Language-only Efficient Training of Zero-shot Composed Image Retrieval

Geonmo Gu*, [1] Sanghyuk Chun*, [2] Wonjae Kim[2] Yoohoon Kang[1] Sangdoo Yun[2]

[1]NAVER Vision [2]NAVER AI Lab * Equal contribution

Abstract

*Composed image retrieval (CIR) task takes a composed query of image and text, aiming to search relative images for both conditions. Conventional CIR approaches need a training dataset composed of triplets of query image, query text, and target image, which is very expensive to collect. Several recent works have worked on the zero-shot (ZS) CIR paradigm to tackle the issue without using pre-collected triplets. However, the existing ZS-CIR methods show limited backbone scalability and generalizability due to the lack of diversity of the input texts during training. We propose a novel CIR framework, only using language for its training. Our LinCIR (**Lin**guage-only tra**in**ing for CIR) can be trained only with text datasets by a novel self-supervision named self-masking projection (SMP). We project the text latent embedding to the token embedding space and construct a new text by replacing the keyword tokens of the original text. Then, we let the new and original texts have the same latent embedding vector. With this simple strategy, LinCIR is surprisingly efficient and highly effective; LinCIR with CLIP ViT-G backbone is trained in 48 minutes and shows the best ZS-CIR performances on four different CIR benchmarks, CIRCO, GeneCIS, FashionIQ, and CIRR, even outperforming supervised method on FashionIQ. Code is available at* `github.com/navervision/lincir`

1. Introduction

Composed image retrieval (CIR) is a challenging vision-language (VL) task that takes a composed query of image and text, aiming to search relative images for both conditions [29]. As language serves as the most natural method for encoding human interaction, CIR provides a higher degree of freedom and a better user experience for image-based search engine applications, such as web commerce.

One of the main challenges of CIR is the expensive dataset collection pipeline. CIR datasets consist of triplets $\langle x_{i_R}, x_c, x_i \rangle$, where x_{i_R} is an image query, x_c is a text query, and x_i is the target image. Unlike image-text paired datasets, such as CC3M [39] or LAION [38], such triplets are almost impossible to collect by web crawling and require expensive human labor to create each triplet. For example, the datasets are constructed by gathering candidates of $\langle x_{i_R}, x_i \rangle$ and manually annotating x_c by human annotators [29, 47], which is hard to scalable. Therefore, the size of the training triplets is usually small (*e.g.*, 46.6k [47], 28.8k [29]), and the existing CIR methods trained on such triplets [2, 6–8, 11, 12, 20–22, 24, 30, 40, 42, 46, 48] suffer from the lack of generalizability to diverse unseen domains.

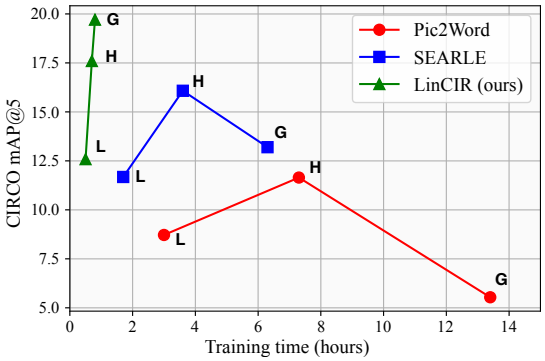

Figure 1. **Training time (hours) vs. Zero-shot Composed Image Retrieval (ZS-CIR) performance.** Thanks to our efficient language-only training strategy, our LinCIR outperforms the previous ZS-CIR methods in both training time and CIR performance. The training time is measured on 8 A100 GPUs. We compare the models on the CIRCO mAP@5 [3] score for a more comprehensive evaluation of CIR models (more results are in Fig. 4). Notably, when we scale up the backbone CLIP [19, 35] model size by ViT-L, ViT-H and ViT-G, LinCIR shows a promising performance boost with surprisingly short training time (48 mins for ViT-G). On the other hand, Pic2Word [37] and SEARLE [3] cannot be scaled up to CLIP ViT-G due to their limitation on restricted textual expressions and the lack of diversity of input texts.

To overcome the drawback, recent studies have explored zero-shot CIR (ZS-CIR), a scalable direction, by eliminating the dependency on the pre-collected triplet datasets. For example, Saito et al. [37] and Baldrati et al. [3] propose projection-based ZS-CIR methods without using triplet datasets. Based on the pre-trained CLIP [35], they train a lightweight projection module ϕ that projects the CLIP image latent embedding z_i to the CLIP text token

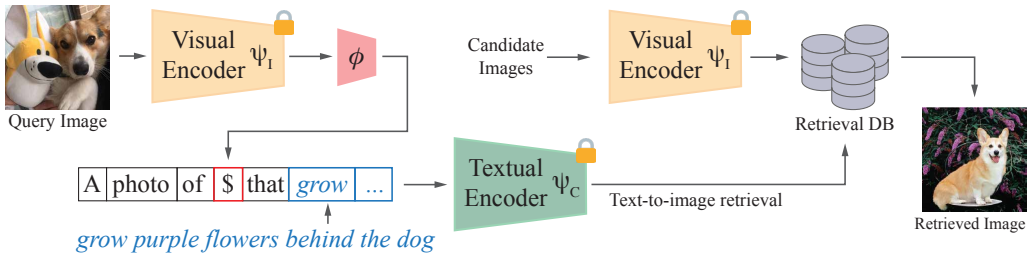

Figure 2. **Overview of ZS-CIR with a projection to the token embedding space.** The mainstream ZS-CIR methods, such as Pic2Word [37], SEARLE [3] and LinCIR (ours), train a projection module ϕ that projects the image latent embedding z_i into the token embedding space e_c with a custom prompt (*e.g.*, a photo of `[$]` that `[cond]`). The textual encoder output is used for CIR.

embedding space e_c (see Fig. 2). These approaches have shown promising generalizability to unseen datasets. However, these methods struggle to handle diverse text conditions because they rely on pre-defined naïve text prompts during training (*e.g.*, a photo of `[$]`). Moreover, their training frameworks need sequential forward operations of the visual and textual encoders (see Fig. 3 (a)), resulting in inefficient training and limited scalability to a larger backbone.

In this paper, we introduce a new paradigm of ZS-CIR, named **L**anguage Only tra**in**ing for **C**omposed **I**mage **R**etrieval (LinCIR)[1]. As shown in Fig. 3 (b), instead of projecting the image latent embedding z_i, we propose to project the text latent embedding z_c to the token embedding space e_c. We introduce a novel self-supervision, named Self-Masking Projection (SMP), for language-only training. We replace all the "keywords" of the original text with the projected text embedding of the original text to produce an embedding $\hat{z}_c$ and apply MSE loss between z_c and $\hat{z}_c$. Here, we define "keyword" as consecutive adjectives and nouns. For example, the keywords of "gray cat sleeps on a pillow" are "gray cat" and "a pillow"; therefore, it becomes "`[$]` sleeps on `[$]`". The purpose of SMP is to make `[$]` token interpreted as the "one-word summarization" of the input by extracting the essential information of the input. During inference, we simply perform text-to-image retrieval by projecting image embedding to the token embedding space using the projection module ϕ as shown in Fig. 2. However, this strategy can suffer from the modality gap between textual and visual modalities [27], *i.e.*, even though our ϕ module works perfectly for text latent embeddings, it can underperform for the target visual embeddings. We mitigate the issue by employing a random noise addition strategy [33], carefully choosing a probability distribution that ensures the diversity of the noise-augmented textual embeddings.

Our paradigm has three advantages over the previous approaches. First, while Pic2Word and SEARLE projection modules are trained with a restricted text prompt, *i.e.*, a photo of `[$]`, the projection module of LinCIR is trained with the diverse text inputs from the actual texts. Due to this reason, Pic2Word and SEARLE show degenerated performances when the backbone size becomes larger (see Fig. 1). On the other hand, LinCIR is more generalizable to complex and diverse text conditions, showing superior ZS-CIR performances than others, especially for larger backbones. Second, as LinCIR only utilizes the textual encoder, our training process is highly efficient and scalable than the methods incorporated with the visual encoder; our language-only training strategy is $\times$6.0 faster than Pic2Word [37] and $\times$8.4 faster than SEARLE [3] with CLIP ViT-L backbone. When we scale up the backbone size to CLIP ViT-G, the gap becomes $\times$16.4 and $\times$17.6, respectively. Even ViT-G training for LinCIR only takes 48 minutes using 8 A100 and less than 2 hours using 1 V100. Third, our method is storage-efficient; for example, the CC3M dataset [39] images occupy about 430GB storage size, while its captions only need 125MB. We train LinCIR only with 571MB storage size for storing the 5.5M training captions. In summary, LinCIR shows the best training time and ZS-CIR performance as shown in Fig. 1 and 4.

Our contribution can be summarized as follows: (1) We propose LinCIR, a novel and efficient language-only training framework for ZS-CIR. (2) We introduce a new self-supervision for language-only training, named Self-Masking Projection (SMP). (3) We employ a better random noise addition strategy than naïve Gaussian noise to mitigate the modality gap. (4) LinCIR achieves the best training time and the ZS-CIR performances on four ZS-CIR benchmarks (CIRCO [3], GeneCIS [44], FashionIQ [47] and CIRR [29]). Notably, LinCIR even outperforms the state-of-the-art supervised method [2] on FashionIQ.

2. Preliminaries

Vision-langauge models (VLM). As with previous ZS-CIR methods, LinCIR utilizes a pre-trained VLM, such as CLIP [19, 35] or BLIP [25]. We use VLMs that map an image input x_i and a text input x_c to the d-dimensional joint embedding space. Here, a given caption x_c is tokenized by the pre-defined tokenizer as $t_c = \{t_c^k \mid k = 1 \cdots K\}$, where K is the number of tokens, and mapped to *token embeddings* $e_c = E_w(t_c) = \{E_w(t_c^k) \mid k = 1 \cdots K\}$, where E_w is the embedding layer parameterized by w. Afterwords, a

[1]Pronounced as "linker", meaning for linking the two modalities.

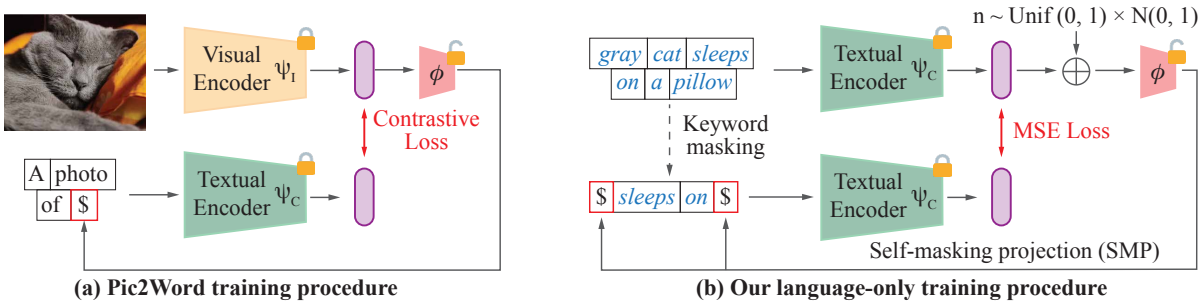

Figure 3. **Comparison of Pic2Word [37] and LinCIR training procedures.** (a) Pic2Word [37] and SEARLE [3] training procedure requires both the visual encoder and the textual encoder. They only need images for training, while the text prompt is pre-defined [37] or automatically generated [3]. (b) LinCIR is trained solely on texts with the frozen textual encoder. First, a projection module ϕ projects a textual latent embedding of a sentence z_t into the token embedding space. Before the projection, a random noise n is added to z_t to reduce the modality gap between text and image. We introduce a new self-supervision, named Self-Masking Projection (SMP), by replacing all keywords of the given caption with the projected embedding by ϕ and extracting a modified text embedding $\widehat{z}_t$. Finally, the projection module ϕ is trained by the MSE loss between z_t and $\widehat{z}_t$. Note that both (a) and (b) use the same inference strategy shown in Fig. 2.

textual encoder ψ_C encodes the token embeddings to extract *textual latent embeddings* $z_c = \psi_C(x_c) = \psi_C(E_w(t_c))$. The *visual latent embeddings* are extracted by the visual encoder ψ_I: $z_i = \psi_I(x_i)$. We will use the terminology *token embeddings* to represent e_c, and use *textual (or visual) latent embeddings* to represent z_c (or z_i).

VLM modality gap. The VLM joint embedding space suffers from the semantic gap between each modality. Namely, the visual and the textual latent embeddings are not exactly aligned with each other, but they are located in completely separate regions of the embedding space [27].

Such a modality gap hinders the harmonization of visual and textual latents in the joint embedding space. To mitigate the gap, language-only training has been recently introduced for image captioning tasks [16, 26, 33]. They train a text decoder from the CLIP textual embeddings to generate image captions from visual embeddings. During training the decoder, the modality gap is bridged by injecting Gaussian noises [16, 33] or projection-based method [26].

In this paper, we utilize language-only training and a noise-addition strategy to mitigate the modality gap. We carefully studied the impact of noise and found a better distribution rather than simple Gaussian noise. We believe our findings can be transferred to other language-only training methods. Moreover, our target task, CIR, needs to model a relationship between triplets of $\langle x_{i_R}, x_c, x_i \rangle$, whereas previous language-only training methods only consider the pair-wise relationship (*i.e.*, image captioning). We tackle this problem by proposing a novel self-supervision, named Self-Masking Projection (SMP), which injects the projected textual embeddings into the original token embeddings.

CIR by projection to token embeddings. The mainstream ZS-CIR methods, such as Pic2Word [37] and SEARLE [3], employ a projection-based method. Namely, they learn a projection module ϕ from the image latent embedding space to the token embedding space. For inference, they project the image latent embedding z_i to the token embedding (`$`), then perform text-to-image retrieval with the prompt "a photo of `[$]` that `[cond]`", where `[cond]` is a text condition. Fig. 2 shows an overview of how textual projection-based ZS-CIR works. The main research point of this field is how to train the projection module ϕ to capture the visual information into the token embedding space.

Pic2Word [37] trains the projection module ϕ by minimizing contrastive loss between image latent embedding and the textual latent embedding of "a photo of `[$]`" (see Fig. 3 (a)). SEARLE [3] employs a similar approach to Pic2Word. First, they employ optimization-based textual inversion to generate pre-defined special tokens for an image and train ϕ to predict the token embedding. SEARLE employs CLIP zero-shot classification to predict the "concept" of the given image and refine the prompt by letting GPT [4] continue the phrase. Both methods only use image inputs x_i for training without accessing the CIR triplets.

Although Pic2Word and SEARLE achieve reasonable ZS-CIR performances, they have two significant problems. First, they heavily rely on the initial prompt, "a photo of `[$]`", limiting the diversity of the textual encoder input. As Fig. 1, we argue that diversifying the input texts during training the projection module ϕ is critical to train with larger backbones (*e.g.*, CLIP ViT-G), while using the naïve prompt is failed to scale up the backbone size. Second, they need image inputs, which are less compact and redundant than text datasets. Moreover, the visual encoder usually needs more computation resources than the textual encoder because the visual encoder takes the fixed length token (*e.g.*, 256). In contrast, the textual encoder takes shorter token lengths (*e.g.*, average token length of CC3M ≈10). We tackle the problems by introducing a language-only training method, showing remarkable efficiency and scalability.

3. Language-only Training of Zero-shot CIR

This section introduces a new paradigm for ZS-CIR, named Language Only training for Composed Image Retrieval (LinCIR). We first introduce a novel language-only self-supervision, Self-Masking Projection (SMP), that enables a language-only training for CIR (§3.1). Then, we explain the modality gap problem and empirically show that adding a carefully chosen random noise can mitigate the problem (§3.2). Finally, we describe the advantage of LinCIR in terms of efficiency and scalability (§3.3).

3.1. Self-Masking Projection (SMP)

We aim to learn a projection module ϕ that captures and retains the original visual information after the projection and the textual encoder. While the previous methods focus on directly mapping visual information to the token space with a naïve prompt (*i.e.*, a photo of `[$]`), we argue that focusing on the textual encoder is more important. Our zero-shot CIR is based on text-to-image retrieval (see Fig. 2). It means that the quality of the textual latent embedding is more critical to the final ZS-CIR performances. Hence, rather than focusing on minimizing the gap between visual information and the naïve prompt, we aim to achieve a projection module ϕ that captures the semantics of the keywords in the given text.

To achieve our goal, we introduce a novel language-only self-supervision named Self-Masking Projection (SMP). First, we project the textual embedding z_c of a given text input x_c with the projection module ϕ to the token embedding space, *i.e.*, $\widehat{e}_c = \phi(z_c)$, where $\widehat{e}_c$ is the projected textual embedding. Then, we replace the token embeddings of all the "keywords" of x_c with the projected token embedding $\widehat{e}_c$. We define the keywords of the sentence as consecutive nouns and adjectives. For example, the keywords of "A Russian Blue cat is gray and cute" will be "A Russian Blue cat", "gray" and "cute"; hence it will be converted to "`[$]` is `[$]` and `[$]`", where `[$]` is a special token to represent the projected token embedding $\widehat{e}_c$. By treating all the main concepts (keywords) in the caption as the same `[$]`, we intend `[$]` to represent the overall essential information of the inputs. Note that similar to `[MASK]` tokens of masked modeling, `[$]` with different positions will be encoded in different features due to the positional embeddings. Using the converted caption, we extract a converted textual latent feature $\widehat{z}_c$ and minimize MSE loss between the original textual embedding z_c and $\widehat{z}_c$. Note that we only train the ϕ module while keeping the textual encoder frozen.

The intuition behind SMP is that semantic information is not balanced across the tokens, but concentrated on the specific keywords. We assume that it is more common that adjectives and nouns in the sentence are more important than other part-of-speechs (POS), such as verbs or adverbs. We empirically observe that our design choice (replacing all keyword token embeddings with $\widehat{e}_c$) is the best among the other variants, such as randomly replacing n keyword tokens ($n = 1, 3, 5$), replacing a random token, replacing all non-keyword tokens, or replacing all noun tokens (See Tab. 8).

SMP has two benefits over the previous image-based ZS-CIR supervision [3, 37]. First, SMP allows the textual encoder to accept more diverse captions rather than "a photo of `[$]`". While previous methods risk being sensitive to natural sentence variations, potentially affecting performances, our approach replaces tokens in natural sentences, maintaining robust performance across a more diverse set of sentence constructions. Second, SMP only requires language inputs; therefore, the overall training procedure is efficient regarding the training time and the storage size. It means that LinCIR can easily scale up in terms of the backbone size and the dataset scale. We will discuss the efficiency and the scalability of LinCIR in the Sec. 3.3.

	No noise	Student-t	Exp	χ^2	$\mathcal{N}(0,1)$	Unif(-1,1)	Ours
L	0.81 (19.8)	0.76 (23.1)	0.76 (23.5)	0.78 (23.5)	0.77 (23.7)	0.74 (25.1)	**0.71 (25.5)**
H	0.63 (31.8)	0.59 (32.4)	0.67 (28.3)	0.64 (27.0)	0.60 (32.8)	0.59 (33.9)	**0.53 (34.8)**
G	0.51 (33.3)	0.55 (36.1)	0.63 (30.8)	0.56 (30.7)	0.55 (35.9)	0.58 (35.3)	**0.48 (36.9)**

Table 1. **Modality gap vs. distributions.** Modality gap [27] (lower denotes less gap) on CC3M and CIRR dev R@1 (higher denotes better performance – in the parentheses) for different noises with different backbone sizes (from ViT-L/14 to ViT-G/14).

3.2. Searching for a better noise distribution for reducing the modality gap.

Although SMP enables the language-only training, we still suffer from the modality gap between textual and visual embedding space [27]. Namely, even if the ϕ module works perfectly for language inputs, it can fail to be generalized to visual inputs. To tackle the problem, we employ a simple noise addition strategy following Nukrai et al. [33]: we add a random noise before the projection during training. Nukrai et al. [33] employed a simple Gaussian noise, but we empirically observe that Gaussian noise is not effective in mitigating the gap. Tab. 1 shows the modality gap [27] measured by various CLIP backbones by adding different probability distributions to textual embeddings. In the table, we observe that the careful choice of distribution greatly affects the modality gap and the final performances.

We also observe that the generally used probabilistic distributions can suffer from a curse of dimensionality in the CLIP embedding space dimensions (*e.g.*, 768-dim). The norm histogram of each probabilistic distribution (Fig. B.2) shows that the samples drawn from a Gaussian distribution have almost identical norm sizes. From this observation, we employ a probability distribution enforcing the diverse norm sizes instead of the Gaussian distribution. We multiply a random scalar value by a random vector drawn from a

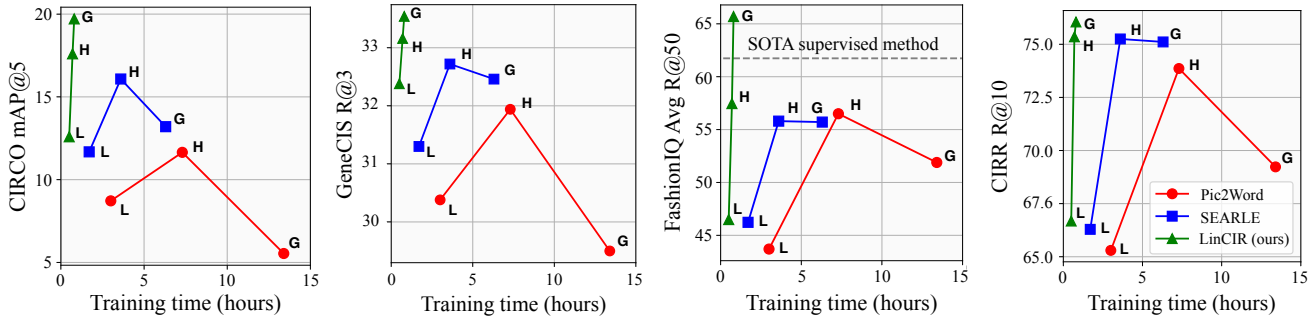

Figure 4. **Training time vs. CIR performances.** We evaluate three CIR methods with three backbone sizes: ViT-L, ViT-H, and ViT-G. To avoid an unreliable assessment due to the nature of R@1, CIR performances are measured in CIRCO mAP@5 [3], GeneCIS average R@3 [44], FashionIQ Average R@50 [47], and CIRR average R@10 [29]. In all evaluation results, LinCIR achieves the best training time-performance trade-off. Moreover, Pic2Word and SEARLE show degenerated performances when scaling up the backbone size.

	Visual encoder				Textual encoder			
	# layers	d	# tokens	TP	# layers	d	# tokens	TP
ViT-L	24	1024	256	18.5	12	768	12.56	65.5
ViT-H	32	1280	256	16.3	24	1024	12.56	35.5
ViT-G	48	1664	256	11.8	32	1280	12.56	26.5

Table 2. **Configuration of CLIP visual and textual encoders.** Every visual encoder uses the patch size 14 and the input resolution 224×224. The text token length is the average token length of the CC3M [39] captions by the CLIP tokenizer. d denotes the hidden dimension of Transformer blocks, and TP denotes throughput per second (higher means faster). TPs are measured by 1 V100 on CC3M [39] captions and COCO [28] images with FP16 weights.

Gaussian distribution, *i.e.*, $n \sim \text{Unif}(0,1) \times \mathcal{N}(0,1)$. Conceptually, our distribution is a randomized Gaussian distribution with varying variances. In the Appendix, we illustrate that our design choice shows a more diverse norm distribution than the other distributions (See Fig. B.2).

In our experiments, we empirically observe that the noise addition strategy improves the overall CIR performances with a large gap by bridging the modality gap. We also empirically observe that our design choice outperforms other probability distributions with a large gap by diversifying the impact of the random noise (see Tab. 9).

3.3. Efficiency and scalability

The most remarkable advantages of LinCIR beyond its generalizability are the training efficiency and scalability. First, a text dataset is storage-efficient; the caption storage size of CC3M dataset [39] is only 125 MB, while its image storage size is about 430GB, about 3,400 times larger. Second, the forward complexity of the textual encoder is notably lower than that of the visual encoder. For example, as shown in Tab. 2, the textual encoder has fewer depth, dimension size, and input token size than the visual encoder taking 224 × 224 resolution images. As a result, the average inference time of the CLIP ViT-L visual encoder is ×3.5 times slower than that of the textual encoder (Tab. 2). Furthermore, the average throughput of the ViT-G textual encoder is even ×1.4 times faster than the ViT-L visual encoder.

All these advantages make LinCIR easily scalable. Even though we increase the backbone size, the overhead of the textual encoder is not significantly increased. We can train LinCIR with the CLIP ViT-G backbone in 48 minutes using 8 A100 GPUs and 2 hours using a single V100.

4. Experiments

4.1. Implementation details

We use three-layered MLP for the ϕ model: `LN` [1] - `Linear` - `GeLU` [18] - `Linear` - `GeLU` - `Linear` - `LN`. The intermediate hidden dimension is set to $4d$ (d for each architecture is shown in Tab. 2). We do not apply the ℓ_2-normalization to the textual encoder outputs during training because, as shown in Fig. B.2, the added random noises have larger norm sizes than 1. If we apply the ℓ_2-normalization, we observe that the ϕ module is not converged. Keywords of the given text are extracted by the POS tagger of `spacy` library. We use the AdamW optimizer [31] with a fixed learning rate of 0.0001, weight decay of 0.01, and mini-batch size of 512. Dropout with probability 50% is applied for the regularization. We use CC3M [39] captions and 2.47M number of the curated StableDiffusion prompts[2] for the training dataset (*i.e.*, there are 5.5M training captions).

For a fair comparison between models, we select the model showing the best zero-shot CIRR [29] dev R@1 score for the model selection. We employ an early stopping strategy by monitoring the validation score. We evaluate the CIR performances of the selected model in a zero-shot manner, *i.e.*, one model is evaluated on four benchmarks. We employ the visual and textual encoders of the official

[2] https://huggingface.co/datasets/FredZhang7/stable-diffusion-prompts-2.47M

		mAP@5	mAP@10	mAP@25	mAP@50
ViT-L	Pic2Word†	8.72	9.51	10.64	11.29
	SEARLE†	11.68	12.73	14.33	15.12
	LinCIR	**12.59**	**13.58**	**15.00**	**15.85**
ViT-H	Pic2Word	11.65	12.33	13.71	14.43
	SEARLE	16.08	16.92	18.81	19.69
	LinCIR	**17.60**	**18.52**	**20.46**	**21.39**
ViT-G	Pic2Word	5.54	5.59	6.68	7.12
	SEARLE	13.20	13.85	15.32	16.04
	LinCIR	**19.71**	**21.01**	**23.13**	**24.18**

Table 3. **CIRCO results.** Results of Pic2Word [37], SEARLE [3], LinCIR by using different CLIP backbones are shown. † denotes that the numbers are measured by the official checkpoint.

		R@1	R@2	R@3
ViT-L	Pic2Word†	11.16	21.47	30.38
	SEARLE†	**12.26**	22.11	31.30
	LinCIR	12.19	**22.76**	**32.38**
ViT-H	Pic2Word	11.89	22.17	31.94
	SEARLE	13.34	23.72	32.72
	LinCIR	**13.76**	**23.87**	**33.16**
ViT-G	Pic2Word	10.67	20.70	29.50
	SEARLE	12.87	22.61	32.46
	LinCIR	**13.66**	**24.64**	**33.54**

Table 4. **GeneCIS results.** The average R@1, R@2, R@3 for "Focus Attribute", "Change Attribute", "Focus Object", and "Change Object" are shown. The full table is in the Appendix.

CLIP ViT-L [35], and OpenCLIP ViT-H and ViT-G [19]. In the Appendix, we show that LinCIR can be easily extended to the other VLMs, such as BLIP [25].

4.2. Experimental protocols

Evaluation benchmarks and metrics. As pointed out by Baldrati et al. [3], the existing CIR benchmarks only have a single positive, which can cause an unreliable evaluation. A similar phenomenon is also reported in the image-text cross-modal retrieval problem by Chun et al. [10]; such benchmarks can lead to a wrong model comparison result. For this reason, we use CIRCO as the main benchmark, which has multiple positives and measures a more reliable ranking-based metric, mAP@K [32]. We also report the R@K evaluation results on three additional datasets, GeneCIS [44], FashionIQ [47], and CIRR [29]. We describe the details of each dataset in the Appendix.

In this paper, we argue that R@1 results can be somewhat noisy due to the false negatives in the dataset. Note that these benchmarks only have a unique positive triplet for each query $\langle x_{i_R}, x_c, x_i \rangle$, *i.e.*, if other plausible images (*i.e.* false negatives) exist in the gallery set, the R@1 score in these benchmarks cannot correctly measure the actual retrieval performance. Due to this reason, we will focus on the mAP score if it is available. Otherwise, we will concentrate on R@K with a larger K (*e.g.*, 10) rather than R@1.

Comparison methods. We compare LinCIR with the recent ZS-CIR methods: Pic2Word [37] and SEARLE [3]. For a fair comparison, we train all methods with the same backbone architecture as LinCIR, namely ViT-H and ViT-G CLIP backbones. ViT-L results are measured using the official checkpoints. We did not directly compare our method with recent methods that require massive external triplet datasets or take a long training time [15, 45]. More comparisons with these methods can be found in the Appendix.

4.3. Main results

The experimental results are summarized in Fig. 4: LinCIR outperforms the comparison methods in training time and retrieval performances. In all benchmarks, we observe that while the performance of LinCIR is enhanced by enlarging the backbone size, Pic2Word and SEARLE show inferior performances with the ViT-G backbone. We presume that it is because Pic2Word and SEARLE have a limited understanding of complex text queries because their ϕ module are trained on texts not diverse enough (*i.e.*, a photo of `[$]`). On the other hand, our ϕ module shows a better understanding of complex texts as LinCIR is trained on diverse real-world texts from the caption datasets.

We also provide the full evaluation results on the four benchmarks below. Tab. 3 shows the evaluation results on the CIRCO dataset. In all experiments, LinCIR outperforms others with a significant gap. We can observe a similar finding in the GeneCIS average R@K results for four different subtasks (Tab. 4), especially for R@K with a larger K. As shown in the Appendix, LinCIR outperforms other models, especially on "Focus Attribute" and "Change Attribute" tasks. We presume that it is because the Pic2Word and SEARLE training prompts are more specialized to objects (a photo of `[$]`), where LinCIR can handle more detailed concepts in the image by altering all the nouns in the sentence (*e.g.*, "gray `[$]` sleeps on a `[$]`").

In the FashionIQ benchmark (Tab. 5), LinCIR even outperforms the state-of-the-art supervised method [2] with a large gap (38.32 vs. 45.11 in the average R@10). Our work is the first ZS-CIR method that outperforms the supervised CIR method despite its generalizability to the other CIR benchmarks and flexibility for handling various conditions.

We observe somewhat mixed results on CIRR (Tab. 6): LinCIR achieves the best R@10 score, but not in some other metrics. We argue that this is because of two reasons. First, R@K with a small K cannot fully reflect the authentic performance. As shown in the Appendix, the retrieval results

		Shirt		Dress		Toptee		Average	
		R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50
ViT-L	Pic2Word†	26.20	43.60	20.00	40.20	27.90	47.40	24.70	43.70
	SEARLE†	26.89	45.58	20.48	43.13	29.32	49.97	25.56	46.23
	LinCIR	**29.10**	**46.81**	**20.92**	**42.44**	**28.81**	**50.18**	**26.28**	**46.49**
ViT-H	Pic2Word	**36.90**	55.99	28.01	51.51	40.18	62.01	35.03	56.50
	SEARLE	36.46	55.45	28.46	51.07	38.81	60.89	34.57	55.80
	LinCIR	**36.90**	**57.75**	**29.80**	**52.11**	**42.07**	**62.52**	**36.26**	**57.46**
ViT-G	Pic2Word	33.17	50.39	25.43	47.65	35.24	57.62	31.28	51.89
	SEARLE	36.46	55.35	28.16	50.32	39.83	61.45	34.81	55.71
	LinCIR	**46.76**	**65.11**	**38.08**	**60.88**	**50.48**	**71.09**	**45.11**	**65.69**
Combiner (supervised) [2]†		39.99	60.45	33.81	59.40	41.41	65.37	38.32	61.74

Table 5. **FashionIQ results.** † denotes that the numbers are from the original paper. LinCIR ViT-G even outperforms the previous state-of-the-art supervised CIR method [2] with a large gap although LinCIR is not directly trained on the FashionIQ dataset.

		Full			Subset		
		R@1	R@5	R@10	R@1	R@2	R@3
ViT-L	Pic2Word†	23.90	51.70	65.30	53.76	74.46	87.08
	SEARLE†	24.24	52.48	66.29	53.76	75.01	88.19
	LinCIR	25.04	53.25	**66.68**	57.11	77.37	88.89
ViT-H	Pic2Word	32.94	63.11	73.86	62.22	81.35	91.23
	SEARLE	34.00	63.98	75.25	64.63	83.21	92.77
	LinCIR	33.83	63.52	**75.35**	62.43	81.47	92.12
ViT-G	Pic2Word	30.41	58.12	69.23	68.92	85.45	93.04
	SEARLE	34.80	64.07	75.11	68.72	84.70	93.23
	LinCIR	35.25	64.72	**76.05**	63.35	82.22	91.98

Table 6. **CIRR results.** Due to the noisy nature of CIRR as a ZS-CIR benchmark, we only highlight the R@10 score for the full CIRR set. The detailed discussion can be found in Sec. 4.3

of LinCIR are plausible to humans, but because the dataset has incomplete positives (*i.e.*, there are many false negatives), the R@1 score cannot correctly evaluate the model performance. According to Chun et al. [10], the similarity between the rankings measured by R@K on a partially annotated benchmark and those measured by mAP on a fully annotated benchmark becomes lower when we use a small K (*e.g.*, 1). Second, as observed by previous works [3, 37], the quality of the CIRR benchmark as a ZS-CIR benchmark is somewhat doubtable. The CIRR text relative captions are often not truly relative (*i.e.*, there exist false positives), and reference images can even be harmful to retrieval. As pointed out by Baldrati et al. [3], this problem becomes more severe when we use a small subset of images, *i.e.*, the subset R@Ks. In summary, due to the noisy nature of CIRR full and subset R@1s, we propose to focus on the CIRR full R@10 scores rather than other metrics. In CIRR full R@10, LinCIR shows the same trends as the other benchmarks.

Supervision design	CIRR dev R@1	Fashion IQ R@10	Fashion IQ R@50
a photo of `[$]` [37]	21.65	24.93	44.35
Ours, but `[$]` extracted by ψ_I	22.63	22.01	39.87
Our SMP design choice	**25.66**	**26.28**	**46.49**

Table 7. **Impact of the supervision design.** Different target text designs (*e.g.*, `[$]` sleeps on `[$]` in Fig. 3 (b)) affect the performances. "Ours, but `[$]` extracted by ψ_I" denotes that the textual encoder before the ϕ module is replaced with the visual encoder.

SMP Masking strategy	CIRR dev R@1	Fashion IQ R@10	Fashion IQ R@50
All non-keyword tokens	20.59	19.61	36.97
Random token	22.98	22.44	40.92
All noun tokens	24.95	25.16	45.19
1 keyword token	23.92	26.42	45.91
3 keyword tokens	24.66	**26.69**	**46.54**
5 keyword tokens	25.14	26.28	46.29
All keyword tokens	**25.66**	26.28	46.49

Table 8. **Impact of the SMP masking strategy.** The CIRR retrieval performances by varying the masking strategy for SMP (*i.e.*, "Keyword masking" in Fig. 3 (b)) are shown. We define "keyword" as consecutive adjectives and nouns, except for "All noun tokens". "All noun tokens" defines keywords as nouns.

4.4. Analysis

In this subsection, we provide detailed analyses of our design choices. If not specified, we compare the models on the CIRR dev split and FashionIQ test split.

Impact of Self-Masking Projection (SMP) supervision. We compare two variants of SMP in Tab. 7. First, instead of employing our keyword masking strategy, we use "a photo

Noise type	CIRR dev R@1	Fashion IQ R@10	Fashion IQ R@50
No noise	19.76	20.42	38.31
Student-t	23.08	22.81	41.00
Exponential	23.51	25.78	45.26
χ^2	23.54	23.11	41.89
$\mathcal{N}(0, 1)$	23.70	23.31	41.89
Unif(-1, 1)	25.14	25.88	45.78
$\mathcal{N}(0, 1) \times$ Unif(0, 1)	**25.47**	**26.05**	**46.29**

Table 9. **Impact of the choice of the random noise.** Adding noise to the textual latent space helps to mitigate the inferior generalizability due to the modality gap. Moreover, using better random noise can significantly boost the overall performances.

CC3M	SDP	COYO	OWT	CIRCO mAP@5	GeneCIS R@3	FashionIQ R@10	CIRR R@10	Avg
✔	✘	✘	✘	**13.72**	32.80	25.11	64.95	34.15
✘	✔	✘	✘	9.52	32.38	23.63	62.58	32.03
✘	✘	✔	✘	11.36	31.48	26.18	65.33	33.59
✘	✘	✘	✔	9.67	30.90	24.41	64.05	32.26
✔	✔	✘	✘	12.59	32.38	26.28	66.68	**34.48**
✔	✔	✘	✔	10.06	**33.36**	21.11	63.16	31.92
✔	✔	✔	✘	11.54	32.08	**26.97**	**66.80**	34.35

Table 10. **Impact of the training corpus.** OpenWebText (OWT) [14] and SD prompts (SDP) are text-only datasets and CC3M [39] and COYO-700M (COYO) [5] are image-text aligned datasets. In our experiments, we use CC3M + SDP for the training corpus, considering the dataset scale and overall CIR performances.

of "`[$]`" as Pic2Word and SEARLE to train the ϕ module. Second, we replace the textual encoder before ϕ module with the visual encoder using the corresponding image of the caption. In the table, "a photo of `[$]` that `[cond]`" variant performs worse than our design choice due to the limited diversity of the input texts. Interestingly, we observe that using both image and text pairs for LinCIR performs worse than our design choice. Note that the second model is trained on CC3M image-text pairs without using 2.47M SD prompts as our design choice. We presume it makes the ϕ model overfitted to CC3M image-text relationships, which significantly differ from our target CIR datasets.

Impact of masking design choice. We compare other masking design choices, such as random tokens or non-keyword tokens, with our design choice in Tab. 8. Tab. 8 shows that (1) masking the keywords performs better than masking the others. (2) Our keyword design – *i.e.*, consecutive adjectives and nouns – is better than defining keywords as nouns. (3) The overall performances are enhanced by increasing the masked keywords. As the differences are not significant, we mask all the keywords for simplicity.

Impact of the random noise addition. As we discussed in Sec. 3.2, the random noise addition is critical to mitigating the modality gap. Tab. 9 supports this claim: when we do not add any noise, the performance becomes the worst. Our design choice shows the best performance among the other noise designs due to the diverse norm size of our distribution as shown in Fig. B.2.

The impact of the training corpus. We evaluate the impact of the training corpus in Tab. 10. We use four corpora, CC3M [39] (3M captions), StableDiffusion Prompts (SDP) (2.47M text prompts), COYO-700M (700M captions) [5] and OpenWebText (OWT) [14] (8M web texts). CC3M and COYO are image-text paired datasets, and OWT and SDP are text-only datasets. The example training samples of each dataset are shown in the Appendix. In Tab. 10, we observe that models trained with image descriptions (CC3M and COYO) perform better than models trained with general texts because general web texts are often irrelevant to visual information. In addition, using multiple corpora improves the overall performance, *e.g.*, CC3M (34.15) $\rightarrow$ CC3M + SDP (34.48). Although we observe that using more massive captions (*e.g.*, 3M + 2.47M + 700M) can be helpful for some CIR tasks, such as FashionIQ and CIRR, we use CC3M and SDP for our training set considering the corpus size (3M + 2.47M), and the overall performances.

Qualitative results. We provide the additional qualitative retrieval results in the Appendix. In summary, LinCIR shows qualitatively plausible retrieval performance even in a large-scale image database, *e.g.* LAION-2B. Also, we observe that the retrieved results by LinCIR often suffer from false negatives in benchmark datasets, showing the limitation of the R@1 evaluation on the existing CIR benchmarks.

5. Conclusion

We propose a novel zero-shot composed image retrieval (ZS-CIR) framework named Language Only training for Composed Image Retrieval (LinCIR). LinCIR presents a breakthrough in addressing the challenges associated with the previous ZS-CIR methods. By leveraging a novel self-supervision technique, Self-Masking Projection (SMP), LinCIR eliminates the dependency on expensive CIR triplets, opting for a training process solely based on text inputs. This innovative approach significantly enhances scalability and generalizability, overcoming limitations observed in existing ZS-CIR methods. Notably, our LinCIR model achieves remarkable efficiency and ZS-CIR performances compared to other methods on multiple CIR benchmarks, including CIRCO, GeneCIS, FashionIQ, and CIRR. We underscore the effectiveness of our language-only training framework, offering a potent solution with wide-ranging implications for image retrieval tasks.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[2] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Conditioned and composed image retrieval combining and partially fine-tuning clip-based features. In *CVPR*, pages 4959–4968, 2022. 1, 2, 6, 7

[3] Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Alberto Del Bimbo. Zero-shot composed image retrieval with textual inversion. In *ICCV*, 2023. 1, 2, 3, 4, 5, 6, 7, 11, 13

[4] 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. *NeurIPS*, 33:1877–1901, 2020. 3

[5] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022. 8, 11

[6] Yanbei Chen and Loris Bazzani. Learning joint visual semantic matching embeddings for language-guided retrieval. In *ECCV*, pages 136–152. Springer, 2020. 1

[7] Yanbei Chen, Shaogang Gong, and Loris Bazzani. Image search with text feedback by visiolinguistic attention learning. In *CVPR*, pages 3001–3011, 2020.

[8] Yiyang Chen, Zhedong Zheng, Wei Ji, Leigang Qu, and Tat-Seng Chua. Composed image retrieval with text feedback via multi-grained uncertainty regularization. *arXiv preprint arXiv:2211.07394*, 2022. 1

[9] Sanghyuk Chun. Improved probabilistic image-text representations. *arXiv preprint arXiv:2305.18171*, 2023. 11

[10] Sanghyuk Chun, Wonjae Kim, Song Park, Minsuk Chang Chang, and Seong Joon Oh. Eccv caption: Correcting false negatives by collecting machine-and-human-verified image-caption associations for ms-coco. In *ECCV*, 2022. 6, 7, 11

[11] Ginger Delmas, Rafael S Rezende, Gabriela Csurka, and Diane Larlus. Artemis: Attention-based retrieval with text-explicit matching and implicit similarity. In *ICLR*, 2022. 1

[12] Eric Dodds, Jack Culpepper, Simao Herdade, Yang Zhang, and Kofi Boakye. Modality-agnostic attention fusion for visual search with text feedback. *arXiv preprint arXiv:2007.00145*, 2020. 1

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 13

[14] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019. 8, 11

[15] Geonmo Gu, Sanghyuk Chun, HeeJae Jun, Yoohoon Kang, Wonjae Kim, and Sangdoo Yun. Compodiff: Versatile composed image retrieval with latent diffusion. *arXiv preprint arXiv:2303.11916*, 2023. 6, 13

[16] Sophia Gu, Christopher Clark, and Aniruddha Kembhavi. I can't believe there's no images! learning visual tasks using only language supervision. In *ICCV*, pages 2672–2683, 2023. 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 11

[18] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 5

[19] 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. If you use this software, please cite it as below. 1, 2, 6

[20] Surgan Jandial, Ayush Chopra, Pinkesh Badjatiya, Pranit Chawla, Mausoom Sarkar, and Balaji Krishnamurthy. Trace: Transform aggregate and compose visiolinguistic representations for image search with text feedback. *arXiv preprint arXiv:2009.01485*, 7, 2020. 1

[21] Surgan Jandial, Pinkesh Badjatiya, Pranit Chawla, Ayush Chopra, Mausoom Sarkar, and Balaji Krishnamurthy. Sac: Semantic attention composition for text-conditioned image retrieval. In *WACV*, pages 4021–4030, 2022.

[22] Jongseok Kim, Youngjae Yu, Hoeseong Kim, and Gunhee Kim. Dual compositional learning in interactive image retrieval. In *AAAI*, pages 1771–1779, 2021. 1

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73, 2017. 11

[24] Seungmin Lee, Dongwan Kim, and Bohyung Han. Cosmo: Content-style modulation for image retrieval with text feedback. In *CVPR*, pages 802–812, 2021. 1

[25] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, pages 12888–12900. PMLR, 2022. 2, 6, 13, 14

[26] Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. Decap: Decoding CLIP latents for zero-shot captioning via text-only training. In *ICLR*, 2023. 3

[27] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *NeurIPS*, 35:17612–17625, 2022. 2, 3, 4

[28] 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. 5, 11

[29] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *ICCV*, pages 2125–2134, 2021. 1, 2, 5, 6, 11

[30] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *ICCV*, pages 2125–2134, 2021. 1

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[32] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 681–699. Springer, 2020. 6, 11

[33] David Nukrai, Ron Mokady, and Amir Globerson. Text-only training for image captioning using noise-injected clip. In *EMNLP*, 2022. 2, 3, 4

[34] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *CVPR*, pages 13018–13028, 2021. 11

[35] 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 *ICML*, pages 8748–8763. PMLR, 2021. 1, 2, 6

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015. 11

[37] 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. 1, 2, 3, 4, 6, 7, 11, 13

[38] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks Track*, 2022. 1

[39] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 1, 2, 5, 8, 11

[40] Minchul Shin, Yoonjae Cho, Byungsoo Ko, and Geonmo Gu. Rtic: Residual learning for text and image composition using graph convolutional network. *arXiv preprint arXiv:2104.03015*, 2021. 1

[41] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*, 2018. 11

[42] Yuxin Tian, Shawn Newsam, and Kofi Boakye. Image search with text feedback by additive attention compositional learning. *arXiv preprint arXiv:2203.03809*, 2022. 1

[43] 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. 13

[44] Sagar Vaze, Nicolas Carion, and Ishan Misra. Genecis: A benchmark for general conditional image similarity. In *CVPR*, 2023. 2, 5, 6, 11

[45] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. CoVR: Learning composed video retrieval from web video captions. *arXiv:2308.14746*, 2023. 6, 13

[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*, pages 6439–6448, 2019. 1

[47] Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. Fashion iq: A new dataset towards retrieving images by natural language feedback. In *CVPR*, pages 11307–11317, 2021. 1, 2, 5, 6, 11, 13

[48] Youngjae Yu, Seunghwan Lee, Yuncheol Choi, and Gunhee Kim. Curlingnet: Compositional learning between images and text for fashion iq data. *arXiv preprint arXiv:2003.12299*, 2020. 1