optimization framework on a state-of-the-art black-box VLM, DALL-E [4], for two illustrative one-shot generative tasks: (1) Text-to-image (T2I) generation (see Figure 3), where we sample challenging text queries from Winoground [63] that involve reasoning over compositions of objects, attributes, and relations. Examples include "`an animal watches a person`" and "`there is less milk than orange juice`", which DALL-E 3 might initially fail to generate. (2) Prompt inversion (see Figure 4), which attempts to reverse-engineer the textual prompt to generate a specific image for later customization [53] (see Table 5). To achieve this, we leverage conversational feedback from a multimodal LLM (GPT4-V [45]) to iteratively refine the prompts based on the current generated images. We present qualitative results in Table 4 and conduct a user study to demonstrate that our framework can be more efficient than manual prompting, even for graphical designers experienced with AI content-generation tools.

**Our contributions.** In this work, we introduce a novel prompting method for VLMs, utilizing an LLM as an *optimizer*. Our black-box approach can surprisingly compete with various white-box methods in a low-shot setting. Additionally, we extensively explore various strategies for conversing with ChatGPT, uncovering several key factors that significantly enhance the efficiency of this tool. We also show that our discovered natural language prompts are not only *interpretable* but also *transfer* better across CLIP architectures, eg., from RN50 to ViT/B-16, than continuous prompts discovered by previous white-box prompting

method [74]. Finally, we show practical applications of our framework on text-to-image generation using black-box DALL-E 3. We release our code for future research on prompt optimization and AI-driven content creation [1].

## 2. Related Works

**LLMs for multimodal tasks.** Cutting-edge LLMs like GPTs [45, 47] have been successfully applied to multimodal tasks, either through zero-shot composition with pre-trained multimodal models [29, 72] or by jointly fine-tuning with modality-specific encoders [1, 28] on large-scale multimodal datasets [56]. LLMs are also utilized as neuro-symbolic reasoners [16, 37, 58, 73], translating natural language instructions into modular programs (like Python code) that invoke APIs of multimodal models. In this work, we show the potential of LLMs as a *black-box optimizer* for multimodal foundation models with language interfaces, and more specifically vision-language models (VLMs).

**Prompt optimization of foundation models.** Following the success of in-context learning [6], which appends user-generated natural language instruction and few-shot samples to text inputs, prompting [35] has emerged as the preferred fine-tuning paradigm for LLMs due to its superior performance and parameter-efficiency. However, recent prompt optimization methods, including continuous prefix-tuning [7, 31, 61, 62, 69] and discrete token-searching [11, 12, 59], still operate in a white-box manner, requiring access to either the tokenizer or output logits. Moreover, black-box prompting methods, such as heuristic-based editing [41, 49], are tailored towards language-only tasks and are thus not applicable in VLM settings.

**LLMs for prompt optimization.** APE [75] leverages an LLM to automatically write prompts using few-shot samples based on instruction induction [21] and paraphrasing [42, 54]. However, it is only designed to address language tasks, while we focus on multimodal tasks using black-box VLMs. LLMs have also proven to be an effective external knowledge base [40, 50, 57] for generating prompts in a zero-shot setting for multimodal models. For example, DCLIP [40] uses GPT3 to come up with rich visual descriptions to improve zero-shot classification with CLIP [51]. We extend this line of work to show that LLMs can *iteratively* optimize prompts for VLMs in a black-box fashion given few-shot samples. We further illustrate that prompt optimization with LLMs can be made more efficient by leveraging *conversational* feedback, such as providing ChatGPT with explicit language feedback on how well the most recent prompt performs. Our findings align with the perspective [9] of LLMs as meta-optimizers that can implicitly perform gradient search through in-context learning.

[1]Project site: `llm-can-optimize-vlm.github.io`

**Few-shot adaptation of VLMs.** Prompting has also been successfully adopted in VLMs [14], as demonstrated by methods like CoOp [74] that fine-tune an ensemble of continuous prefix tokens using cross-entropy loss. [34] achieves state-of-the-art few-shot performance with a cross-modal (image and text) cross-entropy loss. However, these methods all require access to model parameters for gradient backpropagation. We also note that while some concurrent works, such as BlackVIP [44] and LFA [46], claim to operate in a "black-box" setting, they still require access to *privileged* information including output logits and embeddings. In this work, we introduce a truly black-box and gradient-free approach that yields competitive results to white-box approaches in extremely low-shot scenarios.

## 3. Prompting VLMs Using Chat-Based LLMs

We now present our approach for prompting VLMs using chat-based LLMs as optimizers.

**Preliminaries.** Motivated by recent proprietary VLMs [4, 45], we adopt a stricter yet practical black-box setting compared to prior works [44, 46], requiring *minimal* knowledge about the model's inner workings. This is crucial since releasing output logits or embeddings can potentially facilitate unauthorized knowledge extraction through distillation methods [19]. Our objective is to enhance the performance of a VLM equipped with a language interface capable of processing a textual prompt $p \in T$. We assume that the targeted task is accompanied by a training dataset denoted as $D_{train} \subset D$, and its performance can be evaluated with respect to the prompt, represented as a function $F : D \times T \rightarrow \mathbb{R}$. For example, in a classification task, $D_{train} = \{x, y\}_n$ where $x$ is an image and $y$ is its class label. The black-box VLM takes the image as input and returns a predicted label. We measure the performance of the textual prompt by calculating the average classification accuracy as $F(D_{train}, p)$. Our goal in prompt engineering is to search for the optimal prompt $p^*$ without accessing or modifying the black-box VLM.

**Background: human prompt engineering.** Our method draws inspiration from the typical workflow of human prompt engineers. Prompt engineering is often an iterative process that involves: (a) creating an initial prompt $U = \{p_1\}$ based on the understanding of a task, (b) evaluating the performance of prompts in $U$, (c) refining prompts based on the outcomes, (d) repeating the last two steps until convergence, and (e) returning the prompt $p^*$ with the highest $F(D_{train}, p^*)$. This hands-on approach helps optimize the model's performance, but it can be tedious and labor-intensive. Algorithm 1 formally illustrates this process.

**Example: prompting for image classification with CLIP [51].** CLIP is one of the most popular VLM that takes a set of class-specific prompts when performing "zero-shot" image classification. [51] details the laborious prompting procedure over the course of a year. Interestingly, they find

**Algorithm 1** We formalize *human* prompt engineering with the following algorithm, which motivates our LLM-based algorithm (2).

**Require:** $D_{\text{train}} = \{x, y\}_n$: training samples, $F : D \times T \rightarrow \mathbb{R}$: evaluation function

Create initial prompts: $\mathcal{U} \leftarrow \{p_1\}$
Evaluate prompts on training set: $S \leftarrow \{F(D_{\text{train}}, p_1)\}$
**while** not converged **do**
    Generate a new prompt $p'$ based on $S$
    Evaluate the new prompt: $s' = F(D_{\text{train}}, p')$
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{p'\}$
    $S \leftarrow S \cup \{s'\}$
**end while**
**return** optimal prompt $p^* \leftarrow \arg\max_{p \in \mathcal{U}} F(D_{\text{train}}, p)$

that a default class-agnostic prompt (or so-called "template"), "`a photo of a {class}`" can provide a decent boost in accuracy for most datasets compared to using vanilla class labels. In this scenario, the evaluation function $F$ is the classification accuracy on the test set, and the prompt $p = \{$"`a photo of a {class}`"$|c \in C\}$, where $C$ is the set of class names for a given dataset.

**Prompting with chat-based LLMs (our approach).** Given the strong in-context reasoning capabilities of LLMs, we envision them as a *black-box optimizers* that can improve prompts based on their performance outcomes, akin to how human prompt engineers iteratively refine prompts. Specifically, we maintain a pool of prompts $U$ and their corresponding performance outcomes $S$. In each iteration, we provide the LLM with both *positive* and *negative* prompts, such as the highest and lowest-performing candidates. Such textual feedback through in-context prompts offers LLMs an implied "gradient" direction [9], making optimization more efficient than taking random local steps. We facilitate this feedback mechanism through *conversations* with state-of-the-art chat-based LLMs like ChatGPT [47] as illustrated in Figure 1. We note that such a multi-turn conversation is not the only way of conversing with ChatGPT, and ablate different in-context feedback mechanisms in section 8.

## 4. Illustrative Few-Shot Classification Task

We illustrate our approach using a few-shot image classification task. Specifically, a prompt $p \in T$ consists of a set of class-specific prompts – that is, one textual description per class. The evaluation function $F$ takes the prompt $p$, along with an image dataset $D_{train}$, and returns the accuracy using the black-box VLM. To prevent overfitting and simplify our search space, we restrict our search to finding a single class-agnostic template, e.g., `a photo of a {}`, filling in the blank with label names provided with the dataset.

**Algorithm 2** LLM-based prompt engineering on the illustrative classification task. Our algorithm requires a chat-based LLM and a (black-box) evaluation function, such as accuracy. We highlight mechanisms for "exploration" (restart and reset) in blue and "exploitation" (iter) in red. We mark the key component of "conversational feedback" of our approach in violet. The actual prompts are attached in section 8.

**Require:** $D_{\text{train}} = \{x, y\}_n$: training samples, $F : D \times T \rightarrow \mathbb{R}$: evaluation function.
**Require:** $n_{\text{restart}}$: number of initial sampled prompt sets, $n_{\text{reset}}$: number of resets for a prompt set, $n_{\text{iter}}$: number of hill-climbing iterations, $m$: size of one initial prompt set, $k$: number of prompts send to ChatGPT.

$p^* \leftarrow \emptyset$
**for** $1{::}n_{\text{restart}}$ **do**
    Sample a new prompt set, $\mathcal{U}_{\text{init}} \leftarrow \{p_1, ..., p_m\}$
    **for** $1{::}n_{\text{reset}}$ **do**
        Reset to initial prompt set: $\mathcal{U} \leftarrow \mathcal{U}_{\text{init}}$
        **for** $1{::}n_{\text{iter}}$ **do**
            Sort $\mathcal{U}$ by score outcomes $\{F(D_{\text{train}}, p)\}_{p \in U}$
            $\mathcal{U}_{\text{top}} \leftarrow$ top-k prompts in $\mathcal{U}$
            $\mathcal{U}_{\text{bot}} \leftarrow$ bottom-k prompts in $\mathcal{U}$
            Get a new prompt $p_{\text{new}} \leftarrow \text{LLM}(\mathcal{U}_{\text{top}}, \mathcal{U}_{\text{bot}})$
            $\mathcal{U} \leftarrow \mathcal{U} \cup \{p_{\text{new}}\}$
        **end for**
        $p^* \leftarrow \arg\max_{p \in \mathcal{U} \cup \{p^*\}} F(D_{\text{train}}, p)$
    **end for**
**end for**
**return** prompt with highest score $p^*$

**Outline of our approach (Alg. 2).** To start, we sample entirely random initial prompts from a text corpus such as LAION-COCO [55] captions. Our approach follows the classical *stochastic hill-climbing framework with random-restart* [54], which prevents ChatGPT from being trapped in local optima by balancing "exploration" and "exploitation". Our **restart** mechanism is implemented by sampling $n_{\text{restart}}$ initial prompt sets to encourage exploration. Because ChatGPT performs stochastic top-k sampling for text generation (as we adopt the default temperature of 1.0), we also implement a **reset** mechanism to foster additional exploration by retrying a given prompt set $n_{\text{reset}}$ times. For exploitation, we converse with ChatGPT for $n_{\text{iter}}$ iterations. We find that it is critical to balance exploration and exploitation for optimal performance, and thoroughly examine this trade-off in section 9. Lastly, we present ChatGPT both the top and bottom-performing prompts, denoted as $(\mathcal{U}_{top}, \mathcal{U}_{bot})$. We show that this simple adjustment can improve the efficiency of our approach in Figure 2.

**Experimental setup.** We apply our approach to the few-shot image classification benchmark introduced in CoOp [74], which is the most commonly studied setup for

fine-tuning VLMs. This benchmark involves a collection of 11 datasets covering diverse image domains including ImageNet [10] and more niche datasets such as FGVC-Aircraft [39]. For each dataset, we adhere to the same three-fold k-shot train sets in [34], reporting the average accuracy across all folds. Importantly, our method only utilizes the train set to compute the score and does not require the few-shot validation set. We use CLIP following prior work [34, 74] to emulate a black-box VLM, and we employ ChatGPT (GPT3.5) as the chat-based LLM.

**Implementation details.** To start, we sample entirely random 1M initial prompts from a text corpus (LAION-COCO [55] captions). For each caption, we extract all the noun phrases using spaCy part-of-speech tagging [20]. Subsequently, we replace one noun phrase in the caption with ``{}'' (a placeholder where the class name will be inserted) to create a template. Given that each caption contains an average of 2 noun phrases, our initial prompt pool consists of approximately 2M templates. We run our algorithm with $n_{\text{restart}} = 20$ restarts, $n_{\text{resets}} = 50$ resets, and $n_{\text{iter}} = 10$ iterations. We opt to sample $m = 100$ prompts per restart and present the top and bottom $k = 15$ prompts to ChatGPT. We ablate different sets of hyperparameters and explain how we balance the tradeoff between exploration and exploitation in section 9. We adopt `gpt-3.5-turbo-0301` model for ChatGPT using OpenAI's official API and keep the default sampling temperature of 1.0. We also ablate `gpt-4` in Table 10 and find it achieves similar performance. The exact prompts used to converse with ChatGPT are documented in section 8. For a fair comparison, we use CLIP-RN50 for our experiments following prior work [34, 74]. We will open-source our code and release the initial prompt pool (LAIONCOCO-1M) to the public.

**Oracle white-box baselines.** Our black-box setup substantially differs from, and is more constrained than, the scenarios considered in previous white-box baselines. Specifically, we do **not** expose the pre-trained weights, model architectures, feature embeddings, or even output logits of VLMs. These constraints render many established *gradient-based fine-tuning* baselines inapplicable. Among the *oracle* white-box approaches we later compare to, **CoOp** [74] performs continuous prompting and requires backpropagation across all layers. **WiSE-FT** [67] ensembles fine-tuned weights with the original CLIP weights. **Cross-Modal Adaptation** [34] fine-tunes a linear classifier leveraging both image and text embeddings from CLIP. **BlackVIP** [44] and **LFA** [46] are two most recent baselines that apply CLIP logits or embeddings for gradient back-propagation. Finally, while **DCLIP [40]** queries GPT3 for rich visual descriptors for each class and does not require gradient-based fine-tuning, it performs *prompt ensembling* using 4-6 class-specific prompts, which breaches our black-box assumption for accessing the output logits.

**Black-box methods.** We additionally benchmark our method against truly black-box solutions, including the vanilla class-agnostic templates "`{class}`" and "`a photo of a {class}`". Also, we compare our approach to the best **Hand-Engineered** templates released by OpenAI, searched using *test set* performance to represent the theoretical upper bound of human performance, eg., "`a centered satellite photo of {class}.`" for EuroSAT [17]. Finally, we present two versions of conversational feedback of our approach: (a) using 30 positive (**P only**) or (b) using 15 positive and 15 negative prompts (**P+N**) in each iteration. For a fair comparison, both of our approaches start with the same initial sampled prompts, referred to as **LAIONCOCO-1M**. We also show the performance of the best initial sampled prompt searched using trainset performance.

**SOTA one-shot performance against existing methods on 11 datasets.** We report the test set performance of our method versus the aforementioned baselines in a challenging 1-shot classification scenario in Table 1. First, compared to the top-performing initial prompts selected from **LAIONCOCO-1M** based on train set performance, our prompt optimization using ChatGPT notably improves the initial prompts by an average of $5\%$ ($56\%$ to $61\%$). Remarkably, our black-box approach surpasses the two white-box gradient-based fine-tuning techniques CoOp and WiSE-FT by at least $1.5\%$. Given that both CoOp and our method optimize a single class-agnostic template, we attribute this gap in performance to *reduced overfitting*. More specifically, we posit that our optimization space of natural language effectively acts as a regularizer in extremely low-shot tasks, standing as a more robust alternative to the continuous prompting approach of CoOp. Furthermore, our method benefits from textual feedback and shows improved performance by $1.0\%$ when using both positive and negative prompts. In section 9, we show that our approach remains effective across different CLIP and ChatGPT variants.

**Incorporating negative prompts leads to more efficient optimization.** In Figure 2, we demonstrate that incorporating both positive and negative prompts fosters better optimization efficiency, achieving higher accuracy within a much fewer number of resets. Specifically, we hypothesize that LLMs can leverage the implicit "gradient" direction suggested in textual feedback to achieve faster convergence. For additional analysis, we ablate different ways of providing conversational feedback to ChatGPT in section 8 and conclude that iteratively updating both positive and negative prompts is the key for efficient optimization.

## 5. More Benefits of Natural Language Prompts

In this section, we delve deeper into the advantages of utilizing natural language prompts compared to the continuous prompts [74]. We highlight that the prompts derived through

| Method | Dataset | | | | | | | | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Caltech | ImageNet | Aircraft | Food | Pets | Cars | SUN | UCF | DTD | EuroSAT | Flowers | |
| **Oracle white-box approaches** | | | | | | | | | | | | |
| Cross-Modal [34] | 89.1 | 61.6 | 20.6 | 77.1 | 85.7 | 59.0 | 63.4 | 64.7 | 49.9 | 61.8 | 76.3 | 64.7 |
| WiSE-FT [67] | 85.5 | 58.3 | 18.6 | 71.9 | 81.7 | 55.7 | 56.6 | 59.4 | 44.2 | 52.3 | 65.8 | 59.1 |
| CoOp [74] | 87.5 | 57.2 | 9.6 | 74.3 | 85.9 | 55.6 | 60.3 | 61.9 | 44.4 | 50.6 | 68.1 | 59.6 |
| LFA [46] | 81.6 | 52.4 | 17.0 | 63.1 | 75.3 | 41.4 | 58.4 | 56.7 | 38.4 | 60.7 | 74.9 | 56.4 |
| BlackVIP [44] | 85.8 | 58.8 | 15.3 | 76.7 | 85.2 | 56.4 | 57.0 | 58.8 | 40.1 | 30.0 | 61.1 | 56.8 |
| DCLIP [40] | - | 59.6 | - | 76.4 | 83.8 | - | - | - | 41.7 | 34.7 | - | - |
| **Manual prompting approaches** | | | | | | | | | | | | |
| {} | 78.5 | 55.3 | 15.5 | 74.0 | 78.9 | 52.2 | 53.4 | 55.5 | 41.4 | 32.1 | 57.3 | 54.0 |
| a photo of a {} | 84.5 | 57.9 | 15.9 | 74.0 | 83.2 | 53.9 | 58.0 | 56.9 | 38.8 | 28.6 | 60.2 | 55.6 |
| Hand-Engineered [51] | 86.3 | 58.2 | 17.3 | 77.3 | 85.8 | 55.6 | 58.5 | **61.5** | 42.3 | 37.6 | 66.1 | 58.8 |
| **Our black-box approaches** | | | | | | | | | | | | |
| LAIONCOCO-1M | 81.4 | 56.2 | 17.4 | 76.5 | 79.6 | 51.3 | 54.9 | 55.8 | 43.1 | 38.6 | 61.3 | 56.0 |
| Ours (P only) | 89.0 | 59.4 | 17.9 | 77.8 | 85.7 | 55.7 | 60.4 | 58.7 | 43.6 | 46.7 | 66.6 | 60.1 |
| Ours (P+N) | **89.1** | **59.6** | **18.1** | **78.3** | **88.1** | **56.2** | **61.0** | 60.2 | **44.8** | **49.0** | **67.2** | **61.1** |

Table 1. **Comparison of our method with other baselines on one-shot classification tasks.** We report the average accuracy of each method across three folds, optimized using 1-shot training sets. We **bold** the best black-box result for each dataset, and underline the second best result. First, we note that our approach can effectively improve upon the initial prompts selected from LAIONCOCO-1M from 56% to 61%. Our approach is also competitive against the best Human-Engineered prompts released by OpenAI [51] searched using *test set* performance. Additionally, we show that using both positive and negative prompts improves the overall accuracy by 1%. For reference, we report *oracle* white-box approaches in gray. Remarkably, we also surpass white-box solutions such as WiSE-FT [67] and CoOp [74] by 1.5%. These methods require either gradient-based fine-tuning (CoOp/WiSE-FT/Cross-Modal) or prompt ensembling using output logits (DCLIP). While our approach is less effective than the SOTA white-box method (Cross-Modal Adaptation), we stress that our black-box setup is significantly more challenging, because we restrict the optimization space to *natural language* and do *not* access the pre-trained weights, model architectures, feature embeddings, and output logits of VLMs.

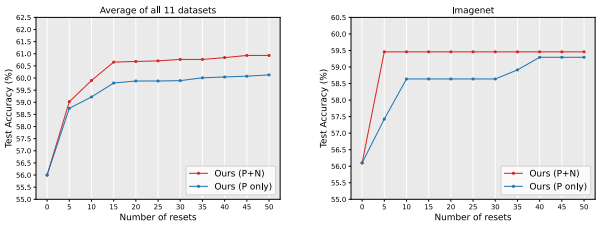


Figure 2. **Conversational feedback incorporating both positive and negative prompts leads to improved efficiency.** We fix the number of restarts to 20 and iterations to 10, and ablate different numbers of resets on all 11 datasets (left) and ImageNet (right). Notably, our approach using "P+N" (both top-15 and bottom-15 prompts) can optimize faster within a much fewer number of resets than using "P-Only" (top-30 prompts), resulting in the highest overall performance.

our method are *interpretable*; for instance, they often contain descriptions of the targeted image domain. Our prompts can also *transfer* across CLIP architectures in a *black-box manner*, such as from RN50 to ViT/B-16.

**Interpretable natural language prompts.** While CoOp [74] concedes that continuous prompts can be difficult to interpret, our method – without explicitly instructing ChatGPT to generate interpretation – often yields interpretable results. Table 2 showcases the templates returned by our algorithm for each dataset, frequently including keywords that reflect the targeted image domain. For example, the template for Food101 [5] mentions "diverse cuisine and ingredients", and the template for UCF101 [60] (an action recognition dataset) mentions "in motion". Likewise, these templates identify general stylistic attributes of the datasets; they refer to "bright and natural lighting" for ImageNet [10] and note images that "emphasize the subject" for Caltech101 [27]. These prompts are particularly intriguing because we do not provide ChatGPT with any information about the downstream task, yet it manages to generate prompts containing domain-specific keywords that are similar to those engineered by human experts.

**Black-box prompt transfer.** Our text prompts also maintain consistently high performance across different CLIP backbones. For comparison, since CoOp uses the same tokenizer for all CLIP architectures (including RN50, RN101, ViT/B-32, and ViT/B-16) and optimizes continuous prompts of the same shape (16 x 512), we assess the transferability of these learned continuous prompts from RN50 to other backbones using the official weights on 16-shot ImageNet. Table 3 showcases the results of this experiment, where we also include the baseline prompt "`a photo of a {}`" for reference. We observe a significant decline in accuracy when transferring CoOp's prompts (up to a 40% decrease despite utilizing more powerful backbones), implying that continuous prompts tend to overfit to the specific CLIP model. In contrast, our natural language prompts maintain their performance and outperform the baseline across all backbones.

| Dataset | Example of Top Templates |
|---|---|
| Caltech [27] | `An image of a {} with a blurred background that emphasizes the subject` |
| DTD [8] | `The essential elements of {} are amplified with visual simplicity` |
| EuroSAT [17] | `A top-down view of {} arranged in a pattern {}` |
| Aircraft [39] | `A clear, high-quality image of a single {} with a white background` |
| Food [5] | `A {} featuring diverse cuisine and ingredients` |
| ImageNet [10] | `An image of a {} with bright and natural lighting` |
| Flowers [43] | `A clear and vivid photograph of the {} in its natural setting` |
| Pets [48] | `A {} with distinct and recognizable characteristics` |
| Cars [26] | `A {} featuring a wide range of color options for easy selection` |
| SUN [68] | `A high-resolution photo of a {} with clear background and natural lighting` |
| UCF [60] | `A black and white photo of a {} in motion` |

Table 2. **Example templates returned by our algorithm on each dataset.** Although we do not provide ChatGPT with any information regarding the targeted dataset, we observe that the resulting templates are remarkably similar to human-engineered templates, with many domain-specific details such as "motion" and "cuisine", and stylistic elements such as "bright and natural lighting".

| Method | RN50 | →RN101 | →ViT-B/32 | →ViT-B/16 |
|---|---|---|---|---|
| `a photo of a {}` | 57.9 | 60.6 | 61.9 | 66.6 |
| CoOp | **63.0** | 20.6 | 31.7 | 39.5 |
| Ours | 59.9 | **60.7** | **62.2** | **67.0** |

Table 3. **Black-box prompt transfer from ResNet-50 to other CLIP architectures.** We evaluate both our natural language prompts and CoOp's continuous prompts on 16-shot ImageNet, which are trained using the RN50 CLIP backbone. As a reference point, we include the baseline prompt "`a photo of a {}`", and show that the prompts derived from our method using RN50 consistently surpass it after transferring to different backbones. In contrast, while CoOp achieves better 16-shot ImageNet performance using RN50, its performance plummets during the transfer, e.g., from 63% to a mere 21% for RN101.

## 6. Application: Text-to-Image Generation

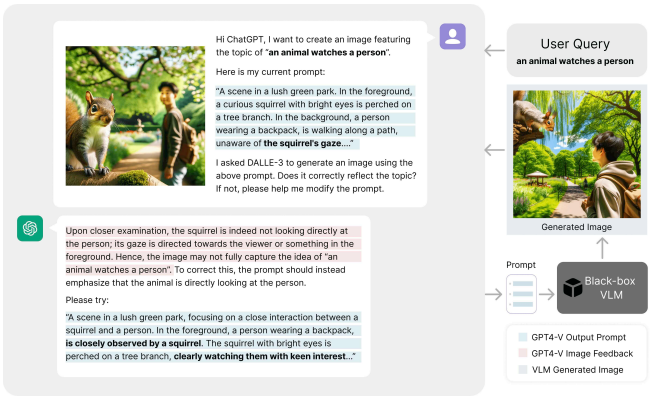


Figure 3. **Improving text-to-image (T2I) generation using chat-based *multimodal* LLMs.** Our framework can use the multimodal GPT4-V [45] to optimize prompts for the state-of-the-art black-box generative VLM, DALL-E 3 [4]. For complex user queries that DALL-E 3 may initially fail to generate, we send the generated image (in violet) along with the current prompt to GPT4-V to ask for feedback on improvements (in red) and then generate a new prompt (in blue). Our simple framework is surprisingly effective at correcting DALL-E 3 mistakes on some challenging Winoground [63] text queries that involve action, logical, and spatial reasoning. We conduct a human evaluation on the quality of generated images in Table 6 and include the actual prompts in section 8.

In this section, we present a direct application of our prompt optimization framework to generative tasks using a truly black-box text-to-image (T2I) VLM, DALL-E 3 [4].

**Optimizing T2I using a multimodal LLM.** DALL-E 3 can generate high-fidelity images following diverse user queries, but crafting effective prompts is tricky even for designers experienced with AI content generation tools [36]. Therefore, we are motivated to implement our LLM-based optimization framework to assist with creative visual design. Our framework is shown in Figure 3 for the illustrative task of text-to-image generation. In this task, the user specifies a query (topic) in text, such as "`an animal watches a person`", and the goal is to write a prompt that can generate an image reflecting this topic. We adopt a *multimodal* LLM GPT4-V [45] (`gpt-4-1106-preview`) to provide feedback on the generated image and optimize the prompt. We find that this framework is surprisingly effective due to GPT4-V's strong visual reasoning capabilities, which can often spot subtle errors in generated images and offer more accurate prompts.

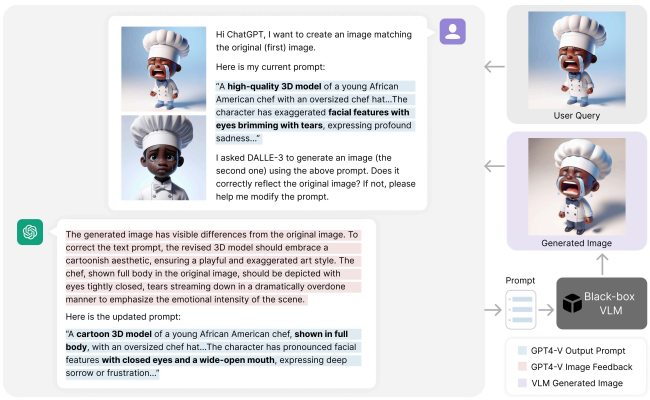


Figure 4. **Prompt inversion using chat-based *multimodal* LLMs.** We apply our framework to reverse engineer the text prompt to generate the same user-queried image. We send the generated image (in violet) along with the original image to GPT4-V to ask for feedback on improvements (in red) and then generate a new prompt (in blue).

**Task setup.** For *T2I generation*, we experiment with a

| User Query | Init. Image | LLM Feedback | Final Image |
|---|---|---|---|
| **Text-to-image generation** | | | |
| There is less milk than orange juice. |  | Incorrect, the milk bottle appears full, more than orange juice... |  |
| A shorter person is covering the eyes of a taller person. |  | Incorrect, the taller person is covering the shorter person's eyes. Instead, ... |  |
| **Prompt inversion** | | | |
|  |  | The scarf should feature red and white stripes, and the fur is fluffy... |  |
|  |  | The coat should be buttoned and the lighting exhibits a stronger contrast... |  |

Table 4. **Examples of T2I optimization.** We show that our framework (Figure 3) can automatically improve the faithfulness of images generated by DALL-E 3, with respect to user-specified textual topics (for T2I generation) or reference images (for prompt inversion). This is achieved through three rounds of prompt optimization, using feedback from the multimodal LLM (GPT4-V). Table 11 and Table 12 shows more examples with actual prompts.

| User Query | Inverted Image | Example 1 | Example 2 | Example 3 | Example 4 | Example 5 |
|---|---|---|---|---|---|---|
|  |  |  Give the dog a cat friend. |  Make the dog be in the middle of a jump. |  Make the dog do a handstand. |  Make the dog lie down on its side. |  Make the dog swim in water. |
|  |  |  Make the owl fight a hawk. |  Make the owl flap its wings. |  Make the owl fully green. |  Make the owl stand in front of the moon. |  Make the owl walk in the city. |

Table 5. **Customization via prompt inversion.** Users can simply append extra descriptions to the inverted prompts to customize their main characters in queried images.

subset of 100 text queries from Winoground [63] that involve complex attribute and relation reasoning, which DALL-E might initially fail to generate. Our framework refines the prompts to capture the user-specified topics using a few (three) iterations. We also attempt a reverse task of *prompt inversion*: given a user-specified reference (query) image, our framework reverse-engineers the prompt to have DALL-E generate the same object or scene in the query image (see Figure 4). This enables users to easily make customizations [53] (see Table 5), such as having the character in a reference image perform various actions or change scenes. For this task, we sample 100 random queries from DiffusionDB [65]. We provide qualitative results in Table 4, Table 11, and Table 12. We hire two volunteers to assess the faithfulness of the images generated by our method, and to compare these with the images manually prompted by two designers (each with one year of experience in AI content generation), as shown in Table 6.

**Remarks on limitations.** While we show promising results, we note some failure cases in Table 13 and Table 14 due to the inherent limitations of foundation models. For example, GPT4-V might fail to describe abstract and artistic details, and DALL-E 3 often fails to generate the correct number of objects. We believe that our framework can benefit from more capable foundation models in the future.

| Task | Method | Init. (std) | Final (std) | Δ |
|---|---|---|---|---|
| Text-to-Image | Human | 2.28 (.45) | 2.86 (.61) | 0.58 |
| | Ours | 2.62 (.36) | 3.56 (.54) | 0.94 |
| Prompt Inversion | Human | 1.58 (.48) | 2.76 (.53) | 1.18 |
| | Ours | 1.94 (.39) | 3.68 (.47) | 1.74 |

Table 6. **Our method enhances faithfulness in T2I generation.** We hire two human annotators to assess the faithfulness of images generated from user queries, e.g., textual topics for Text-to-Image, or reference images for Prompt Inversion. The scores are measured on a 1-to-5 Likert scale, with 1 signifying contradiction and 5 indicating perfect alignment with the user's goal. Our approach benefits from three iterations of prompt optimization and consistently outperforms human-engineered prompts by designers who have one year of experience in AI content generation.

## 7. Discussion and Limitations

**Summary.** We present the first attempt to leverage LLMs as prompt engineers for VLMs. On one-shot image classification, our method surpasses existing human-engineered prompts and even rivals white-box approaches. Central to the success of our method is the use of conversational feedback, enabling chat-based LLMs to efficiently steer VLMs in the right direction. This process leads to a set of interpretable prompts bearing considerable resemblance to those crafted by humans. Importantly, our natural language prompting setup is a lot more constrained than the assumed scenarios of previous white-box or even some black-box settings [44], because we do not expose the model weights and outputs of VLMs. We finally apply our framework to illustrative generative tasks using a truly black-box text-to-image VLM (DALL-E 3).

**Limitations and future work.** While we try to minimize the overall cost and the total number of API calls, the energy consumption associated with LLMs remains a substantial concern. It is vital to note that we do not intend to compete directly with white-box baselines that can improve visual and text representations with more data. Further details on the higher-shot performance of our method can be found in section 9. Lastly, we are limited to costly human evaluation for T2I generation in this study. Future work may adopt automatic evaluation [18, 33] for large-scale experiments.

## References


[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 1, 3

[2] 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*, pages 2425–2433, 2015. 1

[3] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 1

[4] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Joyce Zhuang, Juntang andLee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiang, and Aditya Ramesh. Improving image generation with better captions. *Note on Dalle-3*, 2023. 1, 2, 3, 7

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 6, 7

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[7] Yekun Chai, Shuohuan Wang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Clip-tuning: Towards derivative-free prompt learning with a mixture of rewards. *arXiv preprint arXiv:2210.12050*, 2022. 3

[8] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 7

[9] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *ACL*, 2023. 3, 4

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 5, 6, 7

[11] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P. Xing, and Zhiting Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*, 2022. 3

[12] Shizhe Diao, Xuechun Li, Yong Lin, Zhichao Huang, and Tong Zhang. Black-box prompt learning for pre-trained language models. *arXiv preprint arXiv:2201.08531*, 2022. 3

[13] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 1

[14] Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, and Jianfeng Gao. Vision-language pre-training: Basics, recent advances, and future trends. *Foundations and Trends® in Computer Graphics and Vision*, 14(3-4):163–352, 2022. 3

[15] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 1

[16] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. *arXiv preprint arXiv:2211.11559*, 2022. 3

[17] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification, 2017. 5, 7

[18] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 8

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[20] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1): 411–420, 2017. 5

[21] Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. Instruction induction: From few examples to natural language task descriptions. *arXiv preprint arXiv:2205.10782*, 2022. 3

[22] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 1

[23] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1

[24] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 1

[25] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022. 1

[26] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In

*4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 7

[27] Li, Andreeto, Ranzato, and Perona. Caltech 101, 2022. 6, 7

[28] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1, 3

[29] Shuang Li, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, and Igor Mordatch. Composing ensembles of pre-trained models via iterative consensus. *arXiv preprint arXiv:2210.11522*, 2022. 3

[30] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020. 1

[31] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 3

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1

[33] Zhiqiu Lin, Xinyue Chen, Deepak Pathak, Pengchuan Zhang, and Deva Ramanan. Revisiting the role of language priors in vision-language models. *arXiv preprint arXiv:2306.01879*, 2023. 8

[34] Zhiqiu Lin, Samuel Yu, Zhiyi Kuang, Deepak Pathak, and Deva Ramana. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. *arXiv preprint arXiv:2301.06267*, 2023. 1, 3, 5, 6

[35] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. 3

[36] Vivian Liu. Beyond text-to-image: Multimodal prompts to explore generative ai. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2023. 7

[37] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*, 2023. 3

[38] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1

[39] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 5, 7

[40] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *ICLR*, 2023. 1, 2, 3, 5, 6

[41] Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*, 2021. 3

[42] Melanie Mitchell, John Holland, and Stephanie Forrest. When will a genetic algorithm outperform hill climbing. *Advances in neural information processing systems*, 6, 1993. 3

[43] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 7

[44] Changdae Oh, Hyeji Hwang, Hee-young Lee, YongTaek Lim, Geunyoung Jung, Jiyoung Jung, Hosik Choi, and Kyungwoo Song. Blackvip: Black-box visual prompting for robust transfer learning. In *CVPR*, 2023. 3, 5, 6, 8

[45] OpenAI. Gpt-4 technical report. 2023. 1, 2, 3, 7

[46] Yassine Ouali, Adrian Bulat, Brais Matinez, and Georgios Tzimiropoulos. Black box few-shot adaptation for vision-language models. In *ICCV*, 2023. 3, 5, 6

[47] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 1, 2, 3, 4

[48] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 7

[49] Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. Grips: Gradient-free, edit-based instruction search for prompting large language models. *arXiv preprint arXiv:2203.07281*, 2022. 3

[50] Sarah Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. *ICCV*, 2023. 1, 2, 3

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 6, 13

[52] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1

[53] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 2, 8

[54] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010. 3, 4

[55] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 1, 4, 5

[56] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 1, 3

[57] Sheng Shen, Chunyuan Li, Xiaowei Hu, Yujia Xie, Jianwei Yang, Pengchuan Zhang, Zhe Gan, Lijuan Wang, Lu Yuan, Ce Liu, et al. K-lite: Learning transferable visual models with external knowledge. *Advances in Neural Information Processing Systems*, 35:15558–15573, 2022. 3

[58] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580*, 2023. 3

[59] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020. 3

[60] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6, 7

[61] Tianxiang Sun, Zhengfu He, Hong Qian, Yunhua Zhou, Xuan-Jing Huang, and Xipeng Qiu. Bbtv2: Towards a gradient-free future with large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3916–3930, 2022. 3

[62] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022. 3

[63] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022. 2, 7, 8

[64] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. 1

[65] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv preprint arXiv:2210.14896*, 2022. 8, 15

[66] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022. 1

[67] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021. https://arxiv.org/abs/2109.01903. 2, 5, 6

[68] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *Int. J. Comput. Vision*, 119(1): 3–22, 2016. 7

[69] Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Yanggang Wang, Haiyu Li, and Zhilin Yang. Gps: Genetic prompt search for efficient few-shot learning. *arXiv preprint arXiv:2210.17041*, 2022. 3

[70] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 1

[71] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 1

[72] Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022. 3

[73] Mingkai Zheng, Xiu Su, Shan You, Fei Wang, Chen Qian, Chang Xu, and Samuel Albanie. Can gpt-4 perform neural architecture search? *arXiv preprint arXiv:2304.10970*, 2023. 3

[74] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022. 1, 2, 3, 4, 5, 6

[75] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *ICLR*, 2023. 2, 3, 13, 14