COMPODIFF: VERSATILE COMPOSED IMAGE RETRIEVAL WITH LATENT DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper proposes a novel diffusion-based model, CompoDiff, for solving Composed Image Retrieval (CIR) with latent diffusion and presents a newly created dataset, named SynthTriplets18M, of 18 million reference images, conditions, and corresponding target image triplets to train the model. CompoDiff and SynthTriplets18M tackle the shortages of the previous CIR approaches, such as poor generalizability due to the small dataset scale and the limited types of conditions. CompoDiff not only achieves a new zero-shot state-of-the-art on four CIR benchmarks, including FashionIQ, CIRR, CIRCO, and GeneCIS, but also enables a more versatile and controllable CIR by accepting various conditions, such as negative text and image mask conditions, and the controllability to the importance between multiple queries or the trade-off between inference speed and the performance which are unavailable with existing CIR methods. The code and dataset samples are available at `Supplementary Materials`.

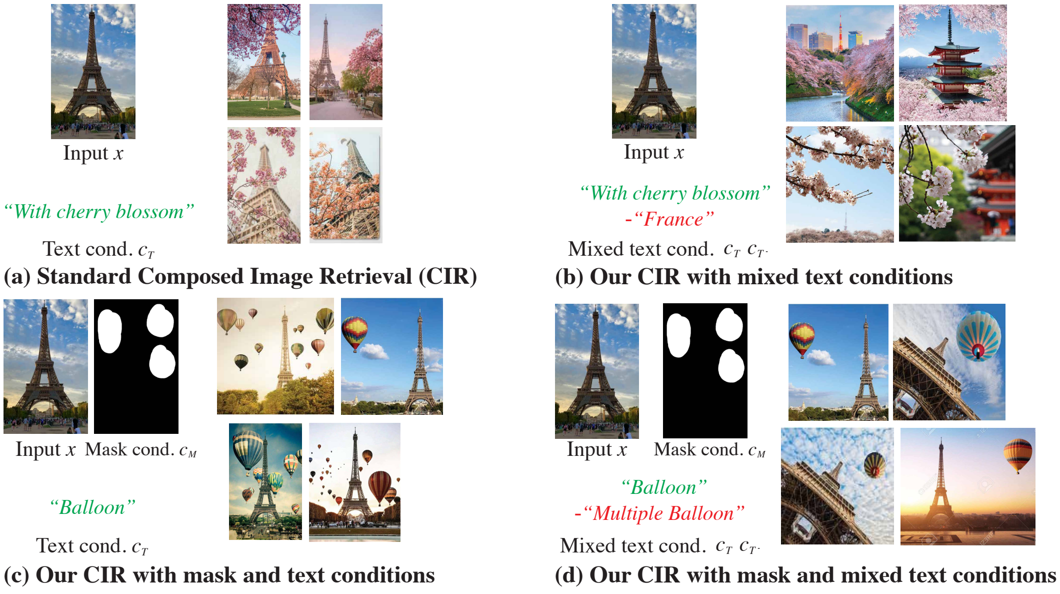
1 INTRODUCTION

Imagine a customer seeking a captivating cloth serendipitously found on social media but not the most appealing materials and colors. In such a scenario, the customer needs a search engine that can process composed queries, *e.g.*, the reference garment image along with text specifying the preferred material and color. This task has been recently formulated as *Composed Image Retrieval (CIR)* (Liu et al., 2021) – Fig. 1 (a). CIR systems offer the benefits of searching for visually similar items as image-to-image retrieval while providing a high degree of freedom to depict text query as text-to-image retrieval. CIR can also improve the search quality by iteratively taking user feedback.

The existing CIR methods address the problem by combining image and text features using additional fusion models, *e.g.*, $z_i = \texttt{fusion}(z_{i_R}, z_c)$ where z_i, z_c, z_{i_R} are the target image, conditioning text, and reference image features, respectively[1]. Although the fusion-based approaches have shown great success (*e.g.*, training an additional module that combines the image and text features (Baldrati et al., 2022)), they have fundamental limitations. First, they need a pre-collected human-verified dataset of triplets $\langle x_{i_R}, x_c, x_i \rangle$ consisting of a reference image (x_{i_R}), a condition (x_c), and the corresponding target image (x_i) to train the fusion module. However, obtaining such triplets can be costly and sometimes impossible, therefore, the size of the existing CIR datasets are small (*e.g.*, 30K triplets for Fashion-IQ (Wu et al., 2021) and 36K triplets for CIRR (Liu et al., 2021)), resulting in a lack of generalizability to other real-world datasets. Second, the fusion module is not flexible; it cannot handle versatile conditions beyond a limited textual one. For instance, a user might want to include a negative text that is not desired for the search ($x_{c_{T^-}}$) (*e.g.*, an image + "with cherry blossom" − "France", as in Fig. 1 (b)), indicate where (x_{c_M}) the condition is applied (*e.g.*, an image + "balloon" + indicator, as in Fig. 1 (c)), or construct a complex condition with a mixture of them. Furthermore, once the fusion model is trained, it will always produce the same z_i for the given z_{i_R} and z_c to users. However, a practical retrieval system needs to control the level of conditions by its applications (*e.g.*, more focusing on the image while the changes by the text should be small).

This paper aims to achieve a generalizable CIR model with diverse and versatile conditions by using latent diffusion. We treat the CIR task as a conditional image editing task on the latent space, *i.e.*, $z_i = \texttt{Edit}(z_{i_R} | z_c, \ldots)$. Our diffusion-based CIR model, named CompoDiff, can easily deal

[1] Throughout this paper, we will use x to denote raw data and z to denote a vector encoded from x.



(a) Standard Composed Image Retrieval (CIR)
(b) Our CIR with mixed text conditions
(c) Our CIR with mask and text conditions
(d) Our CIR with mask and mixed text conditions

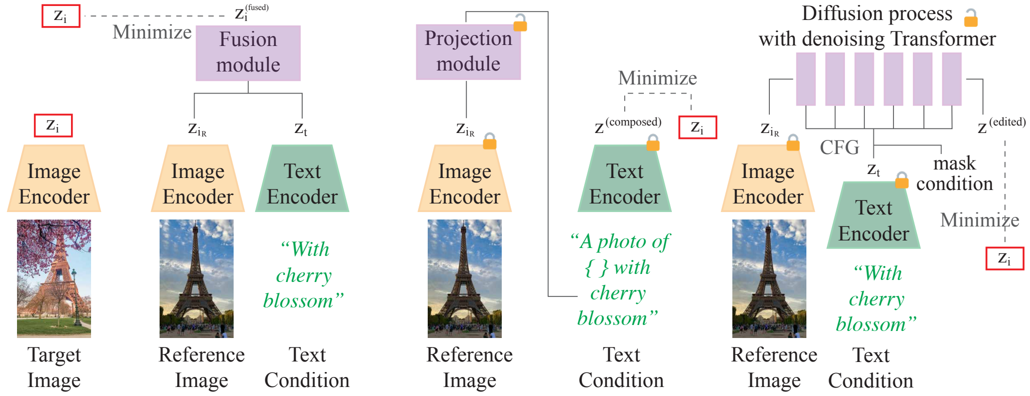
Figure 1: **Composed Image Retrieval (CIR) scenarios.** (a) A standard CIR scenario. (b-d) Our versatile CIR scenarios with mixed conditions (*e.g.*, negative text and mask). Results by CompoDiff on LAION-2B.

with versatile and complex conditions, benefiting from the flexibility of the latent diffusion model (Rombach et al., 2022) and the classifier-free guidance (Ho & Salimans, 2022). Motivated by recent advances in diffusion models, we train a latent diffusion model that translates the embedding of the reference image (z_{i_R}) into the embedding of the target image (z_i) guided by the embedding of the given condition (z_c). As in Fig. 1, CompoDiff can handle various conditions, which is not possible with the standard CIR scenario with the limited text condition x_{c_T}. Although our method has an advantage over the existing fusion-based methods in terms of versatility, our approach also needs to be trained with triplet datasets where the scale of the existing CIR triplet datasets is extremely small.

To address the data scale issue, we synthesize a vast set of high-quality **18.8M** triplets of $\langle x_{i_R}, x_c, x_i \rangle$ by utilizing large-scale generative models such as OPT and Stable Diffusion. Our approach is fully automated without human verification; hence, it is highly scalable even to the 18.8M scale. We follow InstructPix2Pix (IP2P) (Brooks et al., 2022) for synthesizing triplets, while our dataset contains ×40 more triplets and ×12.5 more keywords (*e.g.*, objects, background details, or textures) than IP2P. However, we introduce additional techniques than IP2P, resulting in a better training dataset (See Table 3). Our massive dataset, named **SynthTriplets18M**, is over 500 times larger than existing CIR datasets and covers a diverse and extensive range of conditioning cases. Our dataset itself makes a significant contribution, resulting in a notable performance improvement for any CIR model. For example, ARTEMIS (Delmas et al., 2022) trained exclusively with SynthTriplets18M shows outperforming zero-shot performance even than its FashionIQ-trained counterpart (40.6 vs. 38.2).

To show the generalizability of the models, we evaluate the models on the "zero-shot" (ZS) CIR scenario using four CIR benchmarks: FashionIQ (Wu et al., 2021), CIRR (Liu et al., 2021), CIRCO (Baldrati et al., 2023), and GeneCIS (Vaze et al., 2023); *i.e.*, we report the retrieval results by the models trained on our SynthTriplets18M and a large-scale image-text paired dataset without access to the target triplet datasets. In all experiments, CompoDiff achieves the best zero-shot performances with significant gaps (See Table 2). Moreover, we observe that the fusion-based approaches solely trained on SynthTriplets18M (*e.g.*, Combiner (Baldrati et al., 2022)) show comparable or outperforming zero-shot CIR performances compared to the previous SOTA zero-shot CIR methods, *e.g.*, Pic2Word (Saito et al., 2023) and SEARLE (Baldrati et al., 2023). Furthermore, we qualitatively observe that the retrieval results of CompoDiff are semantically better than previous zero-shot CIR methods, such as Pic2Word, on a large-scale image database, *e.g.*, LAION-2B.

Another notable advantage of CompoDiff is the ability to control various conditions during inference, which is inherited from the nature of diffusion models: Users can adjust the weight of conditions to make the model focus on the user's preference. Users can also manipulate the randomness



(a) Fusion-based approaches (e.g., ARTEMIS, Combiner)

(b) Inversion-based approaches (e.g., Pic2Word, SEARLE)

(c) Ours editing-based approach

Figure 2: **Comparisons of CIR approaches.** (a) Fusion-based approaches, such as ARTEMIS (Delmas et al., 2022) and Combiner (Baldrati et al., 2022) fuse the image and text features using a fusion module. (b) Inversion-based approaches, such as Pic2Word (Saito et al., 2023) and SEARLE (Baldrati et al., 2023), project the image feature into the text space. Then, it performs text-to-image retrieval to CIR. (c) Our editing-based approach applies a diffusion process to the image feature with classifier-free guidance (CFG) by additional conditions, such as text condition and mask. (b) and (c) use frozen encoders, and (a) usually tunes the encoders.

of the models to vary the degree of serendipity. In addition, CompoDiff can control the speed of inference with minimal sacrifice in retrieval performance, accomplished by adjusting the number of sampling steps in the diffusion model. As a result, CompoDiff can be deployed in various scenarios with different computational budgets. Note that all of these controllability features can be achieved by controlling the inference parameters of classifier-free guidance, without any model training.

2 Related Works

Composed image retrieval. The overview of the differences between CIR approaches is illustrated in Fig. 2. The mainstream CIR models have focused on *multi-modal fusion methods*, which combine image and text features extracted from separate visual and text encoders, as in Fig. 2 (a). In Appendix A.1, we provide a detailed explanation of fusion methods which still require expensive pre-collected and human-verified triplets of the target domain.

To solve this problem, recent studies aim to solve CIR tasks in a zero-shot manner. *Inversion-based zero-shot CIR methods*, such as Pic2Word (Saito et al., 2023) and SEARLE (Baldrati et al., 2023), tackle the problem through text-to-image retrieval tasks where the input image is projected into the condition text – Fig. 2 (b). Note that our zero-shot CIR scenario is slightly different from theirs; our zero-shot CIR denotes that the models are not trained on the target triplet datasets, but trained on our synthetic dataset, SynthTriplets18M, and image-text paired dataset, *e.g.*, the LAION-2B dataset (Schuhmann et al., 2022a). On the other hand, Saito et al. (2023) and Baldrati et al. (2023) use the term zero-shot when the CIR models are trained without a triplet dataset.

All the existing CIR models only focus on text conditions (z_{c_T}) (*e.g.*, Fig. 1 (a)) and have difficulties in handling versatile scenarios (*e.g.*, Fig. 1 (b-d)) with a lack of the controllability. On the other hand, our method enables multiple various conditions and controllabilities with strong zero-shot performances by employing (1) a latent diffusion model (Rombach et al., 2022) with classifier-free guidance (Ho & Salimans, 2022) and (2) a massive high-quality synthetic dataset, SynthTriplets18M.

Dataset creation with diffusion models. A conventional data collection process for $\langle x_{i_R}, x_c, x_i \rangle$ is two-staged: collecting candidate reference-target image pairs and manually annotating the modification sentences by human annotators (Wu et al., 2021; Liu et al., 2021). For example, FashionIQ (Wu et al., 2021) collects the candidate pairs from the same item category (*e.g.*, shirt, dress, and top) and manually annotates the relative captions by crowd workers. CIRR (Liu et al., 2021) gathers the candidate pairs from real-life images from NLVR^2 (Suhr et al., 2018). The data collection process for $\langle x_{i_R}, x_c, x_i \rangle$ inevitably becomes expensive, making it difficult to scale CIR datasets. We mitigate this problem by generating a massive synthetic dataset instead of relying on human annotators.

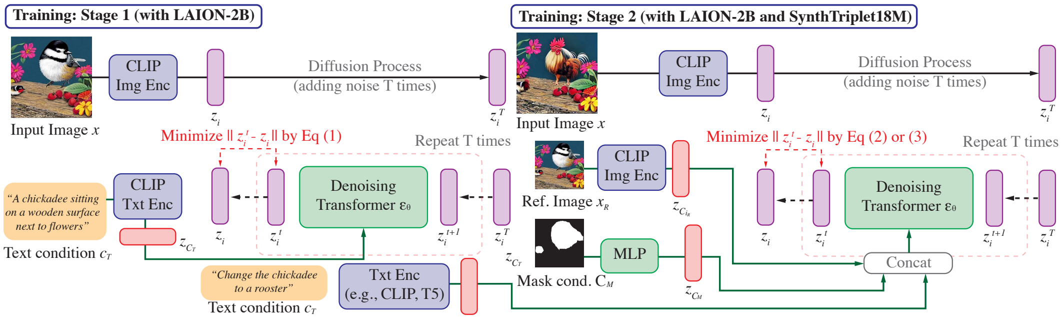


Figure 3: **Training overview.** Stage 1 is trained on LAION-2B with Eq. (1). For stage 2, we alternatively update Denoising Transformer ϵ_θ on LAION-2B with Eq. (1) and 2 and SynthTriplets18M with Eq. (3).

Recently, there have been attempts to generate synthetic data to improve model performance (Brooks et al., 2022; Nair et al., 2022; Shipard et al., 2023) by exploiting the powerful generation capabilities of diffusion models (Ho et al., 2020; Rombach et al., 2022). In particular, Brooks et al. (2022) proposes a generation process for $\langle x_{i_R}, x_c, x_i \rangle$ to train an image editing model. We scale up the dataset synthesis process of Brooks et al. (2022) from 1M triplets to 18.8M. We also make the triplets more diverse by applying the object-level modification process, resulting in better CIR performances on a 1M scale. We describe the details of our dataset generation process in Section 4.

3 CompoDiff: Composed Image Retrieval with Latent Diffusion

This section introduces Composed Image Retrieval with Latent Diffusion (CompoDiff). Fig. 2 (c) shows an overview. CompoDiff employs a diffusion process in the frozen CLIP latent feature space with classifier-free guidance (CFG). Unlike previous latent diffusion models (Rombach et al., 2022), we adopt the Transformer architecture for the denoiser. As we train our model with various tasks, such as text-to-image (T2I) generation, masked T2I and triplet-based generation, CompoDiff can handle various conditions (*e.g.*, negative text, mask, or a mixture of conditions), while the previous approaches only limit beyond the positive text instruction. We will describe the details of training and inference, and how CompoDiff handles various conditions through latent diffusion and CFG.

3.1 Preliminary: Diffusion model

Diffusion models (DMs) are generative models by gradually denoising a Gaussian noise. This process corresponds to the reverse process of a Markov Chain. The recent success of DMs shows remarkable improvements in image synthesis (Song et al., 2020; Rombach et al., 2022; Ramesh et al., 2022; Ho & Salimans, 2022). We provide more backgrounds of DMs in Appendix A.2.

3.2 Training

CompoDiff uses a two-stage training strategy (Fig. 3). In stage 1, we train a text-to-image latent diffusion model on LAION-2B. In stage 2, we fine-tune the model on our synthetic triplet dataset, SynthTriplets18M, and LAION-2B. Below, we describe the details of each stage.

In stage 1, we train a transformer decoder to convert CLIP textual embeddings into CLIP visual embeddings. This stage is similar to training the prior model in Dalle-2 (Ramesh et al., 2022), but our model takes only two tokens; a noised CLIP image embedding and a diffusion timestep embedding. The Dalle-2 prior model is computationally inefficient because it also takes 77 encoded CLIP text embeddings as an input. However, CompoDiff uses the encoded text embeddings as conditions through cross-attention mechanisms, which speeds up the process by a factor of three while maintaining similar performance (See Appendix D.4). Instead of using the noise prediction of Ho et al. (2020), we train the transformer decoder to predict the denoised z_i directly due to the stability.

Now, we introduce the objective of the first stage with CLIP image embeddings of an input image z_i, encoded CLIP text embeddings for text condition z_{c_T}, and the denoising Transformer ϵ_θ:

$$\mathcal{L}_{\text{stage1}} = \mathbb{E}_{t\sim[1,T]} \| z_i - \epsilon_\theta(z_i^{(t)}, t | z_{c_T}) \|^2 \tag{1}$$

During training, we randomly drop the text condition by replacing z_{c_T} with a null text embedding $\varnothing_{c_T}$ in order to induce CFG. We use the empty text CLIP embedding ("") for the null embedding.

In stage 2, we incorporate condition embeddings, injected by cross-attention, into CLIP text embeddings, along with CLIP reference image visual embeddings and mask embeddings (See Fig. 3). We fine-tune the model with three different tasks: a conversion task that converts textual embeddings into visual embeddings (Eq. (1)), a mask-based conversion task (Eq. (2)), and the triplet-based CIR task with SynthTriplets18M (Eq. (3)). The first two tasks are trained on LAION-2B and the last one is trained on SynthTriplets18M. We alternatively update the model with Eq. (1), Eq. (2), and Eq. (3) with the proportions 30%, 30%, 40%. When we train a model with Eq. (3) only, we observe that the representation learned in stage 1 is easily forgotten (Table D.6). Below, we describe the details of the mask-based conversion and triplet-based CIR tasks.

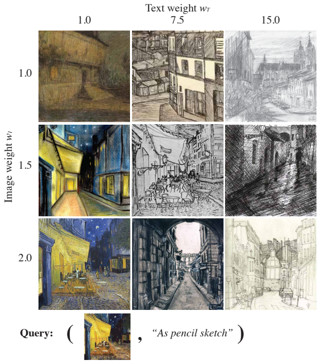


Figure 4: **Condition control for inference.** Example retrieval results on the LAION images by varying image and text weights, w_I and w_T in Eq. (4).

The mask-based conversion task learns a diffusion process that recovers the full image embedding from a masked image embedding. As we do not have mask conditions in the dataset, we extract masks using a zero-shot text-conditioned segmentation model, CLIPSeg (Lüddecke & Ecker, 2022). For the text condition, we use the nouns of the given caption. Then, we add a Gaussian random noise to the mask region of the image and extract $z_{i,\text{masked}}$. We also introduce mask embedding z_{c_M} by projecting a 64×64 resized segmentation mask to the CLIP embedding dimension using a single MLP. Now, we introduce the mask-based conversion task as follows:

$$\mathcal{L}_{\text{stage2\_masked\_conversion}} = \mathbb{E}_{t\sim[1,T]} \| z_i - \epsilon_\theta(z_{i,\text{masked}}^{(t)}, t | z_{c_T}, z_{i,\text{masked}}, z_{c_M}) \|^2, \tag{2}$$

where z_{c_M} is a mask embedding. Finally, we introduce the triplet-based training objective to solve CIR tasks while Eq. (1) and Eq. (2) are trained on LAION-2B. Our last objective is as follows:

$$\mathcal{L}_{\text{stage2\_triplet}} = \mathbb{E}_{t\sim[1,T]} \| z_{i_T} - \epsilon_\theta(z_{i_T}^{(t)}, t | z_{c_T}, z_{i_R}, z_{c_M}) \|^2, \tag{3}$$

where z_{i_R} is a reference image feature and z_{i_T} is a modified target image feature

We randomly drop the conditions of stage 2, as in stage 1, except for the mask conditions. We use an all-zero mask condition for the tasks that do not use a mask condition. When we drop the image condition of Eq. (3), we replace z_{i_R} with the null image feature, an all zero vector.

3.3 Inference

Given a reference image feature z_{i_R}, a text condition feature z_{c_T}, and a mask embedding z_{c_M}, we apply a denoising diffusion process based on CFG (Ho & Salimans, 2022) as follows:

$$\begin{aligned} \tilde{\epsilon}_\theta(z_i^{(t)}, t | z_{c_T}, z_{i_R}, z_{c_M}) = & \epsilon_\theta(z_i^{(t)}, t | \varnothing_{c_T}, \varnothing_{i_R}, z_{c_M}) \\ & + w_I(\epsilon_\theta(z_i^{(t)}, t | \varnothing_{c_T}, z_{i_R}, z_{c_M}) - \epsilon_\theta(z_i^{(t)}, t | \varnothing_{c_T}, \varnothing_{i_R}, z_{c_M})) \\ & + w_T(\epsilon_\theta(z_i^{(t)}, t | z_{c_T}, z_{i_R}, z_{c_M}) - \epsilon_\theta(z_i^{(t)}, t | \varnothing_{c_T}, z_{i_R}, z_{c_M})) \end{aligned} \tag{4}$$

where $\varnothing$ denotes null embeddings, *i.e.*, the CLIP textual embedding for empty text ("") for the text null embedding and an all-zero vector for the image null embedding. One of the main advantages of Eq. (4) is the ability to handle various conditions at the same time. When using negative text, we simply replace $\varnothing_{i_T}$ with the CLIP text embeddings c_{T-} for the negative text.

Another advantage of CFG is the controllability of the queries without training, *e.g.*, it allows to control the degree of focus on image features to preserve the visual similarity with the reference by simply adjusting the weights w_I or w_T in Eq. (4). We show the top-1 retrieved item by varying the image and text weights w_I and w_T from LAION-2B in Fig. 4. By increasing w_I, CompoDiff behaves more like an image-to-image retrieval model. Increasing w_T, on the other hand, makes CompoDiff focus more on the "pencil sketch" text condition. We use (w_I, w_T) = (1.5, 7.5) for our experiments. The retrieval performance by varying w_I and w_T is shown in Appendix D.4.

Since our retrieval model is based on a diffusion process, we can easily control the balance between the inference time and the retrieval quality of the modified feature by varying step size. In practice, we set the step size to 5, which shows the best trade-off. More details can be found in Section 5.3.

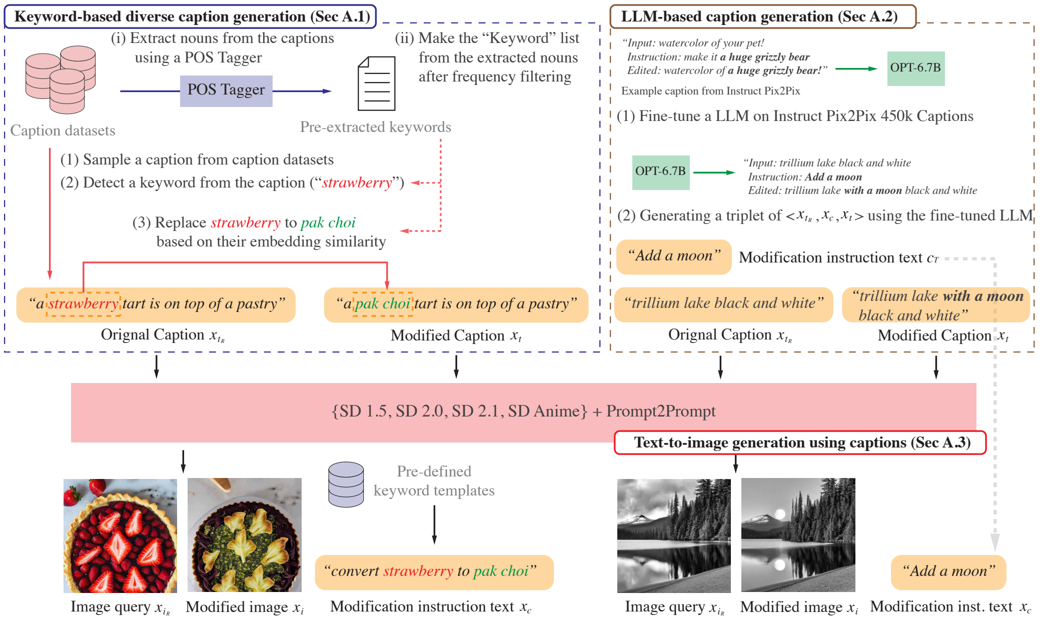

Keyword-based diverse caption generation (Sec A.1)
(i) Extract nouns from the captions using a POS Tagger
POS Tagger
(ii) Make the "Keyword" list from the extracted nouns after frequency filtering
Caption datasets
Pre-extracted keywords
(1) Sample a caption from caption datasets
(2) Detect a keyword from the caption ("strawberry")
(3) Replace strawberry to pak choi based on their embedding similarity
"a strawberry tart is on top of a pastry"
"a pak choi tart is on top of a pastry"
Orignal Caption x_{t_R}
Modified Caption x_t
LLM-based caption generation (Sec A.2)
"Input: watercolor of your pet!
Instruction: make it a huge grizzly bear
Edited: watercolor of a huge grizzly bear!"
OPT-6.7B
Example caption from Instruct Pix2Pix
(1) Fine-tune a LLM on Instruct Pix2Pix 450k Captions
OPT-6.7B
"Input: trillium lake black and white
Instruction: Add a moon
Edited: trillium lake with a moon black and white
(2) Generating a triplet of $<x_{t_R}, x_c, x_t>$ using the fine-tuned LLM
"Add a moon"
Modification instruction text c_T
"trillium lake black and white"
"trillium lake with a moon black and white"
Orignal Caption x_{t_R}
Modified Caption x_t
{SD 1.5, SD 2.0, SD 2.1, SD Anime} + Prompt2Prompt
Text-to-image generation using captions (Sec A.3)
Pre-defined keyword templates
"convert strawberry to pak choi"
"Add a moon"
Image query x_{i_R}
Modified image x_i
Modification instruction text x_c
Image query x_{i_R}
Modified image x_i
Modification inst. text x_c

Figure 5: **Overview of the generation process for SynthTriplets18M.** $\langle x_{i_R}, x_c, x_i\rangle$ from $\langle x_{t_R}, x_c, x_t\rangle$.

	IP2P (Brooks et al., 2022)	SynthTriplets18M (ours)
$\langle x_{t_R}, x_c, x_t\rangle$	450k	60M
Unique object terms	47,345	586,369
$\langle x_{i_R}, x_c, x_i\rangle$ (Keyword-based)	-	11.4M
$\langle x_{i_R}, x_c, x_i\rangle$ (LLM-based)	1M	7.4M
$\langle x_{i_R}, x_c, x_i\rangle$ (Total)	1M	18.8M

Table 1: **Dataset statistics.** We compare SynthTriplets18M and IP2P in terms of the dataset statistics. $\langle x_{t_R}, x_c, x_t\rangle$ denotes the triplet of captions {original caption, modification instruction, and modified caption} and $\langle x_{i_R}, x_c, x_i\rangle$ denotes the triplet of {original image, modification instruction, and modified image}.

4 SynthTriplets18M: Massive High-Quality Synthesized Dataset

CIR requires a dataset of triplets $\langle x_{i_R}, x_c, x_i\rangle$ of a reference image (x_{i_R}), a condition (x_c), and the corresponding target image (x_i). Instead of collecting a dataset by humans, we propose to generate massive triplet data points by using large generative models. We follow the main idea of Instuct Pix2Pix (IP2P) (Brooks et al., 2022). First, we generate $\langle x_{t_R}, x_c, x_t\rangle$ where x_{t_R} is a reference caption, x_c is a modification instruction text, and x_t is the caption modified by x_c. We use two strategies to generate $\langle x_{t_R}, x_c, x_t\rangle$: (1) We collect massive captions from the existing caption datasets and generate the modified captions by replacing the keywords in the reference caption. (2) We fine-tune a large language model, OPT-6.7B (Zhang et al., 2022), on the generated caption triplets from Brooks et al. (2022) – Note that this process is lightweight for the sake of the LoRA fine-tuning (Hu et al., 2021). After generating massive triplets of $\langle x_{t_R}, x_c, x_t\rangle$, we generate images from the caption triplets using StableDiffusion (SD) (Rombach et al., 2022) and Prompt-to-Prompt (Hertz et al., 2022) following Brooks et al. (2022). We employ CLIP-based filtering to ensure high-quality triplets. The entire generation process is illustrated in Fig. 5 and Appendix B due to the page limitation.

Compared to manual dataset collections (Wu et al., 2021; Liu et al., 2021), our approach can easily generate more diverse triplets even if a triplet rarely occurs in reality (See the examples in Fig. 5). Compared to the synthetic dataset of IP2P, SynthTriplets18M contains more massive triplets (1M vs. 18M). Furthermore, since our caption triplets are synthesized based on keywords, SynthTriplets18M covers more diverse keywords than IP2P (47k vs. 586k as shown in Table 1).

		Fashion IQ (Avg)		CIRR			CIRCO		GeneCIS
Method	Arch	R@10	R@50	R@1	R_s@1	mAP@5	mAP@10	mAP@25	R@1
CLIP + IP2P[†]	ViT-L	7.01	12.33	4.07	6.11	1.83	2.10	2.37	2.44
Previous zero-shot methods (without SynthTriplets18M)									
Pic2Word[†]	ViT-L	24.70	43.70	23.90	-	8.72	9.51	10.65	11.16
SEARLE-OTI[†]	ViT-L	27.51	47.90	24.87	53.80	10.18	11.03	12.72	-
SEARLE[†]	ViT-L	25.56	46.23	24.24	53.76	11.68	12.73	14.33	12.31
Zero-shot results with the models trained with SynthTriplets18M									
ARTEMIS	RN50	33.24	47.99	12.75	21.95	9.35	11.41	13.01	13.52
Combiner	RN50	34.30	49.38	12.82	24.12	9.77	12.08	13.58	14.93
CompoDiff	RN50	35.62	48.45	18.02	57.16	12.01	13.28	15.41	14.65
CompoDiff	ViT-L	36.02	48.64	18.24	57.42	12.55	13.36	15.83	14.88
CompoDiff [‡]	ViT-L	37.36	50.85	19.37	59.13	12.31	13.51	15.67	15.11
CompoDiff	ViT-G	**39.02**	**51.71**	**26.71**	**64.54**	**15.33**	**17.71**	**19.45**	**15.48**

Table 2: **Zero-shot CIR comparisons.** † denotes the results by the official model weight, otherwise, models are trained on SynthTriplets18M and LAION-2B (ARTEMIS and Combiner are trained solely on SynthTriplets18M, while CompoDiff is trained on both). ‡ refers to the use of both the CLIP textual encoder and T5-XL as the encoder for the text condition. The full results are shown in Appendix D.

5 Experiments

5.1 Implementation details and experiment settings

CompoDiff is built upon frozen CLIP features. We use two different CLIP models, the official CLIP ViT-L/14 (Radford et al., 2021) and CLIP ViT-G/14 by OpenCLIP (Ilharco et al., 2021), where the feature dimension of the former is 768 and the latter is 1280. In the experiments, we observe that the quality of the frozen encoders is critical to the final performance. Beyond the backbone size, we observe the text condition encoder is also important (Table 5). On top of the frozen CLIP feature, we use a simple Transformer architecture for the denoising procedure, instead of the denoising U-Net (Rombach et al., 2022). We empirically observe that our transformer architecture performs slightly better than the U-Net architecture, but is much simpler. We use the multi-head self-attention blocks as the original Transformer (Vaswani et al., 2017). We set the depth, the number of heads, and the dimensionality of each head to 12, 16, and 64, respectively. The hidden dimension of the Transformer is set to 768 and 1280 for ViT-L and ViT-G, respectively. The denoising Transformer takes two inputs: a noisy visual embedding and a time-step embedding. The conditions (*e.g.*, text, mask and image conditions) are applied only to the cross-attention layer; thereby it is computationally efficient even as the number of conditions becomes larger. Our method is similar to the "DiT block with cross-attention" by Peebles & Xie (2022), but CompoDiff handles more various conditions.

All models were trained using AdamW (Loshchilov & Hutter, 2017). We used DDIM (Song et al., 2020) for the sampling variance method. We did not apply any image augmentation but used pre-extracted CLIP image features for computational efficiency; text features were extracted on the fly as text conditions can vary in SynthTriplets18M. We report the detailed hyperparameters in Table C.1.

We evaluate the zero-shot (ZS) capability of CompoDiff on four CIR benchmarks, including FashionIQ (Wu et al., 2021), CIRR (Liu et al., 2021), CIRCO (Baldrati et al., 2023) and GeneCIS (Vaze et al., 2023). We compare CompoDiff to the recent ZS CIR methods, including Pic2Word (Saito et al., 2023) and SEARLE (Baldrati et al., 2023). We also reproduce the fusion-based methods, such as ARTEMIS (Delmas et al., 2022) and Combiner (Baldrati et al., 2022), on SynthTriplets18M and report their ZS performances. Note that the current CIR benchmarks are somewhat insufficient to evaluate the effectiveness of CompoDiff, particularly considering real-world CIR queries. More details are found in Appendix C.3. Our work is the first study that shows the impact of the dataset scale and the zero-shot CIR performances with various methods, such as our method, ARTEMIS and Combiner. Due to the page limit, the details of each task and method are in Appendix C.1 and C.2. We also compare our training strategy with the other CIR methods in terms of the complexity in Appendix C.1. In summary, we argue that our method is not specifically complex compared to the other methods. The fine-tuned performances on FashionIQ and CIRR are in Appendix D.2.

5.2 Zero-shot CIR comparisons

Table 2 shows the overview of our zero-shot CIR comparison results. CLIP + IP2P denotes the naive editing-based approach by editing the reference image with the text condition using IP2P and performing image-to-image retrieval using CLIP ViT-L/14. In the table, CompoDiff outperforms all the existing methods in all benchmarks with significant gaps. The table shows the effectiveness both of our diffusion-based CIR approach and our massive synthetic dataset. In the SynthTriplets18M-trained group, CompoDiff outperforms previous SOTA fusion-based CIR methods with a large gap, especially on CIRR and CIRCO, which focus on real-life images and complex descriptions. We also can observe that the SynthTriplets18M-trained group also enables the fusion-based methods to have the zero-shot capability competitive to the SOTA zero-shot CIR methods. More interestingly, ARTEMIS even outperforms its supervised counterpart on FashionIQ with a large gap (40.6 vs. 38.2 in average recall). We provide the full results of Table 2, the supervised results of ARTEMIS, Combiner, and CompoDiff trained on SynthTriplets18M, and more discussions in Appendix D.2.

5.3 Analysis on CompoDiff

Due to the page limitation, more analyses such as Stage 2 ablation, text condition ablation and the role of condition strength are shown in Appendix D.4.

Impact of dataset scale. We provide an ablation study on CompoDiff stage 2 by varying the size of the triplet dataset for Eq. (3). Table 3 shows the results. First, at a scale of 1M, CompoDiff trained on our 1M subset significantly outperformed the IP2P triplets. This result indicates that our dataset has a more diverse representation capability. As the size of our dataset increases, the performance gradually improves. Notably, SynthTriplets18M shows consistent performance improvements from 1M to 18.8M, where manually collecting triplets in this scale is infeasible and nontrivial. Thanks to our diversification strategy, particularly keyword-based generation, we can scale up the triplet to 18.8M without manual human labor. It is also worth noting that although the FashionIQ and CIRR scores look somewhat saturated after 10M, these scores cannot represent authentic CIR performances due to the limitations of the datasets. More detailed discussions can be found in Appendix C.3 In Table D.8, more results with Combiner and ARTEMIS are shown.

Training dataset scale	IP2P(1M)	1M	5M	10M	18.8M (proposed)
FashionIQ Avg(R@10, R@50)	27.24	31.91	38.11	**42.41**	42.33
CIRR Avg(R@1 R_s@1)	27.42	28.32	31.50	37.25	**37.83**

Table 3: **Dataset scale vs. performances.** We report zero-shot performances of CompoDiff (ViT-L) trained with different dataset scales. IP2P denotes the public 1M synthetic dataset by (Brooks et al., 2022).

Denoising step size. CompoDiff needs denoising steps to denoise noisy image embeddings. However, it is possible to obtain reliable denoised image embeddings with just a few steps as shown in Table 4. Even with only 5 iterations, our model can produce competitive results. If we use 100 steps, we have a slightly better performance (42.65 vs. 42.33), but a much slower inference time (2.02 sec vs. 0.12 sec). In the experiments, we set the denoising step size to 10.

Step	1	5	10	25	50	75	100
FashionIQ Avg(R@10, R@50)	21.52	42.17	42.33	41.42	42.45	42.61	42.65
Time (sec)	0.02	0.12	0.23	0.56	1.08	1.62	2.02

Table 4: **Performances vs. inference time by varying the number of denoising steps.** Numbers are measured on the FashionIQ validation split. CompoDiff (ViT-L) in Table 2 is equivalent to 10 steps. The inference time was measured on a single A100 GPU with a batch size of 1.

Condition text encoder. As observed by Balaji et al. (2022), using a text-oriented model in addition to the CLIP textual encoder results in improved performance of image-text tasks. Motivated by this observation, we also use both the CLIP textual encoder and the language-oriented encoder for extracting the text features of Eq. (3). In Table 5, we show the choice of the text encoder affects a lot to the performance. The details of each model are in Appendix C.5. We use both T5-XL encoder

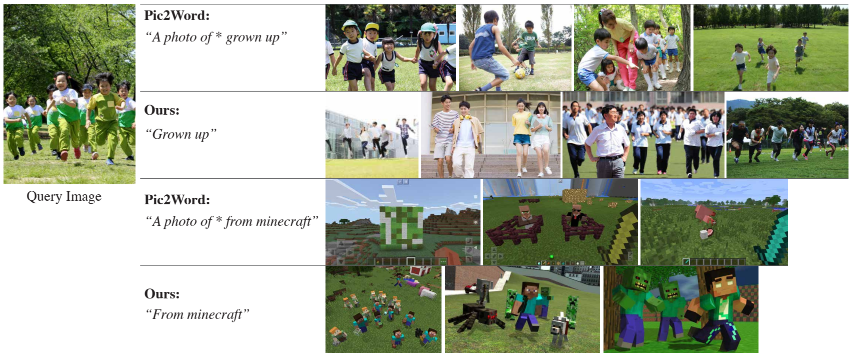


Figure 6: **Qualitative comparison of zero-shot CIR for Pic2Word and CompoDiff.** We conduct CIR on LAION. As Pic2Word cannot take a simple instruction, we made a simple modification for the given instruction.

(Raffel et al., 2020) and CLIP text encoder for the ViT-L model. For the ViT-G model, we use the CLIP text encoder only because we empirically observe that the textual representation power of the ViT-G is much more powerful than the ViT-L. Exploring the power of textual representations, such as learnable prompts to the CFG guidance, could be an interesting future work.

The choice of w_I and w_T. We include the retrieval performances by varying conditions in Fig. D.2. In summary, we can easily control the weight of conditions by w_I and w_T.

Text Enc	T5-XL	CLIP	CLIP + T5-XL
FashionIQ	38.20	42.33	**44.11**
CIRR	29.19	37.83	**39.25**

Table 5: **Text encoder impact of CompoDiff ViT-L**

5.4 Qualitative examples

We qualitatively show the versatility of CompoDiff for handling various conditions. For example, CompoDiff not only can handle a text condition, but it can also handle a *negative* text condition (*e.g.*, removing specific objects or patterns in the retrieval results), masked text condition (*e.g.*, specifying the area for applying the text condition). CompoDiff even can handle all conditions simultaneously (*e.g.*, handling positive and negative text conditions with a partly masked reference image at the same time). To show the quality of the retrieval results, we conduct a zero-shot CIR on entire LAION dataset (Schuhmann et al., 2022a) using FAISS (Johnson et al., 2019) for simulating billion-scale CIR scenarios. We show an example in Fig. 1 and more examples are shown in Appendix D.3, D.5.

Fig. 6 shows qualitative examples of zero-shot CIR results by Pic2Word and CompoDiff. CompoDiff results in semantically high-quality retrieval results (*e.g.*, understanding the "crowdedness" of the query image and the meaning of the query text at the same time). However, Pic2Word shows poor understanding of the given queries, resulting in unfortunate retrieval results (*e.g.*, ignoring "grown up" of text query, or the "crowdedness" of the query image). More examples are in Appendix D.3.

Finally, it is worth noting that CompoDiff generates a feature belonging to the CLIP space. It means unCLIP (Ramesh et al., 2022), which decodes a CLIP image feature to a high-quality image, can be applied to our composed features. We compare the retrieval results from LAION and the generated images in Appendix D.5. CompoDiff can manipulate the given input reflecting the given conditions.

6 Conclusion

In this paper, we have introduced CompoDiff, a novel diffusion-based method for solving complex CIR tasks. We have created a large and diverse dataset named SynthTriplets18M, consisting of 18.8M triplets of images, modification texts, and modified images. Our model has demonstrated impressive zero-shot CIR capabilities, as well as remarkable versatility in handling diverse conditions, such as negative text or image masks, and the controllability to enhance user experience, such as adjusting image text query weights. Furthermore, by training the previous methods on SynthTriplets18M, the models became comparable zero-shot predictors to the SOTA zero-shot methods. We strongly encourage future researchers to leverage our dataset to advance the field of CIR.

Societal Impact

Our work is primarily focused on solving complex composed image retrieval (CIR) challenges and is not designed for image editing purposes. However, we are aware that with the use of additional public resources, such as the community version of the unCLIP feature decoder (Ramesh et al., 2022), our method can potentially be utilized as an image editing method. We would like to emphasize that this unintended application is not the primary objective of our research, and we cannot guarantee the effectiveness or safety of our method in this context.

It is important to note that we have taken steps to mitigate potential risks associated with the unintended use of our method for image editing. For instance, we applied NSFW filters to filter out potentially malicious samples during the creation of SynthTriplets18M. Nevertheless, we recognize the need for continued research into the ethical and societal implications of AI technologies and pledge to remain vigilant about potential unintended consequences of our work.

References

Muhammad Umer Anwaar, Egor Labintcev, and Martin Kleinsteuber. Compositional learning of image-text query for image retrieval. (arXiv:2006.11149), May 2021. URL http://arxiv.org/abs/2006.11149. arXiv:2006.11149 [cs]. 13

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 8, 19

Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Conditioned and composed image retrieval combining and partially fine-tuning clip-based features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4959–4968, 2022. 1, 2, 3, 7, 13, 16, 24

Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Alberto Del Bimbo. Zero-shot composed image retrieval with textual inversion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2, 3, 7, 16, 18, 19

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. 2, 4, 6, 8, 14, 15

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. 14

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 14, 24

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. 13

Junsuk Choe, Song Park, Kyungmin Kim, Joo Hyun Park, Dongseob Kim, and Hyunjung Shim. Face generation for low-shot learning using generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 1940–1948, 2017. 13

Sanghyuk Chun. Improved probabilistic image-text representations. *arXiv preprint arXiv:2305.18171*, 2023. 18

Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio De Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 18

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 *European Conference on Computer Vision (ECCV)*, 2022. 18, 24

Ginger Delmas, Rafael S Rezende, Gabriela Csurka, and Diane Larlus. Artemis: Attention-based retrieval with text-explicit matching and implicit similarity. In *International Conference on Learning Representations*, 2022. 2, 3, 7, 16, 24

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 13

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021. 14

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 6, 14

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2, 3, 4, 5, 13

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 4

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 6

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, July 2021. URL https://doi.org/10.5281/zenodo.5143773. If you use this software, please cite it as below. 7

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 9, 16

Donghoon Lee, Jiseob Kim, Jisu Choi, Jongmin Kim, Minwoo Byeon, Woonhyuk Baek, and Saehoon Kim. Karlo-v1.0.alpha on coyo-100m and cc15m. https://github.com/kakaobrain/karlo, 2022. 24

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryiAv2xAZ. 13

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*, pp. 740–755. Springer, 2014. 18

Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2125–2134, 2021. 1, 2, 3, 6, 7, 16, 18

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7, 19

Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7086–7096, 2022. 5

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*, pp. 681–699. Springer, 2020. 18

Nithin Gopalakrishnan Nair, Wele Gedara Chaminda Bandara, and Vishal M. Patel. Unite and conquer: Cross dataset multimodal synthesis using diffusion models. (arXiv:2212.00793), Dec 2022. URL http://arxiv.org/abs/2212.00793. arXiv:2212.00793 [cs]. 4

Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang. Crisscrossed captions: Extended intramodal and intermodal semantic similarity judgments for ms-coco. *arXiv preprint arXiv:2004.15020*, 2020. 24

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 7

Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13018–13028, June 2021. 18

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021. 7, 13

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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 9, 19

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 4, 9, 10, 14, 24, 25

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. 2, 3, 4, 6, 7, 13, 15

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. *arXiv preprint arXiv:2302.03084*, 2023. 2, 3, 7, 16

Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific reports*, 9(1): 16884, 2019. 13

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022a. 3, 9, 14

Christoph Schuhmann, Andreas Köpf, Richard Vencu, Theo Coombes, and Romain Beaumont. Laion coco: 600m synthetic captions from laion2b-en. `https://huggingface.co/datasets/laion/laion-coco`, 2022b. 14

Jordan Shipard, Arnold Wiliem, Kien Nguyen Thanh, Wei Xiang, and Clinton Fookes. Boosting zero-shot classification with synthetic data diversity via stable diffusion. (arXiv:2302.03298), Feb 2023. URL `http://arxiv.org/abs/2302.03298`. arXiv:2302.03298 [cs]. 4

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 4, 7

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. 3, 18

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 7, 14

Sagar Vaze, Nicolas Carion, and Ishan Misra. Genecis: A benchmark for general conditional image similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2, 7, 18

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6439–6448, 2019. 13, 16

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11307–11317, 2021. 1, 2, 3, 6, 7, 16, 18

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. 13

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023. 18

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 6, 14

Appendix

In this additional document, we provide more details of the related works (§A.1) and diffusion model background (§A.2) in Appendix A. we describe the details of our data generation process in Appendix B, including the keyword-based caption generation process (Appendix B.1), the LLM-based caption generation process (Appendix B.2), the triplet generation process by the generated captions (Appendix B.3), the CLIP-based filtering process (Appendix B.4), the statistics of our generated dataset (Appendix B.5), and example samples from SynthTriplets18M (Appendix B.6). Appendix C contains more experimental details of CompoDiff, such as the details of comparisons methods (Appendix C.1), CIR tasks (Appendix C.2), the limitation of the current CIR benchmarks as zero-shot CIR evaluation (Appendix C.3), implementation details (Appendix C.4) and text encoder details (Appendix C.5). Finally, we provide more experimental results in Appendix D, including the inference time comparisons (Appendix D.1) the full results (Appendix D.2), more qualitative examples (Appendix D.3), ablation study (Appendix D.4), and retrieved examples from LAION and generation examples by using the unCLIP generator (Appendix D.5).

A Background and Related Works

A.1 More related works

We continue the discussion of the previous composed image retrieval methods. The conventional fusion methods combine image and text features from separate visual and text encoders – See Fig. 2 (a). For example, Vo et al. (2019) and Yu et al. (2020) used CNN and RNN, and Chen et al. (2022) and Anwaar et al. (2021) used CNN and Transformer. More recent methods address CIR tasks by leveraging external knowledge from models pre-trained on large-scale datasets such as CLIP (Radford et al., 2021). For example, Combiner (Baldrati et al., 2022) fine-tunes the CLIP text encoder on the triplet dataset to satisfy the relationship of $z_i = z_{i_R} + z_c$, and then trains a Combiner module on top of the encoders.

We also continue the discussion of the previous training data generation methods in the era of diffusion models. In contrast to previous attempts to synthesize training data points by GAN (Choe et al., 2017; Lee et al., 2018; Sandfort et al., 2019), recent diffusion model-based approaches show high image quality and high controllability by textual prompts (*e.g.*, by classifier-free guidance (Ho & Salimans, 2022) and latent diffusion (Rombach et al., 2022)).

A.2 Preliminary: Diffusion models

Given a data sample from the data distribution ($x \sim p(x)$) diffusion model defines a Markov chain of latent variables $z_1, \ldots, z_T$ as: $q(z_t | z_{t-1}) = \mathcal{N}(z_t; \sqrt{\alpha_t} z_{t-1}, (1 - \alpha_t) I)$. It is proven that (1) the posterior $q_{z_{t-1}|z_t}$ is approximated to a Gaussian distribution with a diagonal covariance, and (2) if the size of chain T is sufficiently large, then z_T will be approximated to $\mathcal{N}(0, I)$. In other words, DMs are a general probabilistic method that maps an input distribution $p(x)$ to a normal distribution $\mathcal{N}(0, I)$, without any assumption on the input data x. From this observation, Rombach et al. (2022) proposed to learn DMs on a latent space (*i.e.*, latent diffusion). This idea brings a huge improvement in the efficiency of DMs. In practice, we train a denoising module $\epsilon_\theta(z_t, t)$ (*i.e.*, $z_{t-1} = \epsilon_\theta(z_t, t)$) where the timestamp t is conditioned by employing time embeddings e_t.

The recent DMs use classifier-free guidance (CFG) (Ho & Salimans, 2022) for training conditional DMs without a pre-trained classifier (Dhariwal & Nichol, 2021). The simplicity of CFG extends the usage of DMs beyond mapping an input to a normal distribution; we can generate an instance with language guidance, *i.e.*, text prompt. Using CFG, recent DMs focus on text-to-image generation, while the text condition is given by CFG. In this paper, we provide various conditions via CFG to handle various composed queries. Using CFG has two advantages over the fusion-based CIR approaches. First, we can easily control the intensity of the condition. Second, we can extend the conditions beyond a single text condition, *e.g.*, a negative text condition or a mask condition.

CompoDiff is inspired by the approach of latent diffusion, and therefore, CompoDiff and StableDiffusion share some similarities. For example, their diffusion processes are based on feature spaces. However, CompoDiff has distinct differences from StableDiffusion, with three key contributions. First, StableDiffusion performs the diffusion process on 64 × 64-dimensional latent space where the latent is the embedding space of VQ-GAN (Esser et al., 2021). On the other hand, the diffusion process of CompoDiff is performed on the CLIP image latent embedding space (*e.g.*, 1024-dim feature for ViT-L). Therefore, the edited features by CompoDiff can be directly used for retrieval on the CLIP space, while StableDiffusion cannot. The Dall-e2 prior model (Ramesh et al., 2022) performs diffusion process on the CLIP embedding space, but CompoDiff takes different inputs and conditions with the Dalle-2 prior as described in Section 3.

As the second contribution, CompoDiff uses a different architecture for the de-noising module. While StableDiffusion and other conventional models are based on U-Net structure, CompoDiff uses a simple Transformer (Vaswani et al., 2017). Moreover, CompoDiff is designed to handle multiple conditions, such as masked conditions, and a triplet relationship. StableDiffusion cannot handle the localized condition and is designed for a pairwise relationship (*e.g.*, text-to-image generation). CompoDiff also handles the condition different from Dalle-2 prior. Dalle-2 prior handles conditions as the input of the diffusion model, but our CompoDiff diffusion Transformer takes the conditions via the cross-attention mechanism. This design choice makes the inference speed of CompoDiff faster. If the conditions are concatenated to the input tokens, the inference speed will be highly degenerated. Table D.6 shows that the structure taking all conditions as the concatenated input (*e.g.*, Dalle-2 prior-like) is three times slower than our cross-attention approach.

B Dataset Construction Details for SynthTriplets18M

B.1 Keyword-based diverse caption generation

As the first approach to generating caption triplets, we collect captions from the existing caption datasets and modify the captions by replacing the object terms in the captions, *e.g.*, ⟨"a strawberry tart is ...", "covert strawberry to pak choi", "a pak choi tart is ..."⟩ in Fig. 5. For the caption dataset, We use the captions from COYO 700M (Byeon et al., 2022), StableDiffusion Prompts[2] (user-generated prompts that make the quality of StableDiffusion better), LAION-2B-en-aesthetic (a subset of LAION-5B (Schuhmann et al., 2022a)) and LAION-COCO datasets (Schuhmann et al., 2022b) (synthetic captions for LAION-5B subsets with COCO style captions (Chen et al., 2015). LAION-COCO less uses proper nouns than the real web texts).

We extract the object terms from the captions using a part-of-speech (POS) tagger, provided by `Spacy`[3]. After frequency filtering, we have 586k unique object terms. For each caption, we replace the object term with other similar keywords by using the CLIP similarity score. More specifically, we extract the textual feature of keywords using the CLIP ViT-L/14 text encoder, and we choose an alternative keyword from keywords that have a CLIP similarity between 0.5 and 0.7. By converting the original object to a similar object, we have caption pairs of $\langle x_{t_R}, x_t \rangle$.

Using the caption pair $\langle x_{t_R}, x_t \rangle$, we generate the modification instruction text x_{c_T} based on a randomly chosen template from 48 pre-defined templates shown in Table B.1. After this process, we have the triplet of $\langle x_{t_R}, x_c, x_t \rangle$. We generate ≈30M triplets by the keyword-based method.

B.2 Amplifying InstructPix2Pix (IP2P) triplets by LLM

We also re-use the generated $\langle x_{t_R}, x_c, x_t \rangle$ by IP2P. We amplify the number of IP2P triplets by fine-tuning a large language model, OPT-6.7B (Zhang et al., 2022), on the generated 452k caption triplets provided by Brooks et al. (2022). Using the fine-tuned OPT, we generate ≈30M triplets.

B.3 Triplet generation from caption triplets

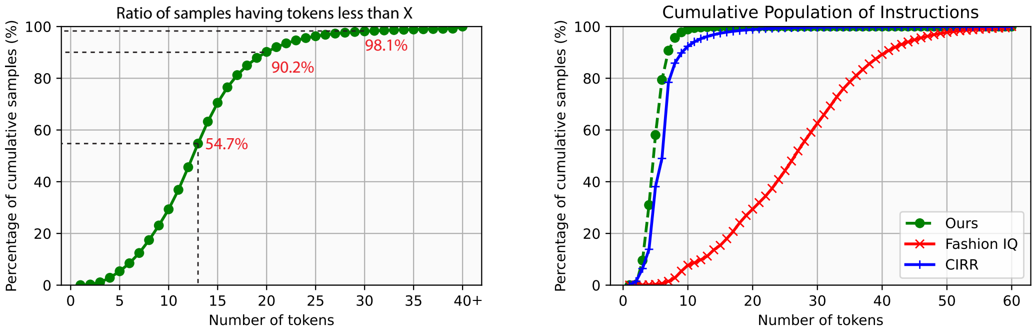
We generate 60M caption triplets $\langle x_{t_R}, x_c, x_t \rangle$ by the keyword-based generation process and the LLM-based generation process Using the triplets, we generate 60M $\langle x_{i_R}, x_c, x_i \rangle$ by using state-of-the-art text-to-image generation models. Note that two captions do not have a guarantee to be semantically similar. Following Brooks et al. (2022), we generate images using Prompt-to-Prompt (Hertz et al., 2022), which aims to generate similar images for multiple generations by sharing cross-attention weights during the denoising steps of diffusion models.

[2] `https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts`

[3] `https://spacy.io/`

Templates to change `${source}` to `${target}`	
"replace `${source}` with `${target}`"	*"substitute `${target}` for `${source}`"*
"change `${source}` to `${target}`"	*"`${target}`"*
"apply `${target}`"	*"add `${target}`"*
"exchange `${source}` with `${target}`"	*"alter `${source}` to `${target}`"*
"convert `${source}` to `${target}`"	*"transform `${source}` into `${target}`"*
"swap `${source}` for `${target}`"	*"replace `${source}` with `${target}`"*
"remodel `${source}` into `${target}`"	*"redesign `${source}` as `${target}`"*
"update `${source}` to `${target}`"	*"revamp `${source}` into `${target}`"*
"if it is `${target}`"	*"substitute `${target}` for `${source}`"*
"modify `${source}` to become `${target}`"	*"turn `${source}` into `${target}`"*
"alter `${source}` to match `${target}`"	*"customize `${source}` to become `${target}`"*
"adapt `${source}` to fit `${target}`"	*"upgrade `${source}` to `${target}`"*
"change `${source}` to match `${target}`"	*"tweak `${source}` to become `${target}`"*
"amend `${source}` to fit `${target}`"	*"`${target}` is the new option"*
"choose `${target}` instead"	*"`${target}` is the updated version"*
"use `${target}` from now on"	*"`${target}` is the new choice"*
"opt for `${target}`"	*"`${target}` is the updated option"*
"`${target}` is the new selection"	*"`${target}` is the new option available"*
"`${target}` is the updated choice"	*"`${source}` is replaced with `${target}`"*
"`${source}` is removed and `${target}` is added"	*"`${target}` is introduced after `${source}` is removed"*
"`${source}` is removed and `${target}` takes its place"	*"`${target}` is added after `${source}` is removed"*
"`${source}` is removed and `${target}` is introduced"	*"`${target}` is added in place of `${source}`"*
"`${target}` is introduced after `${source}` is retired"	*"`${target}` is added as a replacement for `${source}` is removed"*
"`${target}` is introduced as the new option after	

Table B.1: **The full 48 keyword converting templates.**



(a) **Statistics of SynthTriplets18M captions.** We show the population of our captions by the number of tokens per caption. We include captions having larger than 40 tokens in "40+".

(b) **Statistics of instructions of the CIR datasets.** We show the population of instruction captions (*e.g.*, "change A to B") by the number of tokens. We include captions having larger than 60 tokens in "60".

Figure B.1: **Statistics of SynthTriplets18M.**

B.4 CLIP-BASED FILTERING

We employ multiple state-of-the-art text-to-image (T2I) generation models, including StableDiffusion (SD) 1.5 (Rombach et al., 2022), SD 2.0, SD 2.1, and SD anime models to generate diverse images not biased towards a specific model. On the other hand, Brooks et al. (2022) only employs SD 1.5 for generating images, resulting in a lack of diversity of images compared to our dataset.

We apply a filtering process following Brooks et al. (2022) to remove the low-quality $\langle x_{i_R}, x_c, x_i \rangle$. We filter the generated images for an image-image CLIP threshold of 0.70 to ensure that the images are not too different, an image-caption CLIP threshold of 0.2 to ensure that the images correspond to their captions, and a directional CLIP similarity of 0.2 to ensure that the change in before/after captions correspond with the change in before/after images. Additionally, for keyword-based data generation, we filter out for a keyword-image CLIP threshold of 0.20 to ensure that images contain the context of the keyword, and for instruction-based data generation, we filter out for an instruction-modified image CLIP threshold of 0.20 to ensure consistency with the given instructions.

The filtering process prevents low-quality triplets, such as broken images. For example, if a modified caption is really ridiculous to generate a corresponding image, then StableDiffusion and Prompt2Prompt will not be able to generate proper images. These images will be filtered out by CLIP similarity because we measure the similarity between the generated images is sufficiently high, and the broken images will have low similarities with clean images.

After the filtering, we have 11.4M $\langle x_{i_R}, x_c, x_i \rangle$ from the keyword-based generated captions and 7.4M $\langle x_{i_R}, x_c, x_i \rangle$ from the LLM-based generated captions. It implies that the fidelity of our keyword-based method is higher than OPT fine-tuning in terms of T2I generation. As a result, SynthTriplets18M contains 18.8M synthetic $\langle x_{i_R}, x_c, x_i \rangle$. Examples of our dataset are shown in Appendix B.6. **As our dataset is larger than 4.1TB, it is impossible to serve the dataset while keeping anonymity. We provide a small subset of images in the supplementary materials for the reviewers, and the dataset will be hosted through the HuggingFace hub.**

B.5 Dataset Statistics

We show the statistics of our generated caption dataset (*i.e.*, before T2I generation, x_{t_R} and x_t). We use the CLIP tokenizer to measure the statistics of the captions. Fig. B.1a shows the cumulative ratio of captions with tokens less than X. About half of the captions have less than 13 tokens, and 90% of the captions have less than 20 tokens. Only 0.8% of the captions have more than 40 tokens.

We also compare our dataset, FashionIQ (Wu et al., 2021) and CIRR (Liu et al., 2021) in terms of the token statistics of instructions (*i.e.*, x_c). Fig. B.1b shows that our dataset has relatively shorter instructions than other human-annotated instructions. We presume that this is why CompoDiff performs better when fine-tuning on the target dataset.

B.6 Example samples of SynthTriplets18M

We illustrate example samples of SynthTriplets18M in Fig. B.2. Our dataset can express the change of overall context (*e.g.*, “make the landscape a cityscape”), the seasonal change (*e.g.*, “make it sprint”), the change of mood (*e.g.*, “make it a watercolor”), and the change of local objects (*e.g.*, “have the person be a dog”). The full dataset will be hosted through the HuggingFace hub.

C More Experimental Details

C.1 Comparison methods.

We compare CompoDiff with four state-of-the-art CIR methods as follows:

Combiner (Baldrati et al., 2022) involves a two-stage training process. First, the CLIP text encoder is fine-tuned by contrastive learning of z_c and $z_{i_R} + z_c$ in the CLIP embedding space. The second stage replaces $z_{i_R} + z_c$ to the learnable Combiner module. Only the Combiner module is trained during the second stage. Hence, its image embedding space is the original CLIP space as CompoDiff.

ARTEMIS (Delmas et al., 2022) optimizes two similarities simultaneously. The implicit similarity is computed between the combined feature of z_i and z_c, and the combined one of z_{i_R} and z_c. The explicit matching is computed between z_i and z_c. ARTEMIS suffers from the same drawback as previous CIR methods, *e.g.* TIRG (Vo et al., 2019): As it should compute combined feature of z_i and z_c, it is not feasible to use an approximate nearest neighbor search algorithm, such as FAISS (Johnson et al., 2019). This is not a big problem in a small dataset like FashionIQ, but it makes ARTEMIS infeasible in real-world scenarios, *e.g.*, when the entire LAION-2B is the target database.

Pic2Word (Saito et al., 2023) projects a visual feature into text embedding space, instead of combining them. Pic2Word performs a text-to-image retrieval by using the concatenated feature as the input of the CLIP textual encoder. As the projection module is solely trained on cheap paired datasets without expensive triplet datasets, it is able to solve CIR in a zero-shot manner.

SEARLE (Baldrati et al., 2023) is a similar to Pic2Word, while SEARLE employes text-inversion task instead of projection. Baldrati et al. (2023) also proposed a technique, named Optimization-based Textual Inversion (OTI), a GPT-powered regularization method. In this paper, we compare SEARLE-XL and SEARLE-XL-OTI, which use the ViT-L/14 CLIP backbone for a fair comparison.

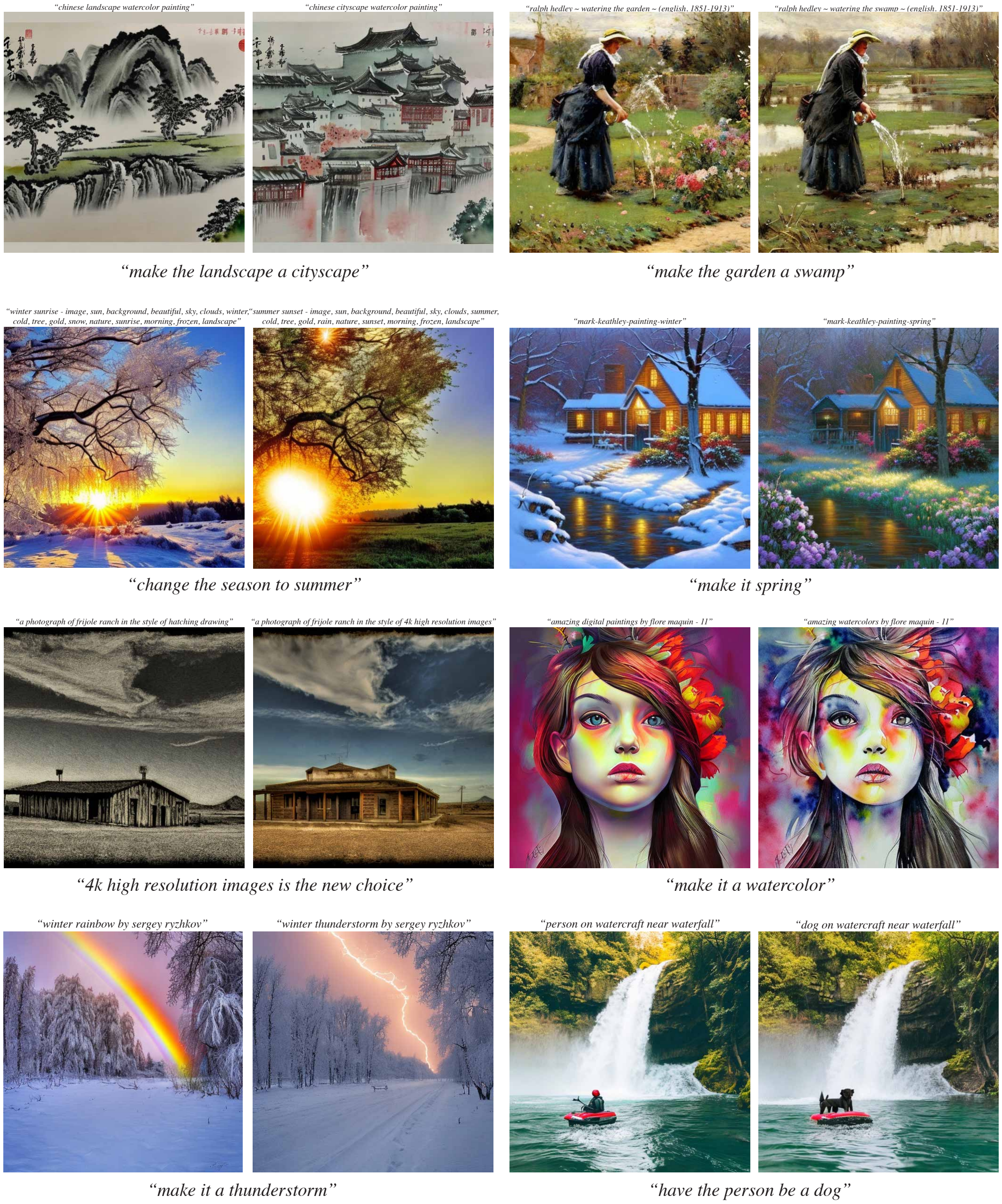

"chinese landscape watercolor painting"
"chinese cityscape watercolor painting"
"ralph hedley ~ watering the garden ~ (english, 1851-1913)"
"ralph hedley ~ watering the swamp ~ (english, 1851-1913)"
"make the landscape a cityscape"
"make the garden a swamp"
"winter sunrise - image, sun, background, beautiful, sky, clouds, winter, cold, tree, gold, snow, nature, sunrise, morning, frozen, landscape"
"summer sunset - image, sun, background, beautiful, sky, clouds, summer, cold, tree, gold, rain, nature, sunset, morning, frozen, landscape"
"mark-keathley-painting-winter"
"mark-keathley-painting-spring"
"change the season to summer"
"make it spring"
"a photograph of frijole ranch in the style of hatching drawing"
"a photograph of frijole ranch in the style of 4k high resolution images"
"amazing digital paintings by flore maquin - 11"
"amazing watercolors by flore maquin - 11"
"4k high resolution images is the new choice"
"make it a watercolor"
"winter rainbow by sergey ryzhkov"
"winter thunderstorm by sergey ryzhkov"
"person on watercraft near waterfall"
"dog on watercraft near waterfall"
"make it a thunderstorm"
"have the person be a dog"

Figure B.2: **Examples of SynthTriplets18M.** We show examples of $\langle x_{i_R}, x_c, x_i \rangle$, *i.e.*, {original image, modification instruction, and modified image}, as well as the generation prompt for x_{i_R} and x_i.

We also compare a naive editing-based retrieval method: editing an image using IP2P with an instruction and retrieving images using the CLIP image encoder. Note that Pic2Word and SEARLE are trained on the image-only datasets, therefore, it is impossible to train them on SynthTriplets18M. Therefore, we only train the previous fusion-based methods, ARTEMIS and Combiner on SynthTriplets18M for comparison.

We would like to emphasize that the two-staged training procedure is not a too complex strategy. For example, SEARLE employs two-staged training, where the first stage is for learning pseudo token embeddings, and the second stage is for learning the projection module by distillation using the learned pseudo token embeddings. Combiner also employs two-staged training, where the first stage is a pre-training stage for the text encoder and the second stage is for training the combiner module using the pre-trained model. Our two-staged training strategy is not a mandatory procedure, but we employ two-staged training for better performance. Conceptually, stage 1 is for a pre-training by training text-to-image generation diffusion model with pairwise relationships. The stage 2 is for a fine-tuning process using triplet relationships. Lastly, the alternative optimization strategy for stage 2 is for helping optimization, not for resolving the instability. As shown in Table D.6, CompoDiff can be trained only with Eq. (3), but adding more objective functions makes the final performances stronger. In terms of diffusion model fine-tuning, we argue that our method is not specifically complex than other fine-tuning methods, such as ControlNet (Zhang et al., 2023).

We partially agree with the reviewer, that both stages are trained with massive data points. However, as shown in Table 4, the massive data points are not the necessary condition for training CompoDiff. In fact, Table 4 shows that CompoDiff follows scale-law, namely, CompoDiff performances are consistently improved by scaling up the data points. As far as we know, this is the first study shows the impact of the dataset scale to the zero-shot CIR performances.

C.2 CIR Datasets.

FashionIQ. FashionIQ, the most popular CIR benchmark, has (46.6k / 15.5k / 15.5k) (training / validation / test) images with three fashion categories: Shirt, Dress, and Toptee. Each category has 18k training triplets and 12k evaluation triplets of $\langle x_{i_R}, x_c, x_i \rangle$. Examples of x_c looks like: “is more strappy and emerald”, “is brighter” or “is the same”. The main drawback of FashionIQ is that the dataset is limited to the specific subsets of fashion domains, hence it is not possible to evaluate whether the methods are truly useful for real-world CIR tasks.

CIRR. CIRR contains more generic images than FashionIQ. CIRR uses the images from NLVR^2 (Suhr et al., 2018) with more complex and long descriptions. CIRR has 36k open-domain triplets divided into the train, validation, and test sets in 8:1:1 split. The examples of x_c are “remove all but one dog and add a woman hugging it”, “It’s a full pizza unlike the other with the addition of spice bottles”, or “Paint the counter behind the brown table white”. As the example captions show, one of the main drawbacks of CIRR is that the text queries are not realistic compared to the real user scenario; we may need more shorter and direct instructions. As the text instruction distribution of CIRR is distinct from others (Fig. B.1b), we observe that sometimes CIRR results are a not reliable measure of open-world CIR tasks (*e.g.*, retrieval on LAION). Baldrati et al. (2023) also observe another main drawback of CIRR. CIRR contains a lot of false negatives (FNs) due to the nature of the image-text dataset (Chun et al., 2021; 2022; Chun, 2023)

CIRCO. To tackle the FN issue of CIRR, Baldrati et al. (2023) introduces the CIRCO dataset. This dataset comprises of 1020 queries, where 220 and 800 of them are used for validation and test, respectively. The images are from the COCO dataset (Lin et al., 2014), where the size of the image database is 120K COCO images. Example x_c of CIRCO includes “has two children instead of cats” or ”is on a track and has the front wheel in the air“. We use mAP scores following Baldrati et al. (2023) which is known to be a robust retrieval metric (Musgrave et al., 2020; Chun et al., 2022).

GeneCIS. GeneCIS (Vaze et al., 2023) is a dataset to evaluate different conditional similarities of four categories: (1) focus on an attribute, (2) change an attribute, (3) focus on an object, and (4) change an object. The first two categories are based on the VAW dataset (Pham et al., 2021), which contains massive visual attributes in the wild. The other categories are sampled from COCO.

C.3 The limitations of the current CIR benchmarks

One of the significant drawbacks of the existing CIR benchmarks, such as FashionIQ (Wu et al., 2021) and CIRR (Liu et al., 2021), is the domain-specific characteristics that cannot be solved without training on the datasets. For example, the real-world queries that we examine are mainly focused on small editing, addition, deletion, or replacement. However, because the datasets are constructed by letting human annotators write a modification caption for a given two images, the text conditions are somewhat different from the real-world CIR queries. For example, CIRR dev set has some text conditions like: “show three bottles of soft drink” (different from the common real-world CIR text conditions), “same environment different species” (ambiguous condition), “Change the type of dog

	Stage1	Stage2	Fine-tuning
Diffusion steps	1000	1000	1000
Noise schedule	cosine	cosine	cosine
Sampling steps	10	10	10
Sampling variance method	DDIM	DDIM	DDIM
Dropout	0.1	0.1	0.1
Weight decay	6.0e-2	6.0e-2	6.0e-2
Batch size	4096	2048	2048
Iterations	1M	200K	50K
Learning rate	1e-4	1e-5	1e-5
Optimizer	AdamW	AdamW	AdamW
EMA decay	0.9999	0.9999	0.9999
Input tokens	$z_i^{(t)}$, t	$z_i^{(t)}$, t	$z_i^{(t)}$, t
Conditions	z_{c_T}	z_{c_T}, z_{i_R}, z_{c_M}	z_{c_T}, z_{i_R}
Training dataset	LAION-2B	LAION-2B, SynthTriplets18M	FashionIQ or CIRR trainset
Image encoder	CLIP-L/14	CLIP-L/14	CLIP-L/14
Text encoder	CLIP-L/14	CLIP-L/14	CLIP-L/14
Denoiser depth	12	12	12
Denoiser heads	16	16	16
Denoiser head channels	64	64	64

Table C.1: **Hyperparameters.** A model trained by Stage 1 and Stage 2 is equivalent to "Zero-shot" in the main table. A "supervised model" is the same as the fine-tuned version.

and have it walking to the left in dirt with a leash." (multiple conditions at the same time). These types of text conditions are extremely difficult to solve in a zero-shot manner, but we need access to the CIRR training dataset. Instead, we believe that the CIRCO dataset (Baldrati et al., 2023) is a better benchmark than FashionIQ and CIRR. It is because CIRCO aims to resolve the false negatives of FashionIQ and CIRR, focusing on the retrieval evaluation.

Furthermore, the existing benchmarks (even more recent ones, such as CIRCO and GeneCIS) cannot evaluate the negative and mask conditions. In practice, when we perform a qualitative study on a large image index (*i.e.*, LAION-2B image index), we observe that CompoDiff outperforms previous methods, such as Pic2Word, in terms of the retrieval quality (See Fig. 6 and Fig. D.1 for examples). Also, we observe that even if the CIR scores are similar (*e.g.*, 10M and 18.8M in Table 3), in the LAION-2B index, the retrieval quality can be significantly better. Unfortunately, it is impossible to evaluate the quality of retrieval results on the LAION-2B, because a quantitative study requires expensive and infeasible human verification.

C.4 Implementation details

We report the detailed hyperparameters in Table C.1. All models were trained using AdamW (Loshchilov & Hutter, 2017) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For computationally efficient training, we precomputed CLIP visual embeddings of the entire image from our training dataset. Since our training dataset was sufficiently large, we did not use any image augmentation methods to extract CLIP visual embeddings. Since the text as a condition can vary each time in training according to the 48 templates (Table B.1), we do not precompute any textual embeddings. In the case of keyword-based generated triplets, we are able to randomly switch query and modified images during training because the instruction for keyword-based triplets is generated according to the 48 templates.

C.5 Text encoder details

As shown Balaji et al. (2022), using a text-oriented model such as T5 (Raffel et al., 2020) in addition to the CLIP textual encoder results in improved performance of text-to-image generation models. Motivated by this observation, we also use both the CLIP textual encoder and the language-oriented encoder. We also observed the positive effect of the text-oriented model and experiment results showed that T5-XL, which has 3B parameters, could improve the performance by a large margin in the overall evaluation metrics. As described in Appendix C.4, all training text embeddings are extracted at every iteration. To improve computational efficiency, we reduced the number of input tokens of the T5 models to 77, as in CLIP (as shown in Fig. B.1a and Fig. B.1b, most of the captions in our dataset have lengths less than 77). A single-layer perceptron was employed to align the dimension of text embeddings extracted from T5 XL with that of CLIP large.

In Table 5, when the CLIP textual encoder and the T5-XL were used together, the experimental results improved by a large margin. We suspect that this is because the strong T5 encoder can help the CLIP text encoder to better understand given captions. Interestingly, we observe that using T5 alone degrades the performance even compared to using the CLIP textual encoder alone. We suspect that this is because T5-XL is specified for long text sequences (*e.g.*, larger than 100 tokens) and text-only data. On the other hand, our caption dataset has an extremely short average length (see Fig. B.1a and Fig. B.1b), which is not specialized by T5. Also, our dataset is based on captions, paired with an image; we also need to consider image information to understand the given caption, but we cannot handle image information alone with T5.

D MORE EXPERIMENTAL RESULTS

D.1 INFERENCE TIME COMPARISON

In this subsection, we compare the inference speed of the CIR methods. In Table D.1, we can confirm that CompoDiff is practically useful with high throughput (120ms) for a single image.

One of the advantages of our method is that we can control the trade-off between retrieval performance and inference time. Note that it is impossible for the other methods. If we need a faster inference time, even with a worse retrieval performance, we can reduce the number of diffusion steps. More detailed experiments for the trade-off are shown in Table 4.

Our another contribution is a novel and efficient conditioning for the diffusion model. Instead of using a concatenated vector of all conditions and inputs as the input of the diffusion model, we use the cross-attention mechanism for conditions and leave the input size the same as the original size. As shown in Table D.6, our design choice is three times faster than the naive implementation.

Method	Backbone	Inference time (secs)
Pic2Word	ViT-L	0.02
SEARLE	ViT-L	0.02
ARTEMIS	RN50	0.005
Combiner	RN50	0.006
CompoDiff	ViT-L	0.12

Table D.1: **Inference time comparisons.** Numbers are measured with an A100 GPU for a single image.

D.2 FULL EXPERIMENT RESULTS OF TABLE 2

In this subsection, we report the full experiment results of Table 2. For FashionIQ and CIRR datasets, we also report supervised fine-tuning results of SynthTriplets18M-trained CIR methods.

Table D.2 shows the comparison of CompoDiff with baselines on the FashionIQ dataset. Following the standard choice, we use recall@K as the evaluation metric. "Zero-shot" means that the models are not trained on FashionIQ. ARTEMIS and Combiner were originally designed for the supervised setting, but, we trained them on SynthTriplets18M for a fair comparison with our method. Namely, we solely train them on our dataset for the zero-shot benchmark and fine-tune the zero-shot weights on the FashionIQ training set for the supervised benchmark.

Table D.3 shows the CIRR results and all experimental settings were identical to FashionIQ. Similar to FashionIQ, CompoDiff also achieves a new state-of-the-art CIRR zero-shot performance. It is noteworthy that Combiner performs great in the supervised setting but performs worse than CompoDiff in the zero-shot setting. We presume that the fine-tuned Combiner text encoder is overfitted to long-tailed CIRR captions. It is partially supported by our additional experiments on text encoder in Section 5.3; a better understanding of complex texts provides better performances.

	Shirt		Dress		Toptee		Average		
Method	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50	Avg.
Previous zero-shot methods (without SynthTriplets18M)									
CLIP + InstructPix2Pix	7.24	12.71	6.31	10.42	7.49	13.85	7.01	12.33	9.67
Pic2Word	20.00	40.20	26.20	43.60	27.90	47.40	24.70	43.70	34.20
SEARLE-XL-OTI	30.37	47.49	21.57	44.47	30.90	51.76	27.61	47.90	
SEARLE-XL	26.89	45.58	20.48	43.13	29.32	49.97	25.56	46.23	
Zero-shot results with the models trained with SynthTriplets18M									
ARTEMIS†	30.70	50.43	33.52	46.54	35.49	47.01	33.24	47.99	40.62
CLIP4Cir†	32.32	51.65	34.92	48.38	35.65	48.10	34.30	49.38	41.84
CompoDiff (ViT-L)†	37.69	49.08	32.24	46.27	38.12	50.57	36.02	48.64	42.33
CompoDiff ‡ (ViT-L)†	38.10	52.48	33.91	47.85	40.07	52.22	37.36	50.85	44.11
CompoDiff (ViT-G)†	**41.31**	**55.17**	**37.78**	**49.10**	**44.26**	**56.41**	**39.02**	**51.71**	**46.85**
Supervised									
JVSM	12.00	27.10	10.70	25.90	13.00	26.90	11.90	26.60	19.25
CIRPLANT w/ OSCAR	17.53	38.81	17.45	40.41	21.64	45.38	18.87	41.53	30.20
TRACE w/ BERT	20.80	40.80	22.70	44.91	24.22	49.80	22.57	46.19	34.38
VAL w/ GloVe	22.38	44.15	22.53	44.00	27.53	51.68	24.15	46.61	35.38
MAAF	21.30	44.20	23.80	48.60	27.90	53.60	24.30	48.80	36.55
ARTEMIS	21.78	43.64	27.16	52.40	29.20	54.83	26.05	50.29	38.17
CurlingNet	21.45	44.56	26.15	53.24	30.12	55.23	25.90	51.01	38.46
CoSMo	24.90	49.18	25.64	50.30	29.21	57.46	26.58	52.31	39.45
RTIC-GCN w/ GloVe	23.79	47.25	29.15	54.04	31.61	57.98	28.18	53.09	40.64
DCNet	23.95	47.30	28.95	56.07	30.44	58.29	27.78	53.89	40.84
AACL	24.82	48.85	29.89	55.85	30.88	56.85	28.53	53.85	41.19
SAC w/ BERT	28.02	51.86	26.52	51.01	32.70	61.23	29.08	54.70	41.89
MUR	30.60	57.46	31.54	58.29	37.37	68.41	33.17	61.39	47.28
CLIP4Cir	39.99	60.45	33.81	59.40	41.41	65.37	38.32	61.74	50.03
Pre-training with SynthTriplets18M & Fine-tuning with FashionIQ									
ARTEMIS†	32.17	53.32	34.80	48.10	36.58	47.63	34.52	49.68	42.10
CLIP4Cir†	37.21	**60.71**	**42.75**	**60.50**	42.98	**65.49**	**40.98**	**62.23**	**51.61**
CompoDiff (ViT-L)†	40.88	53.06	35.53	49.56	41.15	54.12	39.05	52.34	46.31
CompoDiff ‡ (ViT-L)†	40.65	57.14	36.87	57.39	43.93	61.17	40.48	58.57	49.53
CompoDiff (ViT-G)†	**41.68**	56.02	38.39	51.03	**45.70**	57.32	39.81	51.90	47.73

Table D.2: **Comparisons on FashionIQ validation set.** We report two scenarios. The "Zero-shot" scenario performs a composed-image retrieval using a model not trained on the FashionIQ dataset, while models are trained on FashionIQ for the "Supervised" scenario. The last group denotes that a model is trained on CompoDiff (Section 4). † denotes that the models are newly trained by us. ‡ refers to the use of both the CLIP textual encoder and T5-XL as the encoder for the text condition.

For both datasets, we achieve state-of-the-art by fine-tuning the Combiner model trained on SynthTriplets18M to the target dataset. It shows the benefits of our dataset compared to the limited number of CIR triplet datasets.

The detailed CIRCO results are shown in Table D.4. In the table, we can observe that in all metrics, CompoDiff achieves the best performances among the zero-shot CIR methods. This result supports the effectiveness of CompoDiff and SynthTriplets18M in real-world CIR tasks.

Finally, we report detailed GeneCIS in Table D.5. In the table, CompoDiff shows the best performance in the average recall. Our method especially outperforms the other methods in "Change Attribute", "Focus Object" and "Change Object". On the other hand, CompoDiff is less effective than Pic2Word and SEARLE in "Focus attribute". We presume that it is because the instruction distribution of "Focus Attribute" differs a lot from the instruction of SynthTriplets18M. Among other CIR methods in the SynthTriplets18M-trained group, CompoDiff shows the best performances.

Method	R@1	R@5	R@10	R@50	R$_s$@1	R$_s$@2	R$_s$@3	Avg(R@1, R$_s$@1)
Previous zero-shot methods (without SynthTriplets18M)								
CLIP + InstructPix2Pix	4.07	8.41	11.2	15.38	6.11	10.05	13.33	5.09
Pic2Word	23.90	51.70	65.30	87.80	54.12	74.63	87.61	39.01
SEARLE-XL-OTI	24.87	52.31	66.29	88.58	53.80	74.31	86.94	39.34
SEARLE-XL	24.24	52.48	66.29	88.84	53.76	75.01	88.19	39.00
Zero-shot results with the models trained with SynthTriplets18M								
ARTEMIS†	12.75	33.84	47.75	80.20	21.95	43.88	62.06	17.35
CLIP4Cir†	12.82	36.83	48.19	81.91	24.12	46.47	63.07	18.47
CompoDiff (ViT-L)†	18.24	53.14	70.82	90.25	57.42	77.10	87.90	37.83
CompoDiff ‡ (ViT-L)†	19.37	53.81	72.02	90.85	59.13	78.81	89.33	39.25
CompoDiff (ViT-G)†	**26.71**	**55.14**	**74.52**	**92.01**	**64.54**	**82.39**	**91.81**	**45.63**
Supervised								
TIRG	14.61	48.37	64.08	90.03	22.67	44.97	65.14	18.64
TIRG + LastConv	11.04	35.68	51.27	83.29	23.82	45.65	64.55	17.43
MAAF	10.31	33.03	48.30	80.06	21.05	41.81	61.60	15.68
MAAF + BERT	10.12	33.10	48.01	80.57	22.04	42.41	62.14	16.08
MAAF-IT	9.90	32.86	48.83	80.27	21.17	42.04	60.91	15.54
MAAF-RP	10.22	33.32	48.68	81.84	21.41	42.17	61.60	15.82
CIRPLANT	15.18	43.36	60.48	87.64	33.81	56.99	75.40	24.50
CIRPLANT w/ OSCAR	19.55	52.55	68.39	92.38	39.20	63.03	79.49	29.38
ARTEMIS	16.96	46.10	61.31	87.73	39.99	62.20	75.67	28.48
CLIP4Cir	38.53	69.98	81.86	95.93	68.19	85.64	94.17	53.36
Pre-training with SynthTriplets18M & Fine-tuning with FashionIQ								
ARTEMIS†	18.85	51.44	68.01	91.93	38.85	62.00	77.68	28.85
CLIP4Cir†	**39.99**	**73.63**	**86.77**	**96.55**	**68.41**	**86.12**	**94.80**	**54.20**
CompoDiff (ViT-L)†	21.30	55.01	72.62	91.49	58.82	77.60	88.37	40.06
CompoDiff ‡ (ViT-L)†	22.35	54.36	73.41	91.77	62.55	81.44	90.21	42.45
CompoDiff (ViT-G)†	32.39	57.61	77.25	94.61	67.88	85.29	94.07	50.14

Table D.3: **Compasions on CIRR Test set.** Details are the same as Table D.2.

Method	mAP@5	mAP@10	mAP@25	mAP@50
Previous zero-shot methods (without SynthTriplets18M)				
CLIP + InstructPix2Pix	1.83	2.10	2.37	2.44
Pic2Word	8.72	9.51	10.65	11.29
SEARLE-XL	11.68	12.73	14.33	15.12
Zero-shot results with the models trained with SynthTriplets18M				
ARTEMIS	9.35	11.41	13.01	14.13
CLIP4Cir	9.77	12.08	13.58	14.11
CompoDiff (ViT-L)	12.55	13.36	15.83	16.43
CompoDiff ‡ (ViT-L)	12.31	13.51	15.67	16.15
CompoDiff (ViT-G)	**15.33**	**17.71**	**19.45**	**21.01**

Table D.4: **Compasions on CIRCO Test set.** Details are the same as Table D.2.

D.3 More qualitative examples

Open world zero-shot CIR comparisons with Pic2Word. We illustrate further comparisons with Pic2Word in Fig. D.1. Here, we can draw the same conclusions as in the main text: Pic2Word often cannot understand images or instructions (*e.g.*, ignores the "crowdedness" of the images, or retrieves irrelevant images such as images with a woman in the last example). All retrieved results in our paper were obtained using Pic2Word trained on the LAION 2B dataset.

Method	Focus Attribute			Change Attribute			Focus Object			Change Object			Avg
	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	R@1
Previous zero-shot methods (without SynthTriplets18M)													
Pic2Word	15.65	28.16	38.65	13.87	24.67	33.05	8.42	18.01	25.77	6.68	15.05	24.03	11.16
SEARLE-XL	**17.00**	**29.65**	**40.70**	16.38	25.28	34.14	7.96	16.94	25.61	7.91	16.79	24.80	12.31
Zero-shot results with the models trained with SynthTriplets18M													
ARTEMIS	11.76	21.97	25.44	15.41	25.14	33.10	8.08	16.77	24.70	18.84	30.53	39.98	13.52
Combiner	14.11	24.08	34.12	18.39	28.22	37.13	8.49	16.70	25.21	18.72	30.92	40.11	14.93
CompoDiff (ViT-L)	13.50	24.32	36.11	19.20	28.64	37.20	8.11	16.39	25.08	18.71	31.69	40.55	14.88
CompoDiff ‡ (ViT-L)	13.01	25.01	36.75	**19.88**	28.64	37.18	8.60	16.28	**25.85**	**18.94**	**31.80**	**40.58**	15.11
CompoDiff (ViT-G)	14.32	26.70	38.41	19.72	**28.78**	**37.39**	**9.18**	**19.11**	25.77	18.71	31.71	40.22	**15.48**

Table D.5: **Compasions on GeneCIS Test set.** Details are the same as Table D.2.

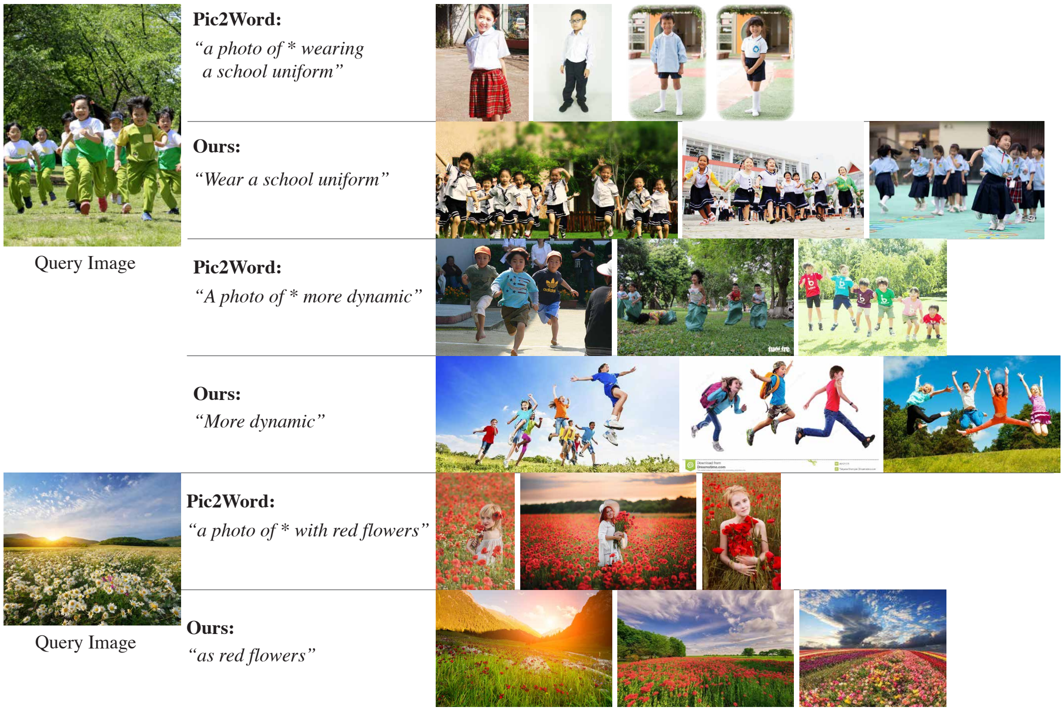

Pic2Word:
*"a photo of * wearing a school uniform"*
Ours:
"Wear a school uniform"
Query Image
Pic2Word:
*"A photo of * more dynamic"*
Ours:
"More dynamic"
Pic2Word:
*"a photo of * with red flowers"*
Query Image
Ours:
"as red flowers"

Figure D.1: **More qualitative comparison of zero-shot CIR for Pic2Word and CompoDiff.**

More versatile CIR examples on LAION. We illustrate more qualitative examples of retrieval results in Fig. D.3, Fig. D.4, Fig. D.5, and Fig. D.6. We will describe the details of "Generated by unCLIP" in the later section.

D.4 More Ablation study

Stage 2 ablation. As we described in Section 3.2, we alternatively update the model using three different objectives. Here, we conduct an ablation study of our design choice. Table D.6 shows the result. Our multi-task learning strategy improves the overall performance. It is because although SynthTriplets18M is a vast and diverse dataset, its diversity is weaker than LAION. As LAION is not used for *only* Eq. (3), we employ additional tasks using LAION, *i.e.*, Eq. (1) and Eq. (2).

How to handle text input? CompoDiff does not take CLIP textual embeddings for text guidance as input tokens of the denoising Transformer but as a condition of cross-attention. Our design choice allows for faster throughput compared to the counterpart that takes CLIP textural embeddings directly as input tokens. We compare the impact of different design choices for handling textual embeddings. First, we evaluate the "Prior" model which converts CLIP textual embeddings into CLIP

	only Eq. (3)	alternative update of Eq. (1), Eq. (2) and Eq. (3) (proposed)
FashionIQ Avg(R@10, R@50)	41.15	**42.33**
CIRR Avg(R@1 R$_s$@1)	35.25	**37.83**

Table D.6: **Stage 2 ablation.** The zero-shot performances of CompoDiff (ViT-L) trained by different objectives.

	COCO 5k			CxC			ECCV Caption			Throughput
Method	R@1	R@5	R@10	R@1	R@5	R@10	mAP@R	R-P	R@1	images/sec
Image to text retrieval										
Prior Lee et al. (2022)	32.04	56.84	67.68	33.76	60.50	71.32	14.19	23.37	46.47	497.28
Prior-like Stage1	34.32	58.40	69.52	35.01	62.35	74.21	16.35	25.20	49.01	497.28
Ours (Stage 1 only)	35.13	59.46	70.26	35.30	62.62	74.02	16.01	25.44	49.64	1475.52
Ours (Stage 1 + Stage2)	33.20	58.00	68.94	34.78	61.68	72.96	15.07	24.39	47.03	1473.92
Text to image retrieval										
Prior Lee et al. (2022)	17.05	34.25	43.16	18.62	37.43	47.07	18.10	26.46	46.40	497.28
Prior-like Stage1	22.62	38.31	48.11	21.42	41.42	51.79	20.70	29.80	51.50	497.28
Ours (Stage 1 only)	22.47	39.18	49.08	22.51	42.40	52.77	21.46	30.30	53.75	1475.52
Ours (Stage 1 + Stage2)	20.00	38.63	48.25	21.57	41.71	51.99	20.82	29.84	51.65	1473.92

Table D.7: **Comparisons of various design choices for handling textual embeddings on cross-modal retrieval results on COCO, CxC and ECCV Caption datasets.** For all metrics, higher is better.

visual embeddings and proposed in unCLIP (Ramesh et al., 2022) (we use a public community model [4] because the official model is not yet publically available). Second, we test the "Prior-like" model by using the denoising Transformer, but taking text guidance as input tokens instead of cross-attention. We also test two more CompoDiff models from our two-stage training strategy.

To measure the ability to understand raw text, we evaluate the models on image-to-text and text-to-image retrieval benchmarks on the MS-COCO Caption dataset (Chen et al., 2015). We also evaluate them on the extension of COCO Caption to mitigate the false negative problem of COCO, namely, CxC (Parekh et al., 2020) and ECCV Caption (Chun et al., 2022). Table D.7 shows the average metrics of each benchmark for image-to-text and text-to-image. In the table, we first observe that our design choice is three times faster than the "Prior-ish" counterparts by handling textual embeddings with cross-attention. Second, we observe that Stage 1 only CompoDiff shows a better understanding of image-to-caption and caption-to-image retrieval tasks. We speculate that this is because Ours (Stage 1 only) is directly optimized by the image-to-text (ITM) matching style dataset, while Ours (Stage 1 + Stage 2) is also trained with other types of conditions (*e.g.*, masks, negative texts, image conditions). Throughput was measured on a single A100 GPU with a batch size of 32.

In summary, our design choice shows × 3 faster inference time than the prior model (Ramesh et al., 2022) but better text-to-image retrieval performances on COCO.

More experiments with varying the dataset scale. Table D.8 shows the same experimental results with Table 3 with other methods, such as ARTEMIS (Delmas et al., 2022) and Combiner (Baldrati et al., 2022). In all methods, we observe a similar phenomenon. Using more data points significantly increases the performance (*e.g.*, CompoDiff FIQ shows 38.11 → 42.33 for 1M → 18.8M). Also, using a better-quality of the data points is remarkably helpful for the performance. For example, the same amount of data points (1M) by IP2P and ours show 27.24 and 31.91 FashionIQ average recalls.

Condition strength As w_I increases, the generated image embeddings become more dependent on the reference image, while increasing w_T results in a greater influence of the text guidance. How-

[4] https://huggingface.co/kakaobrain/karlo-v1-alpha

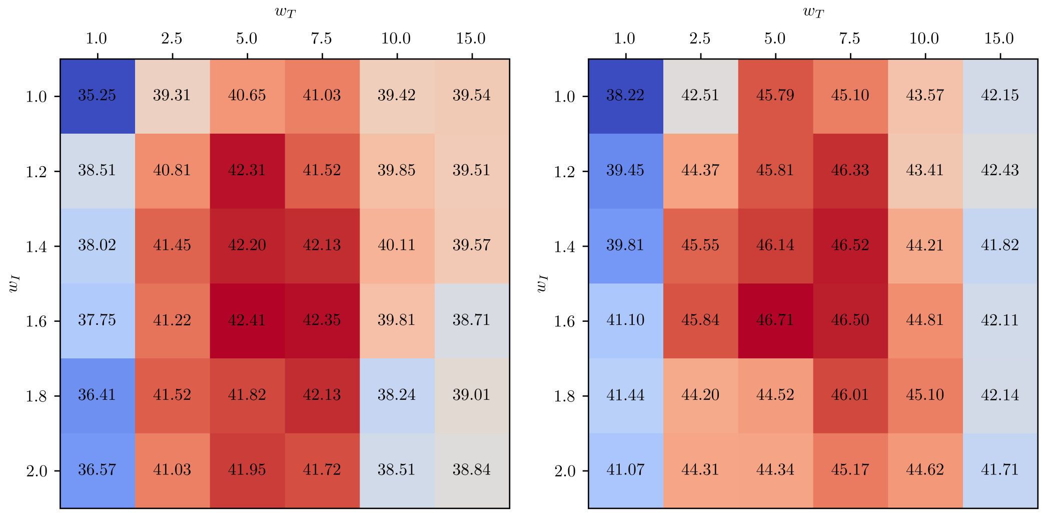
Training dataset scale	IP2P(1M)	1M	5M	10M	18.8M (proposed)
FashionIQ Avg(R@10, R@50)					
ARTEMIS	26.03	27.44	36.17	41.35	40.62
Combiner	29.83	29.64	35.23	41.81	41.84
CompoDiff	27.24	31.91	38.11	42.41	42.33
CIRR Avg(R@1, R_s@1)					
ARTEMIS	14.91	15.12	15.84	17.56	17.35
Combiner	16.50	16.88	17.21	18.77	18.47
CompoDiff	27.42	28.32	31.50	37.25	37.83

Table D.8: **Impact of dataset scale on CIR methods.** The details are the same as Table 3.

ever, large w_I and w_T are not always beneficial. If w_I or w_T is too large, it can lead to unexpected results. To find a harmonious combination of w_I and w_T, we performed a sweeping process as shown in Fig. D.2. We use w_I as 1.5 and w_T as 7.5 considering the best content-condition trade-off.

D.5 Image decoding using unCLIP generator.

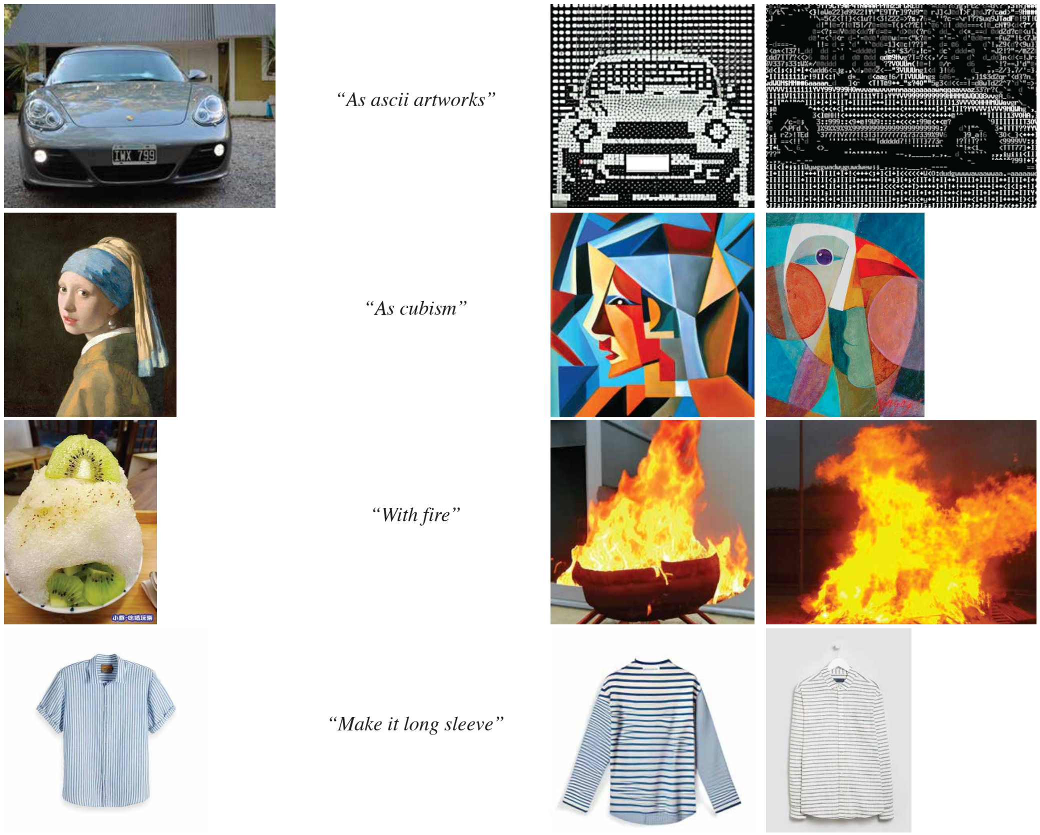
unCLIP (Ramesh et al., 2022) consists of a prior module that converts text embeddings into image embeddings, a decoder that converts image embeddings into low-resolution images, and super-resolution models that convert low-resolution images into high-resolution images. As the official unCLIP model is not publicly available, we employ the community version of unCLIP. Fortunately, since the community unCLIP model uses embeddings from CLIP-L/14, we can directly use this model to generate images from the image embeddings generated by our CompoDiff. To do this, we simply replace Prior with CompoDiff. The generated images are shown in Fig. D.3, D.4, D.5, and D.6. To clarify, the unCLIP model is trained for **text-to-image** generation, not to edit input images and our CompoDiff generates image embeddings rather than generating images. As shown, the results are very promising. It seems that incorporating unCLIP into the search service could potentially improve the user experience by generating images when the desired search results are not available.



(a) Zero-shot results

(b) Supervised results

Figure D.2: **Fashion IQ CIR results by adjusting w_T and w_I.** A red/blue cell denotes a higher/lower score.



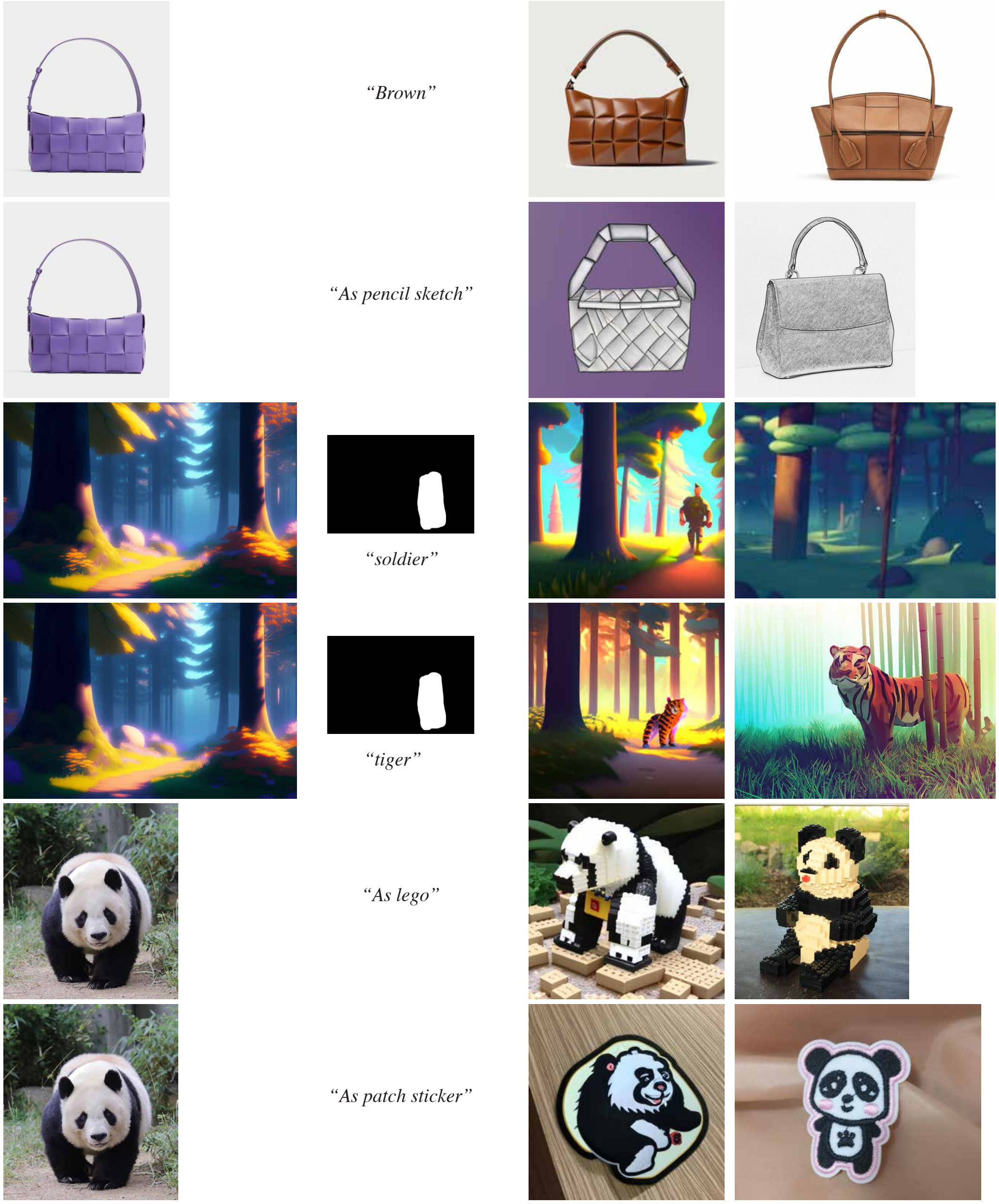
(a) Reference Image

(b) Conditions

(c) Generated by unCLIP

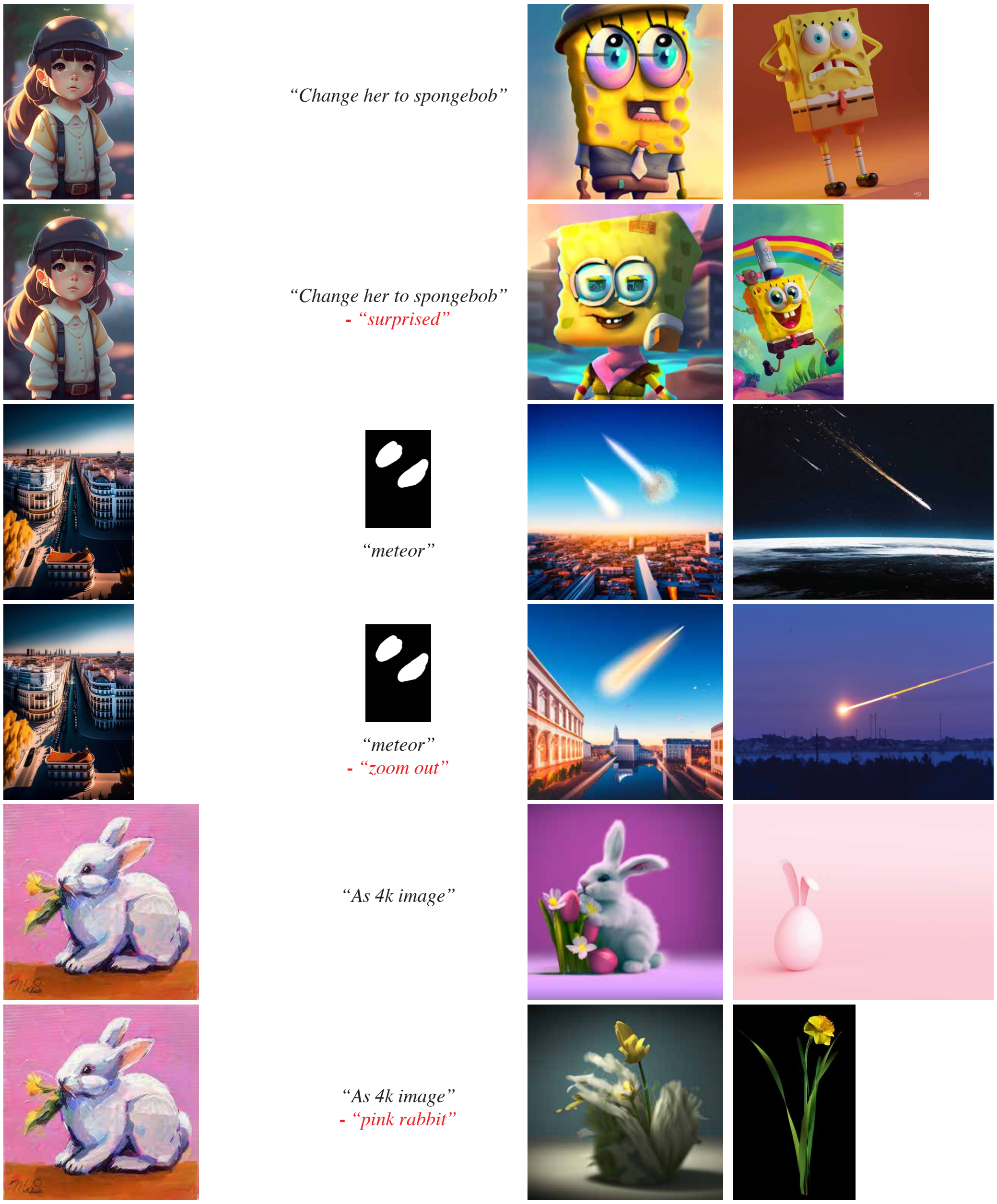
(d) Top-1 item from LAION

Figure D.3: **Generated vs. retrieved images by CompoDiff.** Using the transformed image feature by CompoDiff, Generated images using unCLIP and top-1 retrieved image from LAION.



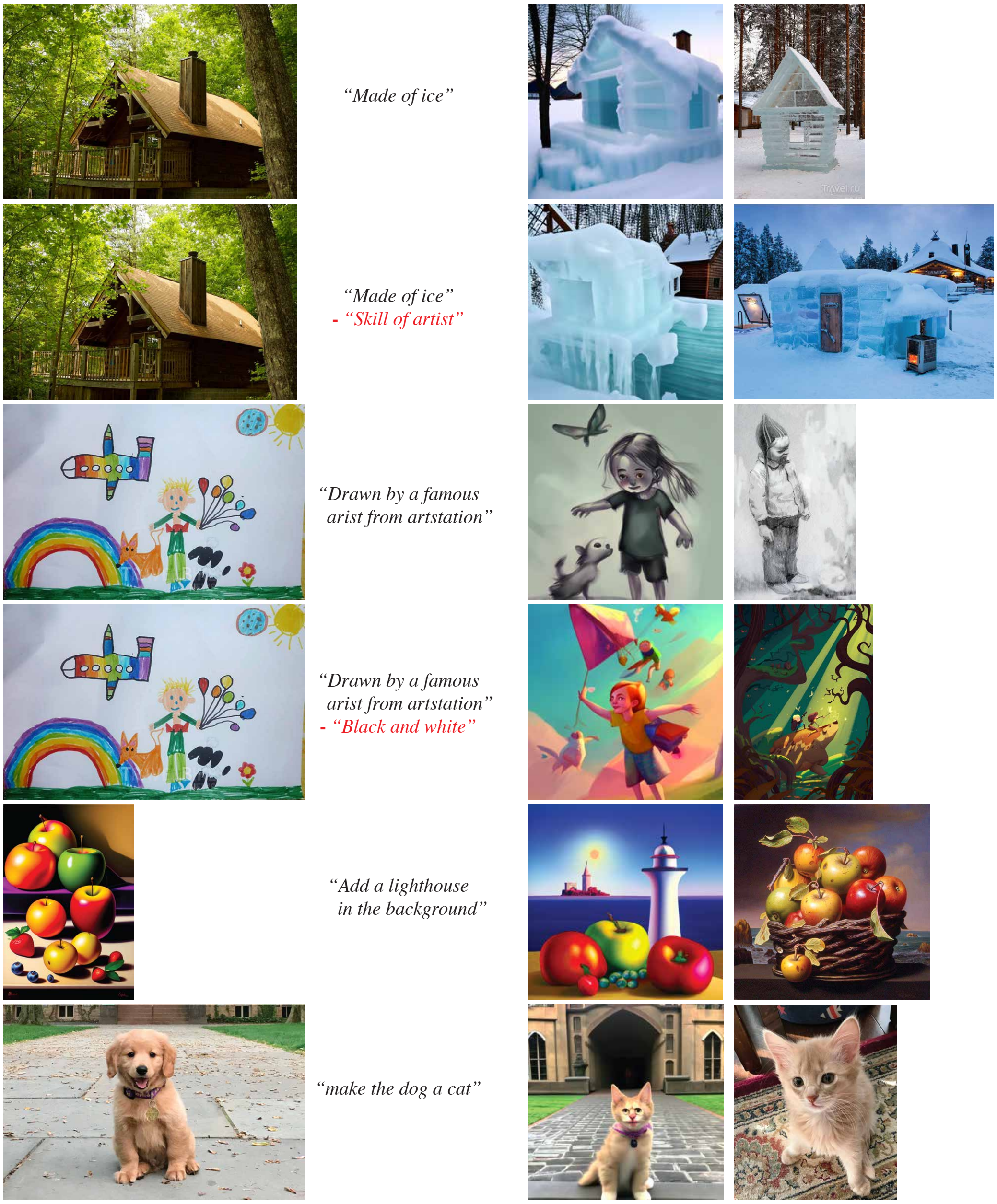
(a) Reference Image (b) Conditions (c) Generated by unCLIP (d) Top-1 item from LAION

Figure D.4: **Generated vs. retrieved images by CompoDiff (Continue).**



(a) Reference Image (b) Conditions (c) Generated by unCLIP (d) Top-1 item from LAION

Figure D.5: **Generated vs. retrieved images by CompoDiff (Continue).**


"Made of ice"
"Made of ice"
- "Skill of artist"
"Drawn by a famous arist from artstation"
"Drawn by a famous arist from artstation"
- "Black and white"
"Add a lighthouse in the background"
"make the dog a cat"

(a) Reference Image (b) Conditions (c) Generated by unCLIP (d) Top-1 item from LAION

Figure D.6: **Generated vs. retrieved images by CompoDiff (Continue).**