This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Leveraging Large Language Models for Multimodal Search

Oriol Barbany[1†] Michael Huang[2*] Xinliang Zhu[2*] Arnab Dhua[2]
[1]Institut de Robòtica i Informàtica Industrial, CSIC-UPC [2]Visual Search & AR, Amazon

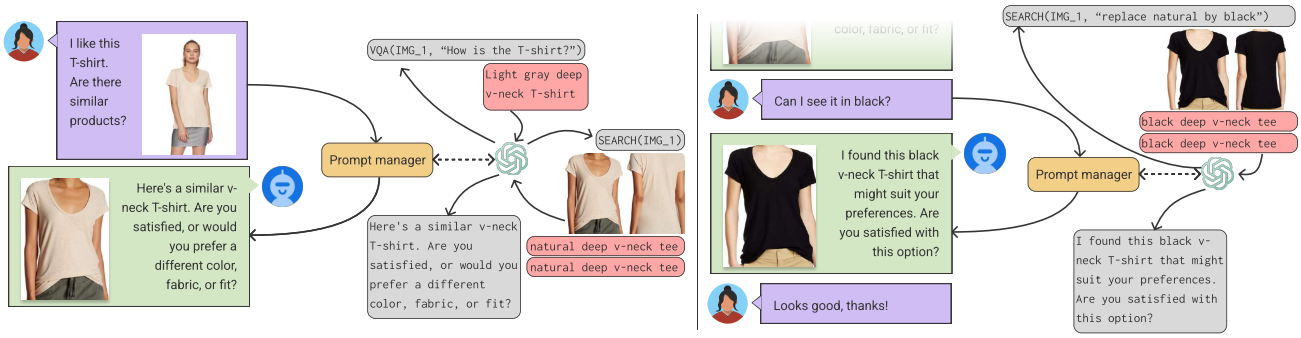

Figure 1. **Overview**: This paper introduces a comprehensive pipeline for multimodal search, presenting a novel composed retrieval model that outperforms previous approaches significantly. Additionally, we propose a system that utilizes a Large Language Model (LLM) as an orchestrator to invoke both our proposed model and other off-the-shelf models. The resulting search interface offers a conversational search assistant experience, integrating information from previous queries and leveraging our novel model to enhance search capabilities.

Abstract

Multimodal search has become increasingly important in providing users with a natural and effective way to express their search intentions. Images offer fine-grained details of the desired products, while text allows for easily incorporating search modifications. However, some existing multimodal search systems are unreliable and fail to address simple queries. The problem becomes harder with the large variability of natural language text queries, which may contain ambiguous, implicit, and irrelevant information. Addressing these issues may require systems with enhanced matching capabilities, reasoning abilities, and context-aware query parsing and rewriting. This paper introduces a novel multimodal search model that achieves a new performance milestone on the Fashion200K dataset [19]. Additionally, we propose a novel search interface integrating Large Language Models (LLMs) to facilitate natural language interaction. This interface routes queries to search systems while conversationally engaging with users and considering previous searches. When coupled with our multimodal search model, it heralds a new era of shopping assistants capable of offering human-like interaction and enhancing the overall search experience.

† Work performed during an internship at Amazon.
* Equal contribution.

1. Introduction

The Composed Image Retrieval (CIR) problem, also known as Text-Guided Image Retrieval (TGIR), involves finding images that closely match a reference image after applying text modifications. For instance, given a reference image of a blue dress and the instruction "`replace blue with red`", the retrieved images should depict red dresses resembling the reference.

It is natural for users to search for products using information from multiple modalities, such as images and text. Enabling visual search allows for finding visually similar correspondences and obtaining fine-grained results. Otherwise, text-only search tools would require extensive textual descriptions to reach the same level of detail. Thus, it is more natural and convenient for users to upload a picture of their desired product or a similar version rather than articulating their search entirely in words.

Traditional search engines often struggle to deliver precise results to users due to the challenges posed by overly specific, broad, or irrelevant queries. Moreover, these engines typically lack support for understanding natural language text and reasoning about search queries while conversationally engaging with the user.

In the context of the Fashion200K benchmark [19], several existing approaches fail to retrieve the correct query among the top matches. Concretely, most of the baselines considered in this work fail to retrieve the correct image

among the top 10 matches in 60% of the cases, as shown in our results in Sec. 4.1.

In this paper, we propose to leverage pretrained large-scale models that can digest image and text inputs. We focus on improving the performance on the Fashion200K dataset [19] and achieve state-of-the-art results that improve upon previous work by a significant margin. However, all the queries in Fashion200K follow the simple formatting "`replace {original_attribute} with {target_attribute}`", which impedes generalizing to natural language text. For this reason, we develop a novel interactive multimodal search solution leveraging recent advances in LLMs and vision-language models that can understand complex text queries and route them to the correct search tool with the required formatting. Leveraging LLMs facilitates digesting natural language queries and allows taking contextual information into account. Moreover, the length of the context recent LLMs can consider allows for incorporating information from previous interactions. We include a high-level overview of our approach in Fig. 1. The main contributions of this work include:

- **Improved Multimodal Search:** We introduce a method that adapts foundational vision and language models for multimodal retrieval, which achieved state-of-the-art results on Fashion200k. We present the technical details in Sec. 3.1 and discuss the experimental results in Sec. 4.1.
- **Conversational Interface:** We propose an interface that harnesses state-of-the-art LLMs to interpret natural language inputs and route formatted queries to the available search tools. We describe the details of the backend in Sec. 3.2 and include examples in Sec. 4.2.

2. Related work

When tackling the CIR problem, the TIRG model [46] computes an image representation and modifies it with a text representation on the same space rather than fusing both modalities to create a new feature as in most of the other works. Crucially, this method is trained first on image retrieval and gradually incorporates text modifications.

The VAL framework [10] is based on computing image representations at various levels and using a transformer [45] conditioned on language semantics to extract features. Then, an objective function evaluates the feature similarities hierarchically.

The text and image encoders of a CLIP model [36] can be used for zero-shot retrieval with a simple Multi-Layer Perceptron (MLP) [40] and leveraging LLMs [5]. Another approach is to perform a late fusion of CLIP embeddings [4], which can be improved by fine-tuning the CLIP text encoder Baldrati et al. [3]. The hypothesis is that image and text embeddings obtained by CLIP are aligned, while the CIR problem requires a text representation that expresses differences w.r.t. the image representation.

CosMo [25] independently modulates the content and style of the reference image based on the modification text. This work assumes that style information is removed by simply performing instance normalization on the image features. With this assumption in mind, the normalized features are fed to the content modulator, which transforms them conditioned on text features. Then, the output of the content modulator is given to the style modulator, which along with the text features and the channel-wise statistics of the normalization, obtains the final representation.

FashionVLP [16] is based on extracting image features using a pretrained feature extractor, not only on the whole image but also on the cropped clothing, fashion landmarks, and regions of interest. The obtained image representations are concatenated with object tags extracted with an object detector, a class token, and the word tokens computed using BERT [13].

An alternative to tackle the problem of generic visual feature extractors not focusing on fashion-specific details without using the multiple inputs required in Goenka et al. [16], is proposed in FashionSAP [20]. FashionSAP leverages the FashionGen dataset [39] for fine-grained fashion vision-language pretraining. To do that, Han et al. [20] use a multi-task objective composed of retrieval and language modeling losses. CIR is then solved by fusing text and image features using multiple cross-attention layers, and the tasks included in the training objective are solved using different heads for each task.

CompoDiff [17] proposes to solve the CIR using a denoising transformer that provides the retrieval embedding conditioned on features of the reference image and the modifying text. Similarly to Rombach et al. [38], the diffusion process is performed in the latent space instead of the pixel space. Given that CompoDiff is a data-hungry method, Gu et al. [17] create a synthetic dataset of 18 million image triplets using StableDiffusion [38] for its training. CompoDiff performs better when using text features obtained with a T5-XL model [37] in addition to the text representations obtained with [36].

Koh et al. [23] uses a frozen LLM to process the input text and visual features that have been transformed with a learned linear mapping as in LLaVA [32]. To counteract the inferior expressiveness of causal attention over its bidirectional counterpart, Koh et al. [23] append a special `[RET]` token at the end of the outputs that allows the LLM to perform an extra attention step over all tokens. The hidden representations of `[RET]` are then mapped to an embedding space that is used for retrieval.

Couairon et al. [12] tackle a similar problem in which the transformation query is not a single word but a tuple of two words corresponding to the original and target attributes. As an example, for a reference image with caption "`A cat is sitting on the grass`", a source text "`cat`"

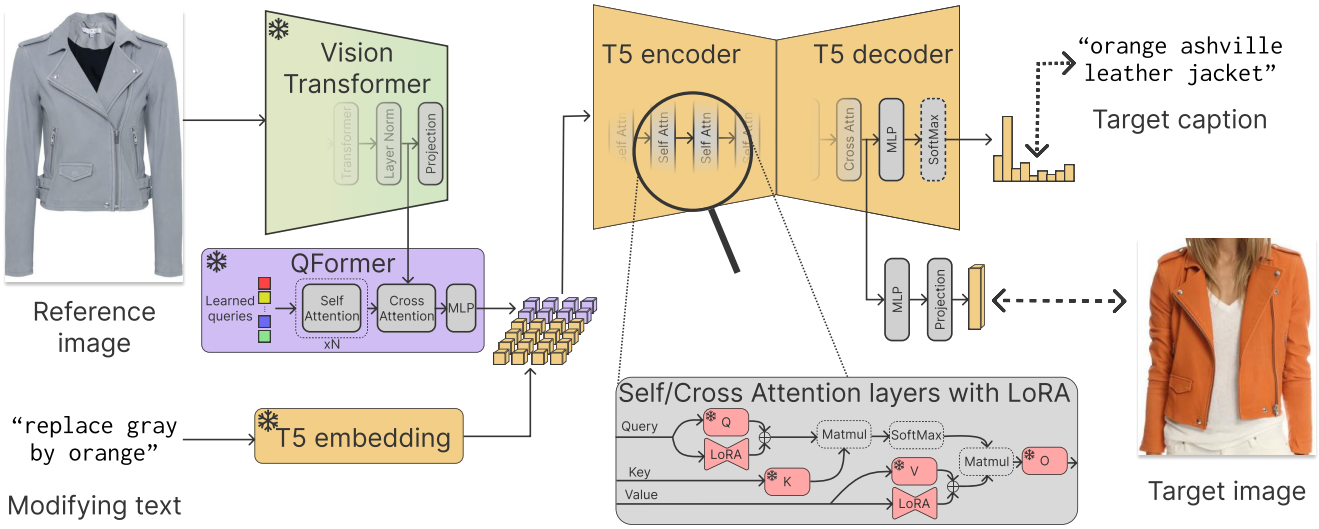

Figure 2. **Proposed architecture**: We extract visual features from the reference image $\mathbf{x}_{\text{ref}}$ using a Vision Transformer [14], specifically, a pretrained CLIP [36] model with frozen weights. We extract features before the projection layer, which are then processed using a Querying transFormer (Q-Former), which performs cross-attention with a set of learned queries. The resulting output of the Q-Former is concatenated with the embeddings obtained from the modifying text ($\mathbf{t}$), which expresses a modification in the reference image. Subsequently, all this information is fed into a T5 model [37], an encoder-decoder LLM. We employ Low-Rank Adaptation (LoRA) [21] to learn low-rank updates for the query and value matrices in all attention layers, while keeping the rest of the parameters frozen. The output of the LLM yields a probability distribution from which a sentence is generated. To ensure alignment with the target caption (*i.e.*, the caption of the target image $\mathbf{x}_{\text{trg}}$, which corresponds to the caption of the reference image after incorporating the text modifications), a language modeling loss is used. The hidden states of the LLM are then projected into a space of embeddings used for retrieval. A retrieval loss term pushes together the embedding of the target image $\mathcal{G}(\mathbf{x}_{\text{trg}})$ and that obtained using the reference image and the modifying text $\mathcal{F}(\mathbf{x}_{\text{ref}}, \mathbf{t})$.

and a target text `"dog"`, the model should be able to retrieve images of dogs sitting on the grass.

3. Method

In this section, we propose a model to perform an image search merging text and image inputs in Sec. 3.1. While this model outperforms alternative approaches by a large margin, it is trained on a dataset with specific formatting (see Sec. 4.1). Instead of artificially augmenting the vocabulary seen during training as in Gu et al. [17], we propose a conversational interface orchestrated by a LLM that can structure the queries to a format understandable to our multimodal search model.

Sec. 3.2 describes the principles of our approach. The proposed framework offers a modular architecture that allows interchanging search models with different formatting constraints while providing enhanced natural language understanding, a working memory, and a human-like shopping assistant experience.

3.1. Improved multimodal search

In the CIR problem, a dataset $\mathcal{D}$ is composed of triplets with reference and target image as well as a modifying text, *i.e.*, $\mathcal{D} := \{(\mathbf{x}_{\text{ref}}^{(i)}, \mathbf{x}_{\text{trg}}^{(i)}, \mathbf{t}^{(i)})\}_{i \in [n]}$. The objective is to learn the transformations

$$\mathcal{F} : \mathbf{x}_{\text{ref}} \times \mathbf{t} \mapsto \Psi \quad ; \quad \mathcal{G} : \mathbf{x}_{\text{trg}} \mapsto \Psi \tag{1}$$

along with a metric space (Ψ, d) with fixed $d : \Psi \times \Psi \to \mathbb{R}$ such that

$$d(\mathcal{F}(\mathbf{x}_{\text{ref}}, \mathbf{t}), \mathcal{G}(\mathbf{x}_{\text{trg}})) < d(\mathcal{F}(\mathbf{x}_{\text{ref}}, \mathbf{t}), \mathcal{G}(\mathbf{x}'_{\text{trg}})) \tag{2}$$

if $\mathbf{x}_{\text{ref}}$ after applying the modifications described by $\mathbf{t}$ is semantically more similar to $\mathbf{x}_{\text{trg}}$ than it is to $\mathbf{x}'_{\text{trg}}$ [6]. Commonly to other works [41, 43, 53], we normalize the space $\Psi$to the unit hypersphere for training stability, and choose d to be the cosine distance.

In this work, we use off-the-shelf foundational models for vision and language to compute the transformation $\mathcal{F}$. Concretely, we use an architecture similar to BLIP2 [29] and adapt it for the CIR problem. BLIP2 [29] uses a module referred to as the Q-Former, which allows ingesting image features obtained by a powerful feature extractor. These image features provide fine-grained descriptions of the input product and are transformed into the space of text embeddings of a LLM. Then, the LLM processes the fused text and image embeddings.

The Q-Former consists of two transformer submodules sharing the same self-attention layers to extract information from the input text and the image features. The image trans-

formers also contain a set of learnable query embeddings, which can be interpreted as a form of prefix tuning [30].

To generate image-only search embeddings using our model, one simply needs to input the images into the model and provide an empty string as the input text. Intuitively, this processes the images without any text modifications. In other words, we use

$$\mathcal{G}(\mathbf{x}) := \mathcal{F}(\mathbf{x}, \texttt{""}) \tag{3}$$

We illustrate the proposed architecture for $\mathcal{F}$ in Fig. 2. We use the image part of the CLIP [36] model to obtain visual features and a T5 model [37] as LLM to process the modifying text and the visual features processed by the Q-Former.

We initialize the model using the BLIP2 weights with all the parameters frozen. The pretrained weights perform the task of image captioning, which is different from the task we are trying to solve. Instead, we define a new task that we refer to as *composed captioning*. The objective of this task is to generate the caption of the product that we would obtain by merging the information of the product in the input image and the text modifications.

We hypothesize that if the proposed model can solve the problem of *composed captioning*, the information captured by the LLM is enough to describe the target product. Intuitively, similarity search happens at a latent space close to the final text representations, making the CIR problem closer to the task of text-to-text retrieval. However, as the proposed model is able to capture fine-grained information by leveraging powerful visual representations, we are able to obtain an impressive retrieval performance. This is expected as the BLIP2 achieves state-of-the-art performance on Visual Question Answering (VQA) benchmarks, showing that image information can be effectively captured.

To adapt the LLM to this task while retaining its knowledge, we applied LoRA [21] to the query and value matrices of all the self-attention and cross-attention layers of the LLM. LoRA [21] learns a residual representation on top of some layers using matrices with low rank. Theoretically, this is supported by the fact that LLMs adapted to a specific task have low intrinsic dimension [1], and in practice it allows training with low computational resources and limited data. Moreover, only modifying a few parameters reduces the risk of catastrophic forgetting, observed in some studies where full fine-tuning of an LLM decreases the performance compared to using it frozen or fine-tuning it with parameter-efficient techniques [23, 31].

The hidden states of the T5 decoder are a sequence of tensors. Instead of using a class-like token as in Koh et al. [23] to summarize the information along the temporal dimension, we perform an average followed by layer normalization [2]. This technique was utilized in EVA [15], which improves over CLIP [36] in several downstream tasks. The result is then projected to the embedding dimension using a ReLU-activated MLP and followed by normalization.

We train the model using a multi-task objective involving the InfoNCE loss [34], a lower bound on the mutual information [27], as retrieval term:

$$\begin{aligned} \mathcal{L}_{\text{InfoNCE}} &:= \mathbb{E}_i \left[\log \frac{\exp\left(S_{i,i} \cdot \tau\right)}{\sum_j \exp\left(S_{i,j} \cdot \tau\right)} \right] \\ S_{i,j} &:= \left\langle \mathcal{F}(\mathbf{x}_{\text{ref}}^{(i)}, \mathbf{t}^{(i)}), \mathcal{G}(\mathbf{x}_{\text{trg}}^{(j)}) \right\rangle , \end{aligned} \tag{4}$$

where τ is a learnable scaling parameter. Practically, given that our model has many parameters, the maximum batch sizes we can achieve have an order of magnitude of hundreds of samples. Given that this can affect the retrieval performance due to a lack of negative samples, we maintain a cross-batch memory as proposed in Wang et al. [47] and use it for the computation of Eq. (4).

On top of that, we add a standard maximum likelihood as a language modeling term $\mathcal{L}_{\text{LM}}$. We compute this objective using teacher forcing [50], based on providing the ground-truth outputs of previous tokens to estimate the next token, and cross-entropy loss. The final loss is

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \omega \mathcal{L}_{\text{InfoNCE}} , \tag{5}$$

where ω is a hyperparameter determining the relative importance of the retrieval task.

3.2. Conversational interface

Inspired by Visual ChatGPT [52], we connect a user chat to a prompt manager that acts as a middle-man to a LLM and provides it with access to tools. Differently from Wu et al. [52], these tools are not only to understand and modify images but also to perform searches with both unimodal and multimodal inputs.

From the user's perspective, the proposed framework allows implicitly using a search tool without requiring any input pattern. For example, interacting with a model like SIMAT [12] could be unintuitive as it requires two words with the original and target attributes. We trained our multimodal search model on Fashion200K [19], which only contains inputs of the form "`replace {original_attribute} with {target_attribute}`" (see Sec. 4.1). We could formulate this prompt using the same inputs that a model like SIMAT requires and thus modify them to match the training distribution of our model.

Since the LLMs can only ingest text information, we add image understanding tools to provide information about the images and their content, as well as search tools:

Image search: Image-only search based on CLIP [36] image embeddings. We use this tool internally when a user uploads an image to show an initial result to users, which

may inspire them to write the follow-up queries. The descriptions of the search results are provided to the LLM to enable Retrieval Augmented Generation (RAG) [26]

Multimodal search: The input of the multimodal search tool is an image and two text strings expressing the original and target attributes. We use our model and feed it the Fashion200K [19] prompt created from these attributes.

VQA model: We use the BLIP [28] pretrained base model[1] to facilitate image understanding to the LLM.

Our approach to providing image information to the LLM is similar to LENS [7], as it is a training-free method applicable to any off-the-shelf LLM.

3.2.1 Workflow

In this section, we describe the main events in the interface and the triggered actions.

Start: When a new user starts a new session, we create a unique identifier used to set a dedicated folder to store images and initialize the memory to store the context. The memory contains a conversation where the lines prefixed with `"Human:"` come from the user, and those starting with `"AI:"` are outputs of the LLM shown to the user.

Image input: When a user uploads an image, we store it in the session folder using file names with sequential numerical identifiers, *i.e.*, `IMG_001.png`, `IMG_002.png`, `IMG_003.png`, *etc*. Then we add a fake conversation to the memory:

```
Human: I provided a figure named {image_filename}. {
     description}
AI: Provide more details if you are not satisfied with
     the results.
```

where `description` is the text output of the search action.

Search: Every time a search tool is used, the results are shown to the user in a carousel of images. Additionally, we add the following information to the memory that will be provided to the LLM once invoked

```
Top-{len(image_descriptions)} results are: {
     image_descriptions}.
```

which contains the descriptions of the top retrieved images. These details help the LLM understand the fine-grained details (*e.g.*, brand, product type, technical specifications, color, *etc*.) and the multimodal search intention. We can interpret this as a form of RAG [26]. RAG is based on using an external knowledge base for retrieving facts to ground LLMs on the most accurate and up-to-date information.

Text input: Every time the user provides some text input, we invoke the LLMthrough the prompt manager. In this stage, the LLM can communicate directly to the user or use special formatting to call some tools. If the LLM wants to perform a multimodal search, it can typically find the target attribute in the text input, which only needs to be formatted and simplified. However, in most cases, the original attribute is not included in the input text as it is implicit in the image. Generally, the descriptions contain enough information to perform the query. Otherwise, the LLM can use the VQA model to ask specific questions about the image.

3.2.2 Prompt manager

The prompt manager implements the workflow described in the previous section and empowers the LLM with access to different tools. The tool calls are coordinated by defining a syntax that processes the output of the LLM and parses the actions and the text visible to the user in the chat.

Every time the LLM is triggered, the prompt manager does so with a prompt that includes a description of the task, formatting instructions, previous interactions, and outputs of the tools.

We crafted a task description that specifies that the LLM can ask follow-up questions to the customers if the search intents are unclear or the query is too broad. In the prompt, we also include examples of use cases written in natural language. The formatting instructions describe when the LLM should use a tool, which are the inputs, how to obtain them, and what are the tool outputs.

For each tool, we have to define a name and a description that may include examples, input and output requirements, or cases where the tool should be used.

In this work, we test two prompt managers:

Langchain [9]: We take the Langchain prompts from Visual ChatGPT [52] and adapt them to our task. The syntax to use a tool is:

```
Thought: Do I need to use a tool? Yes
Action: Multimodal search
Action Input: IMG_001.png;natural;black
```

Our prompt manager: Inspired by the recent success of visual programming [18, 44], we propose to use a syntax similar to calling a function in programming languages:

```
SEARCH(IMG_001.png;natural;black)
```

In Fig. 1, we illustrate an example of a conversation and the actions that the prompt manager and the LLM trigger.

Visual programming typically performs a single call to a LLM, and the output is a single action or a series of actions whose inputs and outputs can be variables defined on the fly by other functions. While Langchain [9] allows performing multiple actions, it requires executing them one at a time. When the LLM expresses the intention to use a tool, Langchain calls the tool and prompts the LLM again with the output of such a tool. The visual programming approach only invokes the LLM once, saving latency and possible costs attributed to API calls. However, in visual programming, the LLM cannot process the output of tools

[1] https://huggingface.co/Salesforce/blip-vqa-base

Table 1. **Quantitative results:** Recall@k on the Fashion200K dataset [19]. Our method is able to successfully fuse image and text information and generate a representation that is useful to caption the resulting image and generate an embedding for retrieval purposes. Best results shown in **boldface**.

Method ↓	R@10	R@50	Average
RN [42]	40.5	62.4	51.4
MRN [22]	40.0	61.9	50.9
FiLM [35]	39.5	61.9	50.7
TIRG [46]	42.5	63.8	53.2
CosMo [25]	50.4	69.3	59.8
FashionVLP [16]	49.9	70.5	60.2
VAL [10]	53.8	73.3	63.6
Ours	**71.4**	**91.6**	**81.5**

but only use their outputs blindly. For the sake of simplicity, we restrict the custom prompt manager to handle single actions, but this could easily be extended following Gupta and Kembhavi [18], Surís et al. [44].

Additionally, we propose to include Chain-of-Thought (COT) [24, 49, 54, 55]. COT is a technique that enforces that the LLM reasons about the actions that should be taken. This simple technique has reportedly found numerous benefits. Following the example above, the complete output expected by the LLM would be as follows:

```
Thought: I can see that human uploaded an image of a
     deep v-neck tee. From the results, the color of the
      tee is natural. The user wants the color to be
     black instead. I have to call search.
Action: SEARCH(IMG_001.png;natural;black)
```

While Langchain and our prompt manager use the special prefix `"Thought"` to handle certain parts of the query, their purposes are distinct. In Langchain, the prefix is used to parse lines in the LLM output. If a line starts with this prefix, Langchain expects to find the question `"Do I need to use a tool?"` followed by `"Yes"` or `"No"`, indicating whether a tool should be used. In contrast, our novel prompt manager does not impose any specific format on lines starting with the `"Thought"` prefix. Instead, these lines are solely dedicated to incorporating COT reasoning.

4. Experiments

4.1. Multimodal search on Fashion200K

Implementation details: We use the Flan T5 XL model [11], which is a 3 billion parameter LLM from the T5 family [37], finetuned using instruction tuning [48]. We obtain the visual features with CLIP-L model [36], a model with patch size 14 and 428 million parameter. In total, the model has around 3.5 million parameter, which requires splitting the model across different GPUs for training. Concretely, we use 8 NVIDIA V100 GPUs.

LoRA is performed with a rank of $r = 16$, scaling $\alpha = 32$ and dropout of 0.5 on the query and value matrices of the attention layers of the LLM. The hidden representation obtained from the LLM is transformed with a linear layer of size 1024, passed through a ReLU activation, and then transformed with another linear layer that yields an embedding of size 768. Such an embedding is normalized to have unit norm and used for retrieval.

We optimize the model with AdamW [33] with a learning rate of 10^{-5} and weight decay of 0.5 for a total of 300 epochs. The learning rate is linearly increased from 0 to the initial learning rate during the first 1000 steps. We set the weight of the language modeling loss as $\omega = 1$. The effective batch size considering all the GPUs is 4,096, and the total number of embeddings included in the cross-batch memory Wang et al. [47] is 65,536.

Dataset: The Fashion200K [19] is a large-scale fashion dataset crawled from online shopping websites. The dataset contains over 200,000 images with paired product descriptions and attributes. All descriptions are fashion-specific and have more than four words, *e.g.*, "Beige v-neck bell-sleeve top". Similarly to Vo et al. [46], text queries for the CIR problem are generated by comparing the attributes of different images and finding pairs with one attribute difference. Then, a query is formed as `"replace {original_attribute} with {target_attribute}"`.

When trained on Fashion-200K [19], our method achieves state-of-the-art results, improving the retrieval performance of competitive methods by 20% recall at positions 10 and 50. Tab. 1 includes the comparison with some of the CIR methods reviewed in Sec. 2 [10, 16, 25, 46], as well as the visual reasoning-based baselines RN [42], MRN [22], and FiLM [35].

One of the reasons is that the model can exploit the image and text understanding prior of a foundational model that can perform image captioning, and adapt it for the related task of *composed captioning*. The hidden representations of the model contain enough information to describe the target image and are effectively used for that purpose. Adapting to this new task becomes easier given the specific formatting of the modifying text, which facilitates extracting the important parts of the query.

The results show that it is possible to distill knowledge from a large vision and language model trained on large-scale datasets. While our model has billions of parameters, which is far more than the other models, we are able to learn a new task similar to the ones that the pretrained model could solve with only learning a few parameters consisting of a very small percentage of the total model size.

We include some qualitative examples in Fig. 3. These show that our model can successfully incorporate text information and modify the internal description formed about

(a) Successful examples

(b) Failure examples

Figure 3. **Qualitative results**: Examples of queries of the Fashion-200k dataset [19] and the 4 best matches. The correct matches are shown in **green** and incorrect ones in **red**. In the succesful examples, we can see that our proposal is able to incorporate modifications to the input product involving changes to color and material among others. Despite not retrieving the correct products in the failure examples, almost all the retrieved images satisfy the search criteria.

the input image. The successful results in Fig. 3a show that the proposed model retrieves visually similar and can incorporate modifications of different attributes such as the color and the material.

The failures in Fig. 3b show that all the first retrieve results satisfy the search criteria, with some of them even belonging the same product. This hints at our model having an even better performance in practice than what the benchmark reflects.

Overall, we can see from all the qualitative examples that all the top-ranked results are relevant. The only exception is the inclusion of the reference image, which is a common error in retrieval systems given that the search embedding is computed from such an image.

4.2. Search interface

One of the key drivers of performance is based on reformulating the examples. While the examples in Langchain are written using natural language, we advocate for using LLM model instructions. In this sense, the examples contain exactly the input that the LLM would receive including the product type, top-k product titles, and user input. Such examples also contain the expected model output including the COT reasoning and the action itself. This reinforces the format instructions and the benefits of RAG.

Note that the proposed reformulation introduces some redundancy w.r.t. the Langchain formatting instructions. Additionally, it requires to allocate much more space for examples. Despite these considerations, we find our approach beneficial. For a fair comparison, we also limit the full prompt to fit the context of the smallest LLM and empirically find that allocating more space to examples is beneficial even if this is at the cost of removing the prefix.

We tested different LLMs for the search interface. Among all, GPT-3 [8], concretely the `text-davinci-003` model, was empirically found to be the best performing. Fig. 4 shows an example of our conversational interface displaying a real example in which composed retrieval is performed.

Besides GPT-3 [8], we compared different open-source models from the `transformers` library [51]. Surprisingly, these models performed poorly. Digging into the outputs of the LLMs we could see that one of the failure cases of Fastchat [56] had the following output:

```
Thought: Do I need to use a tool? Yes
Action: Multimodal altering search
Action Input: image_file: IMG_001.png, attribute value
     of the product in the image: chair, desired
     attribute value: sofa
```

While the former contains the correct action to take and the correct inputs, *i.e.* image file, negative text and positive text, it is not correctly formatted for Langchain. Instead, GPT-3 [8] is able to generate a correctly formatted output:

```
Thought: Do I need to use a tool? Yes
Action: Multimodal altering search
Action Input: IMG_001.png;chair;sofa
```

This example shows that FastChat [56] has the knowledge to perform a successful query but struggles to use the complicated formatting of Langchain. This example is the main motivation why we developed the novel prompt manager in Sec. 3.2.2.

5. Limitations

The model in Sec. 3.1 can achieve an impressive performance on Fashion200K. As discussed in Sec. 4.1, the char-

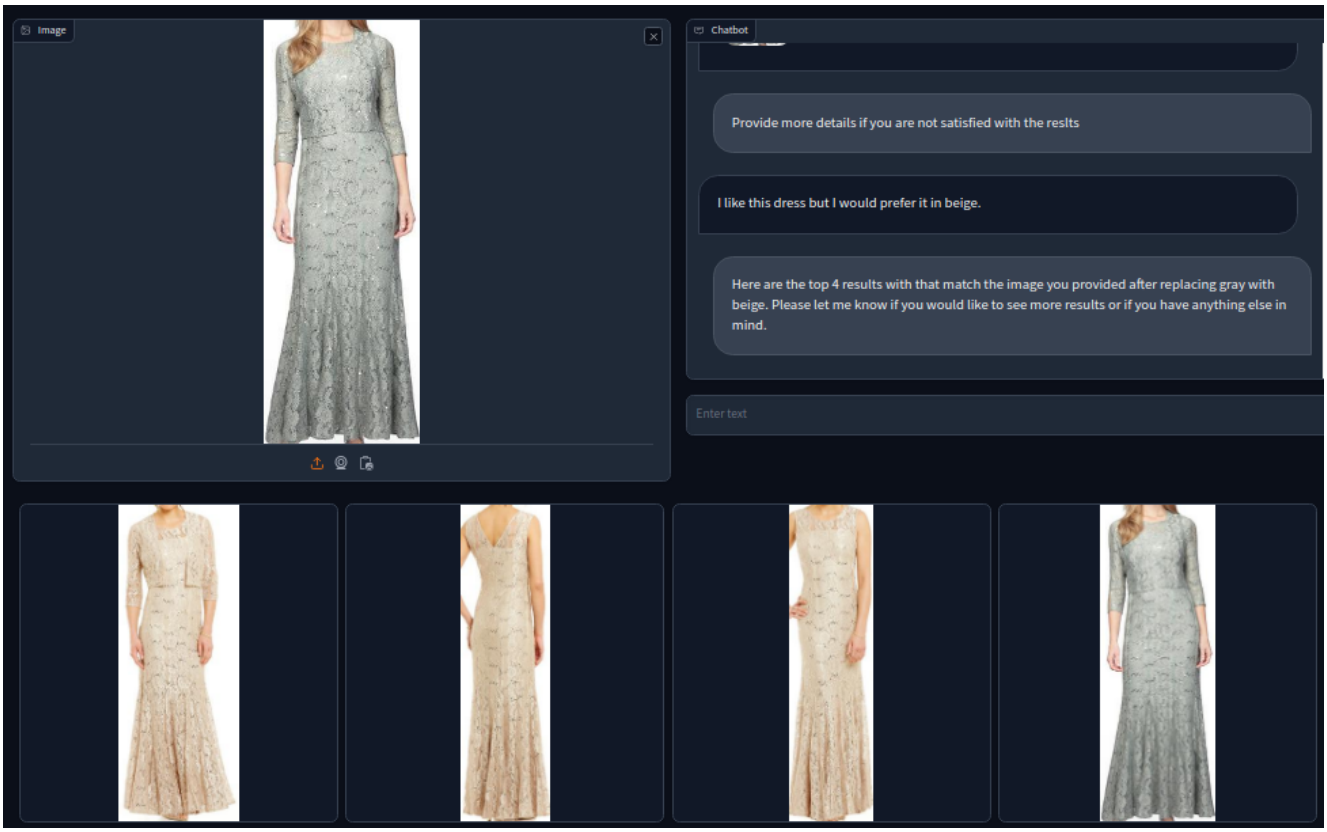

Figure 4. **Proposed conversational multimodal search system**: In this example, the user uploads an image from the Fashion200K dataset [19] and provides text input intending to search an a dress similar to the product in the image but in a different color. An LLM, specifically GPT-3 [8], processes the user's prompt and invokes our novel multimodal search model with the uploaded image and a formatted text query. The desired attribute indicated by the user is "beige", which can be inferred from the text input. The original attribute is required by the prompt used during the training of our model and is correctly identified by the LLM as "gray". In this case, the LLM can obtain this information leveraging the RAG based on obtaining the product descriptions of the first matches using image search with the uploaded picture. The conversational nature of the interactions with the user offers an improved search experience.

acteristics of this dataset are ideal for our model to excel but may hinder generalizing to natural language queries. This is solved with our conversational interface, but the current setup is restricted to modifying a single attribute at a time.

Using hard prompts to encode the task description is simple and applicable to black-box models such as LLMs accessed through an API. However, it reduces the effective context length of LLMs and requires prompt engineering, which is a tedious process.

Although LLMs have a large context size, the prompt yields an effective input size that is relatively small, and the memory rapidly fills up. In practice, the memory gets truncated if conversations are too long, hence discarding the first interactions.

6. Conclusions

This paper presents a comprehensive pipeline to perform image retrieval with text modifications, addressing the CIR problem. Our novel composed retrieval model, built upon the BLIP2 architecture [28] and leveraging LLMs, has demonstrated superior performance on the Fashion200K dataset [19] compared to previous models.

In this work, we also describe the integration of LLMs into a search interface, offering a conversational search assistant experience that enhances user interaction. We implement a prompt manager to enable using small LLMs and incorporate the COT [24, 49] and RAG [26] techniques to improve system performance.

Our experiments underscore the importance of addressing inherent challenges in multimodal search, including enhancing matching capabilities and handling ambiguous natural language queries.

Acknowledgments

The authors acknowledge René Vidal for constructive discussions. O.B. is part of project SGR 00514, supported by Departament de Recerca i Universitats de la Generalitat de Catalunya.

References

[1] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *International Joint Conference on Natural Language Processing*, 2021. 4

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv:1607.06450, 2016. 4

[3] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Conditioned and composed image retrieval combining and partially fine-tuning CLIP-based features. In *CVPRW*, New Orleans, LA, USA, 2022. 2

[4] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Effective conditioned and composed image retrieval combining CLIP-based features. In *CVPR*, 2022. 2

[5] Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Alberto Del Bimbo. Zero-Shot Composed Image Retrieval with Textual Inversion. In *ICCV*, 2023. 2

[6] Aurélien Bellet, Amaury Habrard, and Marc Sebban. *Metric Learning*. Morgan & Claypool Publishers (USA), Synthesis Lectures on Artificial Intelligence and Machine Learning, pp 1-151, 2015. 3

[7] William Berrios, Gautam Mittal, Tristan Thrush, Douwe Kiela, and Amanpreet Singh. Towards Language Models That Can See: Computer Vision Through the LENS of Natural Language. arXiv:2306.16410, 2023. 5

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *NeurIPS*, 2020. 7, 8

[9] Harrison Chase. LangChain. https://github.com/langchain-ai/langchain, 2022. 5

[10] Yanbei Chen, Shaogang Gong, and Loris Bazzani. Image search with text feedback by visiolinguistic attention learning. In *CVPR*, 2020. 2, 6

[11] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. arXiv:2210.11416, 2022. 6

[12] Guillaume Couairon, Matthijs Douze, Matthieu Cord, and Holger Schwenk. Embedding Arithmetic of Multimodal Queries for Image Retrieval. In *CVPRW*, 2022. 2, 4

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3

[15] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023. 4

[16] Sonam Goenka, Zhaoheng Zheng, Ayush Jaiswal, Rakesh Chada, Yue Wu, Varsha Hedau, and Pradeep Natarajan. FashionVLP: Vision Language Transformer for Fashion Retrieval with Feedback. In *CVPR*, 2022. 2, 6

[17] Geonmo Gu, Sanghyuk Chun, Wonjae Kim, HeeJae Jun, Yoohoon Kang, and Sangdoo Yun. CompoDiff: Versatile Composed Image Retrieval With Latent Diffusion. arXiv:2303.11916, 2023. 2, 3

[18] Tanmay Gupta and Aniruddha Kembhavi. Visual Programming: Compositional Visual Reasoning Without Training. In *CVPR*, 2023. 5, 6

[19] Xintong Han, Zuxuan Wu, Phoenix X. Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S. Davis. Automatic spatially-aware fashion concept discovery. In *ICCV*, 2017. 1, 2, 4, 5, 6, 7, 8

[20] Yunpeng Han, Lisai Zhang, Qingcai Chen, Zhijian Chen, Zhonghua Li, Jianxin Yang, and Zhao Cao. Fashionsap: Symbols and attributes prompt for fine-grained fashion vision-language pre-training. In *CVPR*, 2023. 2

[21] 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, 2021. arXiv:2106.09685 [cs]. 3, 4

[22] Jin-Hwa Kim, Sang-Woo Lee, Donghyun Kwak, Min-Oh Heo, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Multimodal residual learning for visual qa. In *NeurIPS*, 2016. 6

[23] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding Language Models to Images for Multimodal Inputs and Outputs. In *ICML*, 2023. 2, 4

[24] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners. In *NeurIPS*, 2022. 6, 8

[25] Seungmin Lee, Dongwan Kim, and Bohyung Han. Cosmo: Content-style modulation for image retrieval with text feedback. In *CVPR*, 2021. 2, 6

[26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *NeurIPS*, 2020. 5, 8

[27] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 4

[28] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified

vision-language understanding and generation. In *ICML*, 2022. 5, 8

[29] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models, 2023. 3

[30] Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation, 2021. arXiv:2101.00190 [cs]. 4

[31] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *NeurIPS*, 2022. 4

[32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. arXiv:2304.08485, 2023. 2

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv:1807.03748, 2018. 4

[35] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 6

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 2, 3, 4, 6

[37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 2, 3, 4, 6

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752, 2021. 2

[39] Negar Rostamzadeh, Seyedarian Hosseini, Thomas Boquet, Wojciech Stokowiec, Ying Zhang, Christian Jauvin, and Chris Pal. Fashion-gen: The generative fashion dataset and challenge. arXiv:1806.08317, 2018. 2

[40] Kuniaki Saito, Kihyuk Sohn, Xiang Zhang, Chun-Liang Li, Chen-Yu Lee, Kate Saenko, and Tomas Pfister. Pic2Word: Mapping Pictures to Words for Zero-shot Composed Image Retrieval. In *CVPR*, 2023. 2

[41] Artsiom Sanakoyeu, Vadim Tschernezki, Uta Büchler, and Björn Ommer. Divide and conquer the embedding space for metric learning. In *CVPR*, 2019. 3

[42] Adam Santoro, David Raposo, David G.T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *NeurIPS*, 2017. 6

[43] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 3

[44] Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual Inference via Python Execution for Reasoning. arXiv:2303.08128, 2023. 5, 6

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[46] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing Text and Image for Image Retrieval - an Empirical Odyssey. In *CVPR*, 2019. 2, 6

[47] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *CVPR*, 2020. 4, 6

[48] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. 6

[49] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*, 2022. 6, 8

[50] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1989. 4

[51] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020. 7

[52] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models. arXiv:2303.04671, 2023. 4, 5

[53] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, 2017. 3

[54] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic Chain of Thought Prompting in Large Language Models. In *ICLR*, 2023. 6

[55] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal Chain-of-Thought Reasoning in Language Models. arXiv:2302.00923, 2023. 6

[56] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. arXiv:2306.05685, 2023. 7