CoVR-2: Automatic Data Construction for Composed Video Retrieval

Lucas Ventura, Antoine Yang, Cordelia Schmid, Gül Varol

CoVR-2: Automatic Data Construction for Composed Video Retrieval

Lucas Ventura, Antoine Yang, Cordelia Schmid, *Fellow, IEEE,* Gül Varol

Abstract—Composed Image Retrieval (CoIR) has recently gained popularity as a task that considers *both* text and image queries together, to search for relevant images in a database. Most CoIR approaches require manually annotated datasets, comprising image-text-image triplets, where the text describes a modification from the query image to the target image. However, manual curation of CoIR *triplets* is expensive and prevents scalability. In this work, we instead propose a scalable automatic dataset creation methodology that generates triplets given video-caption *pairs*, while also expanding the scope of the task to include Composed *Video* Retrieval (CoVR). To this end, we mine paired videos with a similar caption from a large database, and leverage a large language model to generate the corresponding modification text. Applying this methodology to the extensive WebVid2M collection, we automatically construct our WebVid-CoVR dataset, resulting in 1.6 million triplets. Moreover, we introduce a new benchmark for CoVR with a manually annotated evaluation set, along with baseline results. We further validate that our methodology is equally applicable to image-caption pairs, by generating 3.3 million CoIR training triplets using the Conceptual Captions dataset. Our model builds on BLIP-2 pretraining, adapting it to composed video (or image) retrieval, and incorporates an additional caption retrieval loss to exploit extra supervision beyond the triplet, which is possible since captions are readily available for our training data by design. We provide extensive ablations to analyze the design choices on our new CoVR benchmark. Our experiments also demonstrate that training a CoVR model on our datasets effectively transfers to CoIR, leading to improved state-of-the-art performance in the zero-shot setup on the CIRR, FashionIQ, and CIRCO benchmarks. Our code, datasets, and models are publicly available at https://imagine.enpc.fr/~ventural/covr.

Index Terms—Composed Video Retrieval, Composed Image Retrieval.

1 Introduction

Consider the scenario where a traveller takes a picture of a landmark or scenic spot and wants to discover videos that capture the essence of that location, by specifying certain conditions via text. For example, the query image in Figure 1 (of a fountain in Barcelona), along with the text "during show" should bring the video showcasing the fountain show. Further refining the text query such as "during show at night", would allow the traveller to decide whether to wait for the show until the night time. In this work, our goal is composed video retrieval (CoVR), where the user performs such multi-modal search, by querying an image of a particular visual concept and a modification text, to find videos that exhibit the similar visual characteristics with the desired modification, in a dynamic context.

CoVR has many use cases, including but not limited to searching online videos for finding reviews of a specific product, how-to videos of a tool for specific usages, live events in specific locations, sports matches of specific players. Similar to composed image retrieval (CoIR), CoVR is also particularly useful when conveying a concept with a visual is easier and/or more accurate than only using words (e.g., unknown location/object, a specific camera view, a specific color).

Given the increased momentum in vision and language research in the recent years [1], [2], CoIR has emerged as a new task [3], and since then witnessed improvements of both models and benchmarks [4]–[9]. However, to the best of our knowledge, CoVR was not studied before. A key challenge in building CoVR models is the difficulty of gathering suitable training data of video-text-video triplets. We overcome this limitation by developing an automatic approach to generate triplets from existing video-caption collections. Specifically, we mine video pairs whose corresponding captions slightly differ in text space. We automatically describe this difference with a language model, which we train for a *modification-text generation* task. In particular, we use manually annotated triplets, each containing: (a) source caption, (b) target caption, (c) the modification text. We then finetune a large language model (LLM) [10] by inputting (a-b), and outputting (c). We assume the resulting modification to describe the difference between the corresponding videos, thus obtaining video-text-video triplets (see Figure 2 for an overview). When training our CoVR/CoIR models, we can flexibly select one or more frames from the videos, enabling multiple settings (i.e., retrieving images or videos).

Fig. 1. **Task:** Composed Video Retrieval (CoVR) seeks to retrieve *videos* from a database by searching with both a query image and a query text. The text typically specifies the desired modification to the query image. In this example, a traveller might wonder how the photographed place looks like during a fountain show, by describing several modifications, such as "during show at night, with fireworks".

- *Lucas Ventura and Gül Varol are with LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, France.*
  Email: lucas.ventura@enpc.fr
- *Antoine Yang is with Google DeepMind in London.*
- *Cordelia Schmid is with WILLOW project-team, ENS/Inria/CNRS, France.*

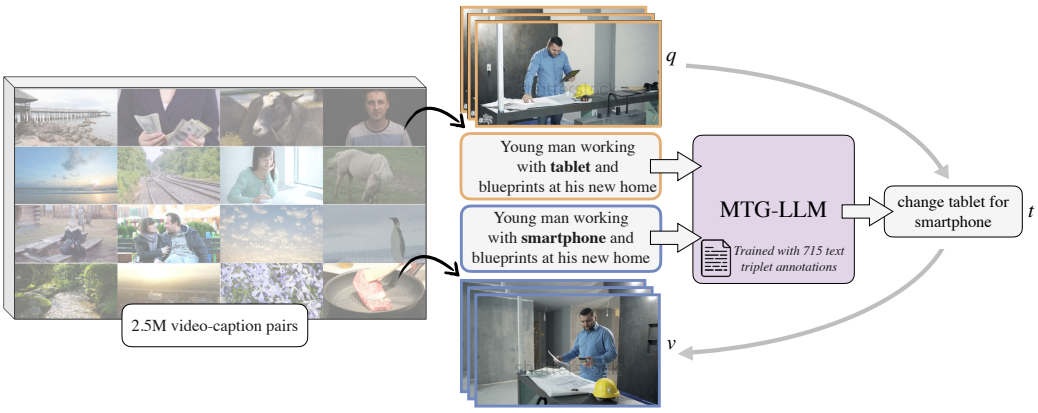

Fig. 2. **Method overview:** We automatically mine similar caption pairs from a large video-caption database from the Web, and use our modification text generation language model (MTG-LLM) to describe the difference between the two captions. MTG-LLM is trained on a dataset of 715 triplet text annotations [11]. The resulting triplet with the two corresponding videos (query q and target video v) and the modification text (t) is therefore obtained fully automatically, allowing a scalable CoVR training data generation.

We apply our triplet generation approach to two seed datasets: (1) WebVid2M [12] with 2.5M video-caption pairs, and (2) Conceptual Captions [13] with 3.3M image-caption pairs. We call the resulting training sets as WebVid-CoVR and CC-CoIR, which contain 1.6M CoVR and 3.3M CoIR triplets, respectively. By virtue of its automatic generation procedure, our training data is inherently noisy. To efficiently train on such large-scale and noisy data, we use a contrastive loss [14], adopting the HN-NCE variant from [15] to upsample the significance of hard negatives. Moreover, we integrate an additional contrastive loss, by also retrieving the textual embeddings of the target image, aiming to enhance the attention on the modification text detail. We design a CoVR model based on the cross-modal BLIP-2 [16] and use query scoring [17] to exploit information from multiple video frames. Training this model on WebVid-CoVR shows strong transferability to the CoIR task, in both zero-shot and finetuning settings, achieving state-of-the-art results on the standard CIRR [8], FashionIQ [9], and CIRCO [4] benchmarks in the zero-shot setup. We further improve the performance by jointly training on our WebVid-CoVR and CC-CoIR data together. Finally, to foster research in CoVR, we repeat our generation procedure on a distinct subset of the WebVid10M dataset [12] and manually select correctly generated samples to constitute WebVid-CoVR-Test, a test set of 2,435 CoVR triplets. We find that our model achieves promising results on WebVid-CoVR-Test compared to standard baselines.

To summarize, our contributions are: (i) We propose a scalable approach to automatically generate composed visual retrieval training data. With this methodology, we generate 1.6M WebVid-CoVR and 3.3M CC-CoIR triplets. (ii) We show that training composed retrieval models on our generated datasets transfers well to the CoIR benchmarks, and achieves state-of-the-art results on CIRR, FashionIQ, and CIRCO datasets in the zero-shot setup. (iii) We evaluate our model on WebVid-CoVR-Test, a new CoVR benchmark that we manually annotate. Our code, datasets, and models are publicly available at https://imagine.enpc.fr/~ventural/covr.

TABLE 1
Existing datasets: We compare our proposed CC-CoIR and WebVid-CoVR training datasets, along with its manually annotated test set WebVid-CoVR-Test with existing composed visual retrieval datasets. 📷 denotes image, 🎥 denotes video datasets. We contribute the largest training datasets for the natural domain. Note that, while SynthTriplets18M is larger, the transfer performance to real images is ineffective potentially due to a domain gap (see Table 4).

Dataset	Type	#Triplets	#Unique visuals	#Unique words	Avg. text length	Domain
CIRR [8]	📷	36,554	21,185	7,129	59.51	Natural
FashionIQ [9]	📷	30,132	7,988	4,425	27.13	Fashion
CIRCO-Test [4]	📷	800	-	870	50.94	Natural
LaSCo [7]	📷	389,305	121,479	13,488	30.70	Natural
SynthTriplets18M [6]	📷	18,000,000	-	-	-	Synthetic
CC-CoIR	📷	3,315,773	356,582	28,183	24.65	Natural
WebVid-CoVR	🎥	1,644,276	130,559	25,654	23.36	Natural
WebVid-CoVR-Test	🎥	2,556	4,886	1,910	21.97	Natural

2 Related Work

Composed image retrieval (CoIR). CoIR [3] has been an active area of research in recent years [3]–[6], [8], [9], [18]–[22]. Most methods designed for this problem use manually annotated image-text-image triplets for training [5], [8], [9], [19]. Recent works, such as Pic2Word [21], SEARLE [4], TFCIR [23], and LinCIR [24], explore zero-shot CoIR setups where no manually annotated CoIR triplet is used. More recently, Karthik et al. [25] propose a training-free method for CoIR. The approaches [4], [21], [23], [24] build on CLIP [2] and train a mapping network using image-only data for text inversion so that they can be flexibly composed with text descriptions. Our approach is similar in that it avoids collecting manual triplets; however, we instead perform supervised training on automatically generated image-text-video triplets given only video-text pairs. We also differ from above works by focusing on the composed video retrieval (CoVR) task, as opposed to only CoIR.

Datasets for composed image retrieval. CIRR [8] and Fashion-IQ [9] are the two most widely used CoIR benchmarks. Very recently, Baldrati et al. proposed a new test CoIR dataset CIRCO [4], which has gained popularity for having multiple ground truths and many distractors. All three datasets are manually annotated, hence small scale (about 30k triplets, see Table 1) due to the high cost implied

in collecting CoIR triplets. To scale up, two recent works proposed larger, automatically generated CoIR datasets: LaSCo [7] and SynthTriplets18M [6]. The LaSCo dataset [7] is generated using the visual question answering annotations and the pairing between images and counterfactual images in the VQAv2 dataset [26]. In detail, this dataset provides for each (image, question, answer) triplet a counterfactual triplet with the same question and different image and answer. In contrast, we do not rely on such expensive annotation schemes. SynthTriplets18M [6] uses the text-conditioned image editing framework InstructPix2Pix [11] to automatically generate CoIR data. Their edit text generation process is similar to ours, but our generation process differs in that we automatically mine similar videos from a dataset of video-text pairs to construct CoVR triplets instead of generating visual data. In experiments, we show the superiority of our triplet construction procedure as we achieve much higher CoIR results (e.g., 43.7% vs 26.7% zero-shot R@1 on CIRR while generating fewer data). Similar conclusions hold when pretraining on our automatic CoIR triplets (CC-CoIR). We additionally provide a controlled experiment by training our model on their synthetic images [6], [11] and demonstrate the advantages of using real data. Lastly, our WebVid-CoVR dataset is not limited to still images and considers videos, while standing out as larger than all previous composed retrieval datasets in the natural domain, as depicted in Table 1.

Vision-language pretraining. Many strong multi-modal models have been pretrained on large datasets of image-caption pairs [1], [2], [16], [27]–[33] or video-caption pairs [34]–[42]. In contrast, we generate CoVR training data from video-caption pairs instead of directly training on them. Our data generation approach is also related to other generation approaches used for other tasks, e.g., action recognition [43], visual question answering [44] and visual dialog [45]. However, unlike all these tasks, the CoVR task requires retrieving visual data.

Video retrieval. Text-to-video retrieval has received great attention over the last few years [39], [46]–[55]. We also make use of multiple video frames with query scoring similar to [17]. However, different from these methods, we focus on *composed* video retrieval, where the query consists of both text and visual data.

3 Automatic Triplet Generation and Training

The goal of our composed video retrieval (CoVR) task is, given an input image q and a modification text t, to retrieve a modified video v in a large database of videos[1]. Our goal is to avoid the manual annotation of (q, t, v) triplets for training. Hence we automatically generate such triplets from Web-scraped video-caption pairs, as explained in Section 3.1 and illustrated in Figure 2. The resulting WebVid-CoVR dataset, together with its manually curated evaluation set, is presented in Section 3.2. Finally, we present how we train a CoVR model using WebVid-CoVR in Section 3.3.

1. Note that q could also be a video query, but in our main experiments we focus on an image query, and provide more results in (Section D.1 of the Appendix) with video queries.

3.1 Generating composed video retrieval triplets

Given a large (Web-scraped) dataset of video-caption pairs, we wish to automatically generate video-text-video CoVR triplets (q, t, v) where the text t describes a modification to the visual query q. However, the dataset of video-caption pairs neither contains annotations of paired videos, nor modification text that describes their difference. Hence we propose a methodology to automatically mine paired videos and describe their difference, as described below. Note that for illustration, we take as an example the WebVid2M dataset [12] with 2.5M video-caption pairs, but this methodology could be applied to other large datasets of video-text (or image-text) pairs. To strengthen our conclusions, we employ the same methodology with the Conceptual Captions image-text dataset [13], which is briefly described along with experiments in Section 4.3. While the rest of this section focuses on CoVR, the data generation pipeline is similar for CoIR.

Mining paired videos by pairing captions. In order to obtain video pairs that exhibit visual similarity while differing in certain aspects, we leverage their associated captions. The core idea is that videos with similar captions are likely to have similar visual content. Specifically, we consider captions that differ by a single word, excluding punctuation marks. For instance, the caption *"Young woman smiling"* is paired with *"Old woman smiling"* and *"Young couple smiling"*. In the 2M distinct captions from WebVid2M, this process allows us to identify a vast pool of 1.2M distinct caption pairs with 177k distinct captions, resulting in 3.1M paired videos. In the following, we describe further steps to filter the data into a smaller set.

Filtering caption pairs. We wish to automatically generate the modification text between paired videos using their (paired) captions. However, caption pairs with the same meaning are likely to result in meaningless differences. On the contrary, caption pairs that differ too much are likely to result in large visual differences that cannot be easily described. To address these issues, we filter out caption pairs that are too similar and too dissimilar. Specifically, we exclude caption pairs with CLIP text embedding similarity ≥ 0.96 (e.g., *"Fit and happy young couple playing in the park"* and *"Fit and happy young couple play in the park"*) and caption pairs with CLIP text embedding similarity ≤ 0.6 (e.g., *"Zebra on a white background"* and *"Coins on a white background"*). We also exclude pairs where the captions differ by a digit (which mostly consist of a date in practice), a word not part of the English dictionary, or by a rare word. Rare words are detected based on the `zipf` frequency [**?**]. Finally, we remove templated captions such as *"abstract of"*, *"concept of"*, and *"flag of"* which are over-represented in WebVid2M. At the end of this filtering stage, we have 370k distinct caption pairs with 92k distinct captions, resulting in 1.2M paired videos that we will use to generate the modification text. Note that we can use these paired videos in both directions to generate triplets, as the source and target videos can be swapped.

Generating a modification text from paired captions. In order to generate a modification text between paired videos, we develop and apply a "modification text generation large language model" (MTG-LLM) to their corresponding paired

captions. We describe the MTG-LLM inference process below and then explain its training details. The MTG-LLM takes as input two paired captions and generates a modification text that describes the difference between the two captions (see Figure 2). In detail, the generation is auto-regressive, i.e., we recursively sample from the token likelihood distribution conditioned on the previously generated tokens until an end-of-sentence token is reached. Examples of the input-output, and details about the prompt format, which involves concatenating the two captions with a delimiter, can be found in Section C.4 of the Appendix. We use top-k sampling [56] for generating the tokens instead of maximum-likelihood methods such as beam search. Note that we only generate a single modification text per caption pair for computational efficiency, but the MTG-LLM could be used to generate multiple modification texts per caption pair which could serve as a data augmentation in future work.

We now describe the training details of the MTG-LLM. We start from a LLM pretrained with a next token prediction objective on a Web-scale text dataset, namely LLaMA [10]. We then finetune this LLM for the MTG task on a manually annotated text dataset. In particular, we repurpose the editing dataset from InstructPix2Pix [11], which provides a modification text and a target caption for 700 input captions. We augment this dataset with 15 annotations that cover additional cases. More details about the additional examples can be found in Section C.4 .

Filtering video pairs. We wish to avoid some modification texts being over-represented in the dataset as it could harm training. Hence, if there are more than 10 video pairs associated with the same pair of captions (therefore leading to the same modification text), we only select top 10 video pairs. As the CoVR task typically involves similar query-target video pairs, we choose pairs of videos with the highest visual similarity, as measured by the CLIP visual embedding similarity computed at the middle frame of the videos.

3.2 Our resulting WebVid-CoVR dataset

In the following, we describe the training and test partitions of our CoVR data. While our training set is automatically generated, our test set is manually verified.

WebVid-CoVR: a large-scale CoVR training dataset. By applying the previously described pipeline to the WebVid2M dataset [12], we generate WebVid-CoVR, a dataset containing 1.6M CoVR triplets, which is significantly larger than prior datasets (see Table 1). On average, a video lasts 16.8 seconds, a modification text contains 4.8 words, and one target video is associated with 12.7 triplets. We study the effect of the modification text length in Section D.5 of the Appendix. WebVid-CoVR is highly diverse with 131k distinct videos and 467k distinct modification texts. Examples of CoVR triplets from the WebVid-CoVR dataset are illustrated in Figure 3. These examples (along with additional ones included in Section E.3) demonstrate the diversity present in WebVid-CoVR, highlighting a wide range of content and variations in the modification texts. However, it is important to acknowledge that some noise naturally exists in the dataset, as shown in the bottom example of Figure 3, where the text does not describe the difference between the two videos due to both videos describing beautiful fields. We provide

Fig. 3. **Examples of generated CoVR triplets in WebVid-CoVR:** The middle frame of each video is shown with its corresponding caption, with the distinct word highlighted in bold. Additionally, the generated modification text is displayed on top of each pair of videos. The bottom example illustrates a noisy generated modification text, as 'beautiful' is subjective and both target and query videos can be considered as beautiful fields.

further analysis such as removal of inappropriate content, and dataset statistics of WebVid-CoVR in Section A of the Appendix.

WebVid-CoVR-Test: a new CoVR evaluation benchmark. Due to the noise in WebVid-CoVR, we manually annotate a small test set, dubbed WebVid-CoVR-Test, for evaluation. For this, we first repeat the data generation procedure described in Section 3.1, but on a different corpus of video-caption pairs. Specifically, we consider video-caption pairs from the WebVid10M corpus [12] that are not included in the WebVid2M dataset, resulting in a pool of 8 million video-caption pairs. This ensures that other models using WebVid2M for pretraining have not been exposed to any of the test examples. In the video pairs filtering stage, for each pair of captions, we here only keep one pair of videos (the one with the highest visual similarity). This results in 163k candidate triplets that could be used for testing purposes. We randomly sample 7k triplets that we use for validation and randomly sample 3.2k other triplets that we manually annotate as described below.

We augment the 3.2k triplets by generating two additional modification texts with the MTG-LLM. The annotator reads the three generated modification texts, looks at three frames from the query and target videos, and either keeps the best modification text if at least one is valid or discards the sample. Through this meticulous annotation process, we ensure that the test set comprises high-quality and meaningful CoVR triplets. This results in a test set of 2.5k triplets, i.e., about 22% of the examples are considered as noisy and are discarded.

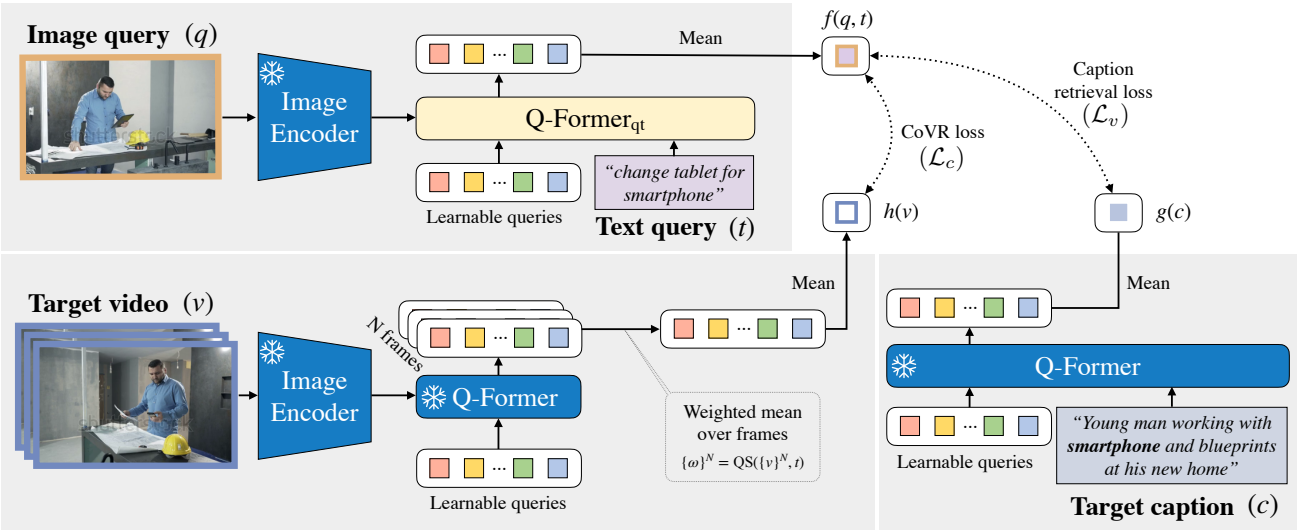

Fig. 4. **Model architecture of CoVR-BLIP-2:** The BLIP-2 [16] image encoder extracts visual features from the image query q. These visual features are combined with the text query t (modification text) through the BLIP-2 image-grounded text encoder to obtain a multi-modal query embedding $f(q,t)$. To encode videos, N frames are individually encoded with the BLIP-2 image encoder and its Q-Former, and aggregated via a weighted mean into a single video embedding $h(v)$. The goal of CoVR (video retrieval) is to maximize similarity between the multi-modal query $f(q,t)$ and the target video $h(v)$. During training, an additional caption retrieval loss $\mathcal{L}_\rfloor$ is defined between $f(q,t)$ and the target caption embedding $g(c)$. Note that for simplicity, we visualize one Q-Former block, but in practice there are 12 blocks as in [16]. To reduce the 32 tokens output by the Q-Former, we simply average them before computing cosine similarities. While each Q-Former is initialized from the BLIP-2 pretraining, they are finetuned on our CoVR/CoIR data. When training for CoIR, the target becomes a single image, removing the need for weighted mean. See Section 3.3 for more details.

3.3 Training on WebVid-CoVR

Here, we describe our CoVR model architecture and how we train it on our WebVid-CoVR dataset.

CoVR-BLIP-2 model architecture. Our model, illustrated in Figure 4, builds on a pretrained image-text model, BLIP-2 [16], an enhancement over the original BLIP [1]. The main architectural difference with the CoVR-BLIP approach in our preliminary work [57] is the use of BLIP-2 [16] instead of BLIP [1]. BLIP-2 introduces a lightweight Querying Transformer (Q-Former) with learnable queries for more efficient visual feature extraction. Like its predecessor, BLIP-2 is pretrained on a large dataset of image-caption pairs with three vision-language objectives: image-text contrastive learning, image-text matching, and image-conditioned language modeling. However, BLIP-2 is not trained for composed visual retrieval with both visual and text inputs. Therefore we adapt BLIP-2 to the CoIR/CoVR task as follows.

We use the BLIP-2 image encoder to encode the image query q (which corresponds to the middle frame of the video in case of WebVid-CoVR). The resulting visual features and the modification text (t) are then forwarded to the BLIP-2 image-grounded text encoder together, which outputs a multi-modal embedding $f(q,t) \in \mathbb{R}^d$ where d is the embedding dimension. Note that we compute mean across the 32 output tokens corresponding to the learnable query inputs of the Q-Former, after passing them through linear projection layer initialized from BLIP-2 text projection layer.

To retrieve a target video v_k from a database of videos V, we compute embedding vectors for all gallery videos as follows. We uniformly sample N frames from the video and compute a weighted mean of the BLIP-2 image embeddings to obtain the video embedding vector $h(v_k) \in \mathbb{R}^d$. The weights $\{\omega\}^N$ are obtained by computing the similarity between the corresponding frame and the modification text using the pretrained BLIP-2 image and text encoders, respectively (introduced as 'query scoring' in [17] in the context of text-to-video retrieval). Using pretrained and frozen BLIP-2 embeddings allows us to precompute and store all these weights, which we refer to as $\{\omega\}^N = QS(\{v\}^N, t)$ for query-scoring between each of the N frames of the video v and the text t.

At test time, given a multi-modal embedding $f(q,t)$, the retrieved video is the one that maximizes the embedding similarity, i.e., $\arg\max_{v_k \in V}(h(v_k) \cdot f(q,t)^T)$.

Training. In order to train on WebVid-CoVR, we use a contrastive learning approach [14], [15], as it has been shown to be effective to learn strong multi-modal representations from large-scale noisy data [2]. We make the following design choices. First, we create a training batch by sampling distinct target videos; and for each target video, we randomly sample an associated image-text query pair. Iterating over videos ensures that the same target video appears only once in a batch and maximizes the number of different target videos that can be used as negatives in contrastive learning. We show the benefit of this approach in Section 4.6 (Table 10). Second, we employ HN-NCE [15] which increases the weight of most similar samples and uses as negatives all target videos $v_{j \in \mathcal{B}}$ in the batch $\mathcal{B}$. Formally, given a training batch $\mathcal{B}$ of triplets (q_i, t_i, v_i), we minimize the following loss:

$$\mathcal{L}_v(\mathcal{B}) = -\sum_{i \in \mathcal{B}} \log \left(\frac{e^{S_{i,i}/\tau}}{\alpha \cdot e^{S_{i,i}/\tau} + \sum_{j \neq i} e^{S_{i,j}/\tau} w_{i,j}} \right) - \sum_{i \in \mathcal{B}} \log \left(\frac{e^{S_{i,i}/\tau}}{\alpha \cdot e^{S_{i,i}/\tau} + \sum_{j \neq i} e^{S_{j,i}/\tau} w_{j,i}} \right) \quad (1)$$

where α is set to 1, temperature τ is set to 0.07, $S_{i,j}$ is

the cosine similarity between the multi-modal embedding $f(q_i, t_i)$ and the target video embedding $h(v_j)$, and $w_{i,j}$ is set as in [15] with $\beta = 0.5$.

Composed caption retrieval as an additional loss term. In addition to using the video as a target, our approach also leverages the supervision from the caption corresponding to each video. This is possible in our training data because we in fact have 5 elements (video1, caption1, video2, caption2, modification text) for each data sample. This new loss term involves aligning the multi-modal query embedding $f(q, t)$ not only with the video embedding $h(v)$ but also with its descriptive caption c. To this end, we encode the caption into an embedding $g(c)$, using frozen BLIP-2 text encoding and define an additional contrastive loss term ($\mathcal{L}_c$). Therefore, our final loss can be expressed as:

$$\mathcal{L}(\mathcal{B}) = \lambda_v \cdot \mathcal{L}_v(\mathcal{B}) + \lambda_c \cdot \mathcal{L}_c(\mathcal{B}), \tag{2}$$

where λ_v and λ_c are both set to 0.5. Here, $\mathcal{L}_v$ refers to the initial loss in Eq.(1) utilizing cosine similarity with the target video embedding $h(v_j)$, and $\mathcal{L}_c$ employs the same loss structure but applies it to the cosine similarity with the target video caption embedding $g(c_j)$.

4 Experiments

We first describe the experimental protocol including the datasets, evaluation metrics, and implementation details (Section 4.1). We then present the results of CoVR on our new video benchmark (Section 4.2). Additionally, we introduce CC-CoIR, a new large-scale CoIR training dataset derived with a similar methodology, from the Conceptual Captions (CC) dataset (Section 4.3). We show transfer results of CoIR on standard image benchmarks, together with an extensive state-of-the-art comparison (Section 4.4). We further provide a comparison by training on other automatic triplet datasets (Section 4.5). Finally, we provide ablations on our key components such as the caption retrieval loss and data scale. (Section 4.6) and we illustrate qualitative examples in Section 4.7).

4.1 Experimental setup

Datasets. WebVid-CoVR is our proposed training CoVR dataset, and **WebVid-CoVR-Test** is our new CoVR benchmark, both presented in Section 3.2. For pre-training on CoIR datasets, we use **WV-CC-CoVIR**, a combination of both WebVid-CoVR and **CC-CoIR** which is a new large-scale CoIR training dataset derived from the Conceptual Captions dataset, that we generate using the same methodology as WebVid-CoVR. See Section 4.3 for more details.

CIRR [8] is a manually annotated CoIR dataset that contains open-domain natural images from NLVR2 [58], comprising 36.5k queries annotated on 19k images. CIRR includes two evaluation protocols: a standard one with the entire validation set as the search gallery, and a fine-grained *subset*, where the search space is a subgroup of six images similar to the query image (based on pretrained ResNet15 feature distance). The dataset is divided into training, validation, and testing splits with 28225/16742, 4181/2265 and 4148/2178 queries/images, respectively.

FashionIQ [9] is another CoIR dataset that contains images of fashion products, divided into three categories of Shirts, Dresses, and Tops/Tees. The query and target images were automatically paired based on title similarities (crawled from the web), and modification texts were then manually annotated. This dataset consists of 30k queries annotated on 40.5k different images. It is divided into training and validation splits with 18000/45429 and 6016/15415 queries/images, respectively.

CIRCO [4] is an open-domain dataset for CoIR, specifically designed for zero-shot CoIR tasks, as there is no specific training split provided for this dataset. It is unique in its inclusion of multiple ground truths for each query, with an average of 4.53 ground-truth images per query. This feature allows for a more reliable and robust evaluation using the mean Average Precision (mAP) metric. CIRCO is divided into a validation split (220 queries) and a test split (800 queries). The evaluation protocol uses all 120,000 images from the COCO dataset as its gallery set.

Evaluation metrics. Following standard evaluation protocols [8], we report the video retrieval recall at rank 1, 5, 10, and 50. Recall at rank k (R@k) quantifies the number of times the correct video is among the top k results. MeanR denotes the average of R@1, R@5, R@10, and R@50. Higher recall means better performance.

Implementation details and environmental costs. For our MTG-LLM, we use LLaMA 7B model [10] that we finetune for one epoch with an initial learning rate of $3e{-5}$. For our CoVR model, we use the BLIP-2 with ViT-G/14 [59] at 364 pixels finetuned for text-image retrieval on COCO and freeze the ViT for computational efficiency. We train our CoVR model on WebVid-CoVR for 5 epochs with a batch size of 2048 and an initial learning rate of $1e{-5}$. To finetune on CIRR/FashionIQ, we train for 6 epochs with a batch size of 2048/1024 and an initial learning rate of $1e{-4}$. We set hyperparameters based on the validation curve of WebVid-CoVR. Experiments are conducted on 4 NVIDIA A100-SXM4-80GB GPUs. The experiments conducted in this study incurred an environmental cost of approximately 49kg of CO_2 emissions. More details are included in Section C of the Appendix.

4.2 Composed video retrieval results

We provide a number of baselines for our new benchmark on WebVid-CoVR-Test. Table 2 summarizes these CoVR results. We first report the random chance performance in the first row. The rest of the table is split into two. The top block uses existing pretrained text and image encoders from BLIP [1], BLIP-2 [16] or CLIP [2] backbones without any finetuning. Models in the bottom block are finetuned on WebVid-CoVR. We report results with the composed query, as well as with the individual modalities. For combining modalities, we experiment with the simple average fusion baseline (Avg) when using frozen embeddings, and fusion with a randomly-initialized MLP or BLIP-pretrained cross-attention (CA) layers when finetuning. Note that the MLP fusion baseline is similar to Combiner [5] that concatenates the image and text embeddings from CLIP (or BLIP in CASE [7]), and is referred to as late fusion by CASE. For finetuning individual modalities, we train and test either with text-only query using the modification text, or with the visual-only image query. Finally, we experiment with using

TABLE 2
Benchmarking on the WebVid-CoVR-Test set: We observe that using both the visual and text input modalities performs better than individual modalities alone, both with/without finetuning on WebVid-CoVR (shown at the top/bottom of the table, respectively). When using pretraining models without finetuning, we apply average fusion (Avg) for the embeddings. BLIP performs slightly better than CLIP on this benchmark. Finetuning on WebVid-CoVR brings significant benefits. In this case, fusing with the pretrained cross-attention (CA) from BLIP is more effective than training a randomly-initialized MLP fusion as done in [5]. Moreover, using multiple frames to embed the target video brings further improvements over using the middle frame. The first row represents the random performance.

	Input modalities	Fusion	Backbone	WebVid-CoVR-Test R@1	R@5	R@10	R@50
Random	-	-	-	0.08	0.23	0.35	1.76
Not finetuned on WebVid-CoVR	Text	-	BLIP-2	18.74	38.11	47.97	67.21
	Visual	-	BLIP-2	33.10	59.55	69.80	88.85
	Visual + Text	Avg	CLIP	44.37	69.13	77.62	93.00
	Visual + Text	Avg	BLIP	45.46	70.46	79.54	93.27
	Visual + Text	Avg	BLIP-2	**45.66**	**71.71**	**81.30**	**94.80**
Finetuned on WebVid-CoVR	Text	-	BLIP-2	23.32	46.21	56.22	78.36
	Visual	-	BLIP-2	36.03	64.24	74.77	92.64
	Visual + Text	MLP	CLIP	50.86	77.46	85.32	96.75
	Visual + Text	MLP	BLIP	50.59	74.65	83.57	95.46
	Visual + Text	MLP	BLIP-2	51.88	79.38	86.46	97.42
	Visual + Text	CA	BLIP	55.95	81.22	89.05	98.08
	Visual + Text	CA	BLIP-2	**59.82**	**83.84**	**91.28**	**98.24**

the weighted average of target video frame embeddings as explained in Section 3.3 (with the exception that visual-only experiments use equal weights due to not having access to the modification text for computing the scores).

We make several conclusions. (i) Combining both visual and text modalities yields better performance than the models with individual modalities. This result highlights that our new CoVR benchmark requires paying attention to both modalities. (ii) Visual-only outperforms text-only suggesting that the video pairs automatically mined through their caption similarity indeed exhibits visual similarity, and that the image captures the target video better than the modification text. (iii) Finetuning on WebVid-CoVR obtains substantial improvements over using pretrained and frozen embeddings. (iv) When finetuning, fusion with BLIP-2 cross-attention (CA) performs better than the MLP fusion. (v) Results with the BLIP-2 backbone are higher than those with CLIP or BLIP. We analyze the effect of the number of frames in Section D.6.

4.3 Effect of training with CC-CoIR

We apply the same automatic triplet generation procedure to the Conceptual Captions dataset [13] (CC3M) resulting in the formation of CC-CoIR. This new dataset comprises 3.3M CoIR triplets, utilizing the diverse and contextually rich content of CC3M [13]. In contrast to WebVid-CoVR, which is video-based, CC-CoIR exclusively incorporates images, adding a different modality to our training material. The average length of modification texts in this dataset is 24.65, featuring 130k unique images and 28k distinct modification texts.

Table 3 presents the results when training with our newly introduced CC-CoIR dataset. When used independently for pretraining (zero-shot) we observe the following: (a) CC-CoIR outperforms WebVid-CoVR on the CIRR dataset, (b) CC-CoIR has similar performance than WebVid-CoVR in FashionIQ, (c) CC-CoIR has lower performance for the CIRCO benchamark,

TABLE 3
Training with CC-CoIR: We compare the performance metrics when pretraining with various combinations of WebVid-CoVR and CC-CoIR datasets. We observe that in the zero-shot setting, combining both datasets yields the best overall results.

WebVid-CoVR	CC-CoIR	WebVid-CoVR-T. R@1	CIRR R@1	FashionIQ R@10	CIRCO mAP@5
✗	✓	53.83	43.35	36.49	23.50
✓	✗	**59.82**	41.42	36.81	**28.88**
✓	✓	57.71	**43.74**	**38.15**	28.29

and (d) CC-CoIR performance has good zero-shot results on our video retrieval test set WebVid-CoVR-Test. Note that WebVid-CoVR-Test constitutes a zero-shot setting only when trained using the CC-CoIR pretraining, whereas pretraining on WebVid-CoVR means the model has seen samples from the same data distribution during training. By combining both WebVid-CoVR and CC-CoIR into a unified pretraining dataset WV-CC-CoVIR, we observe a slight improvement on the image-based datasets while experiencing a minor drop on WebVid-CoVR-Test compared to using only WebVid-CoVR. We hypothesize that this minor drop is due to a domain gap with CC-CoIR, which is included in WV-CC-CoVIR. We therefore opt to use the jointly pretrained model going forward, as it leverages the strengths of both data sources to perform well across different datasets (i.e., best average recall across all datasets).

4.4 State-of-the-art comparison on CoIR benchmarks

While our focus is video retrieval, we also experiment with transferring our CoVR models to image retrieval tasks on standard CoIR benchmarks. We define zero-shot CoIR as not using any manually annotated CoIR triplet for training. To perform zero-shot CoIR, we directly apply our model, which has been trained on our automatically generated WV-CC-CoVIR dataset, to CoIR tasks. In addition to zero-shot

TABLE 4
State-of-the-art comparison on the CIRR and FashionIQ test sets: Our model benefits from training on our datasets in the zero-shot setting, and in the finetuning setting where it performs competitively. † denotes results reported by [8]. For methods that pretrain specifically for composed retrieval, we indicate their pretraining data. CC3M denotes Conceptual Captions 3M [13], ST18M denotes [6], and COCO denotes [60].

Mode	Method	Year	Pretraining Data	CIRR Recall@K K=1	K=5	K=10	K=50	CIRR R_{subset}@K K=1	K=2	K=3	FashionIQ Dress R@10	R@50	Shirt R@10	R@50	Toptee R@10	R@50	Average R@10	R@50
Train	TIRG [3]†	2019	-	14.61	48.37	64.08	90.03	22.67	44.97	65.14	14.87	34.66	18.26	37.89	19.08	39.62	17.40	37.39
	TIRG+LastConv [3] †	2019	-	11.04	35.68	51.27	83.29	23.82	45.65	64.55	-	-	-	-	-	-	-	-
	CurlingNet [61]‡	2019	-	-	-	-	-	-	-	-	26.15	53.24	21.45	44.56	30.12	55.23	25.90	51.01
	JVSM [62]	2020	-	-	-	-	-	-	-	-	10.70	25.90	12.00	27.10	13.00	26.90	11.90	26.60
	TRACE w/BER [63]	2020	-	-	-	-	-	-	-	-	22.70	44.91	20.80	40.80	24.22	49.80	22.57	46.19
	VAL w/GloVe [64]	2020	-	-	-	-	-	-	-	-	22.53	44.00	22.38	44.15	27.53	51.68	24.15	46.61
	MAAF [65]†	2020	-	10.31	33.03	48.30	80.06	21.05	41.81	61.60	23.80	48.60	21.30	44.20	27.90	53.60	24.30	48.80
	MAAF-BERT [65]†	2020	-	10.12	33.10	48.01	80.57	22.04	42.41	62.14	-	-	-	-	-	-	-	-
	MAAF-IT [65]†	2020	-	9.90	32.86	48.83	80.27	21.17	42.04	60.91	-	-	-	-	-	-	-	-
	MAAF-RP [65]†	2020	-	10.22	33.32	48.68	81.84	21.41	42.17	61.60	-	-	-	-	-	-	-	-
	RTIC-GCN [66]‡	2021	-	-	-	-	-	-	-	-	29.15	54.04	23.79	47.25	31.61	57.98	28.18	53.09
	CoSMo [67]	2021	-	-	-	-	-	-	-	-	25.64	50.30	24.90	49.18	29.21	57.46	26.58	52.31
	DCNet [20]	2021	-	-	-	-	-	-	-	-	28.95	56.07	23.95	47.30	30.44	58.29	27.78	53.89
	CIRPLANT [8]†	2021	-	19.55	52.55	68.39	92.38	39.20	63.03	79.49	17.45	40.41	17.53	38.81	61.64	45.38	18.87	41.53
	SAC w/BERT [68]	2022	-	-	-	-	-	-	-	-	26.52	51.01	28.02	51.86	32.70	61.23	29.08	54.70
	FashionVLP [69]	2022	-	-	-	-	-	-	-	-	32.42	60.29	31.89	58.44	38.51	68.79	34.27	62.51
	ARTEMIS [19]	2022	-	16.96	46.10	61.31	87.73	39.99	62.20	75.67	27.16	52.40	21.78	43.64	29.20	53.83	26.05	50.29
	LF-CLIP [5] †	2022	-	33.59	65.35	77.35	95.21	62.39	81.81	92.02	31.63	56.67	36.36	58.00	38.19	62.42	35.39	59.03
	LF-BLIP [5], [7] †	2022	-	20.89	48.07	61.16	83.71	50.22	73.16	86.82	25.31	44.05	25.39	43.57	26.54	44.48	25.75	43.98
	Combiner [5]	2022	-	33.59	65.35	77.35	95.21	62.39	81.81	92.02	31.63	56.67	36.36	58.00	38.19	62.42	35.39	59.03
	CompoDiff [6]	2023	ST18M+LAION2B	32.39	57.61	77.25	94.61	67.88	85.29	94.07	38.39	51.03	41.68	56.02	45.70	57.32	39.81	51.90
	CASE [7]	2024	LaSCo [7]	48.68	79.98	88.51	97.49	76.39	90.12	95.86	**47.44**	69.36	48.48	70.23	50.18	72.24	48.79	70.68
	CASE [7]	2024	LaSCo+COCO	49.35	80.02	88.75	97.47	76.48	90.37	95.71	-	-	-	-	-	-	-	-
	CoVR-BLIP [57]	2024	WebVid-CoVR	49.69	78.60	86.77	94.31	75.01	88.12	93.16	44.55	69.03	48.43	67.42	**52.60**	**74.31**	48.53	70.25
	CoVR-BLIP-2	2024	-	**50.87**	80.80	88.84	98.00	76.70	90.31	95.45	45.25	68.86	49.95	69.95	51.37	72.56	48.86	70.46
	CoVR-BLIP-2	2024	CC-CoIR	50.63	81.04	**89.35**	**98.15**	76.53	**90.43**	**96.00**	46.49	69.31	**51.67**	70.49	51.53	73.07	49.90	70.96
	CoVR-BLIP-2	2024	WV-CC-CoVIR	50.43	**81.08**	88.89	98.05	**76.75**	90.34	95.78	46.53	**69.60**	51.23	**70.64**	52.14	73.27	**49.96**	**71.17**
Zero Shot	Random†		-	0.04	0.22	0.44	2.18	16.67	33.33	50.00	0.26	1.31	0.16	0.79	0.19	0.95	0.06	0.32
	CompoDiff [6]	2023	ST18M+LAION2B	26.71	55.14	74.52	92.01	64.54	82.39	91.81	**37.78**	49.10	41.31	55.17	**44.26**	56.41	**39.02**	51.71
	Pic2Word [21]	2023	CC3M	23.90	51.70	65.30	87.80	-	-	-	20.00	40.20	26.20	43.60	27.90	47.40	24.70	43.70
	SEARLE-XL-OTI [4]	2023	CC3M	24.87	52.31	66.29	88.58	53.80	74.31	86.94	21.57	44.47	30.37	47.49	30.90	51.76	27.61	47.90
	SEARLE-XL [4]	2023	CC3M	24.24	52.48	66.29	88.84	53.76	75.01	88.19	26.89	45.58	20.48	43.13	29.32	49.97	25.56	46.23
	GRB [23]	2023	-	24.19	52.07	65.48	85.28	60.17	79.13	89.34	24.14	45.56	34.54	55.15	33.55	53.60	30.74	51.44
	GRB+LCR [23]	2023	-	30.92	56.99	68.58	85.28	66.67	78.68	82.60	-	-	-	-	-	-	-	-
	TFCIR [23]	2023	-	32.82	61.13	71.76	85.28	66.63	78.58	82.68	24.84	45.56	35.38	55.15	33.10	53.60	31.11	51.44
	CASE [7]	2024	LaSCo	30.89	60.75	73.88	92.84	60.17	80.17	90.41	-	-	-	-	-	-	-	-
	CASE [7]	2024	LaSCo+COCO	35.40	65.78	78.53	94.63	64.29	82.66	91.61	-	-	-	-	-	-	-	-
	CIReVL [25]	2024	-	34.65	64.29	75.06	91.66	67.95	84.87	93.21	27.07	49.53	33.71	51.42	35.80	56.14	32.19	52.36
	LinCIR (ViT-L) [24]	2024	-	25.04	53.25	66.68	-	57.11	77.37	88.89	20.92	42.44	29.10	46.81	28.81	50.18	26.28	46.49
	LinCIR (ViT-G) [24]	2024	-	35.25	64.72	76.05	-	63.35	82.22	91.98	**38.08**	**60.88**	**46.76**	**65.11**	**50.48**	**71.09**	**45.11**	**65.69**
	CoVR-BLIP [57]	2024	WebVid-CoVR	38.48	66.70	77.25	91.47	69.28	83.76	91.11	21.95	39.05	30.37	46.12	30.78	48.73	27.70	44.63
	CoVR-BLIP-2	2024	CC-CoIR	43.35	**73.78**	83.66	96.07	**75.25**	**88.89**	**95.23**	33.17	55.13	38.81	58.24	37.48	58.49	36.49	57.29
	CoVR-BLIP-2	2024	WV-CC-CoVIR	**43.74**	73.61	**83.95**	**96.10**	72.84	87.52	94.39	34.26	56.22	41.22	59.32	38.96	59.77	38.15	58.44

TABLE 5
State-of-the-art comparison on the CIRCO test set: Our model benefits from training on WV-CC-CoVIR in the zero-shot setting. For methods that pretrain specifically for composed retrieval, we indicate their pretraining data. CC3M denotes Conceptual Captions 3M [13] and ST18M denotes [6].

Method	Year	Pretraining Data	mAP@K K=5	K=10	K=25	K=50
Random		-	0.00	0.00	0.00	0.00
SEARLE-XL [4]	2023	CC3M	11.68	12.73	14.33	15.12
CompoDiff (ViT-G) [6]	2023	ST18M	15.33	17.71	19.45	21.01
GRB+LCR [23]	2023	-	25.38	26.93	29.82	30.74
TFCIR [23]	2023	-	26.52	28.25	31.23	31.99
CIReVL [25]	2024	-	26.77	27.59	29.96	31.03
LinCIR (ViT-G) [24]	2024	-	19.71	21.01	23.13	24.18
CoVR-BLIP [57]	2024	WebVid-CoVR	21.43	22.33	24.47	25.48
CoVR-BLIP-2	2024	CC-CoIR	23.50	24.66	27.05	28.04
CoVR-BLIP-2	2024	WV-CC-CoVIR	**28.29**	**29.55**	**32.18**	**33.26**

TABLE 6
Adding the composed caption retrieval loss: We compare models with and without using our additional caption retrieval loss, on both CoVR-BLIP [57] and CoVR-BLIP-2. We observe a significant improvement on CoIR datasets when training on the WebVid-CoVR with this additional loss.

Method	Caption ret. loss	WebVid-CoVR-T. R@1	CIRR R@1	FashionIQ R@10	CIRCO mAP@5
CoVR-BLIP	✗	56.81	37.18	23.20	22.86
CoVR-BLIP	✓	55.95	40.46	33.90	25.71
CoVR-BLIP-2	✗	**60.09**	34.43	27.78	25.06
CoVR-BLIP-2	✓	59.82	**41.42**	**36.81**	**27.84**

CoIR, for the datasets that have training split (CIRR and FashionIQ), we investigate how our model performs when finetuned on the training set of the downstream benchmark.

We report results in Table 4 on CIRR and Fashion-IQ datasets, and in Table 5 for the newer CIRCO dataset. In Table 4, we report both finetuning (top block), and zero-shot (bottom block) settings. In Table 5, we only report the zero-shot setting since CIRCO does not have a training set to finetune on. These results show that our model highly benefits from training on WV-CC-CoVIR, especially in the zero-shot setting. Remarkably, our model attains state-of-the-art performance on CIRR and CIRCO, exhibiting a 5% improvement in CIRR (43.74 vs 38.48 R@1) and a 2% in CIRCO (28.29 vs 26.77 mAP@5) compared to the nearest competitor's results. As for the FashionIQ dataset, we outperform all other methods but CompoDiff [6] and

TABLE 7
Synthetic vs real training images: We compare the CoIR performance of our proposed method (CoVR-BLIP-2) and dataset (WV-CC-CoVIR) against the CompoDiff [6] method and their proposed dataset (SynthTriplets). The results demonstrate that our training data achieves better performance with 1M triplets compared to 1M or even 18M triplets of the SynthTriplets dataset, containing synthetically generated target images. IP2P denotes InstructPix2Pix, 1M public synthetic triplets by [11].

Model	Pretraining Data	Data Size	WebVid-CoVR-T R@1	CIRR Avg(R@1, R_s@1)	FashionIQ Avg(R@10, R@50)
CompoDiff [6]	IP2P [11]	1M	-	27.42	27.24
	SynthTriplets [6]	1M	-	28.32	31.91
	SynthTriplets [6]	18M	-	37.83	42.33
CoVR-BLIP-2	IP2P [11]	1M	16.55	34.72	15.42
	SynthTriplets [6]	1M	45.42	28.44	33.23
	WV-CC-CoVIR	1M	**55.87**	**58.79**	**48.14**

LinCIR [24] with the ViT-G backbone. With CompoDiff, the results are mixed depending on the metric (38.15 vs 39.02 R@1 and 58.44 vs 51.71 R@10). For LinCIR, if we compare with the ViT-L backbone, more similar to ours, our methods obtains better results (38.15 vs 26.28). However, CompoDiff/LinCIR perform poorly on the other two datasets CIRR (43.74 vs 26.71/35.25) and CIRCO (28.29 vs 15.33/19.71). Our CoVR-BLIP-2 remains therefore the overall best zero-shot model when evaluating across three datasets. In addition to this strong zero-shot performance, our model reaches state-of-the-art performance when finetuned on both CIRR and FashionIQ benchmarks (top block of Table 4). For each setting, we also provide results with pretraining only on the CC-CoIR subset, as opposed to the full WV-CC-CoVIR, and observe similar performance on CIRR and FashionIQ, but lower on the challenging CIRCO dataset.

4.5 Comparison with synthetic training images

To further compare against other approaches that propose automatic triplets for training (i.e., by generating synthetic target images [6], [11]), we train *our* CoVR-BLIP-2 model on *their* data in a controlled experiment. We summarize the results in Table 7. In the top block, we show numbers directly taken from the CompoDiff [6] work, comparing their method trained on various synthetic datasets: InstructPix2Pix dataset of 1-million triplets [11], a 1M subset of their SynthTriplets [6], and the full 18M version. Note that we use the same metrics as in their paper (e.g., average of R@1, R_s@1 for CIRR) and show the results with their ViT-L model. In the bottom block, we train our model on the 1M synthetic dataset versions, as well as a 1M subset of our real visual data from WV-CC-CoVIR. When comparing the two models on the same training data, we observe similar performances (e.g., 28.32 vs 28.44 on CIRR), suggesting that the main difference comes from training data, rather than the model. Training on our WV-CC-CoVIR with 1M triplets outperform SynthTriplets with both 1M and 18M versions, highlighting the importance of real training images.

4.6 Ablation studies

We now ablate the importance of several key aspects of our method by focusing primarily on experiments trained in WebVid-CoVR, and also examine the impact of different datasets on model performance.

The additional composed caption retrieval loss. As explained in Section 3.3, we integrate a caption retrieval loss term as additional supervision. For both methods (CoVR-BLIP [57] and CoVR-BLIP-2), this led to a significant improvement in the CoIR performance and a slight decrease on WebVid-CoVR on which it was originally trained on, see Table 6 (34 vs 41 R@1 on CIRR and 27 vs 36 R@1 on FashionIQ for instance).

Importance of data scale. In Table 9, we evaluate the effect of the number of video-caption pairs used as a seed for our triplet generation pipeline. We construct subsets of videos such that larger ones include smaller ones, and only keep triplets that contain the sampled videos for training. We find that results steadily increase when using more videos, demonstrating that our method largely benefits from scaling the size of the seed dataset of video-captions. We also observe the importance of the filtering techniques described in Section 3.1, as the model trained on unfiltered data underperforms.

Modification text generation. We use a large language model finetuned for modification text generation (MTG-LLM) as explained in Section 3.1. We here compare this solution to prompting it without any training and to a simple rule-based baseline that uses several templates to generate the modification text given the two captions that differ by one word. For prompting, we prepend few-shot examples of pairs of captions and desired generated texts, before adding the two captions in question. Please check Section C.5 of the Appendix for the full prompt. Table 8 shows that finetuning the MTG-LLM for generating the training data is much more effective than prompting it without finetuning, as measured by CoVR performance on WebVid-CoVR-Test and CoIR performance on CIRR. For the rule-based experiment, the modification text is based on the two different words from the captions. We generate templates that use these words and choose one at random during training. These templates include variations such as "Remove `txt_diff`$_1$" and *"Change* `txt_diff`$_1$ *for* `txt_diff`$_2$". A full list of all the templates can be seen in Section C.3 of the Appendix. Additionally, we investigate the possibility of paraphrasing the rule-based modification texts using `GPT-3.5-turbo` from OpenAI [70] as a source of augmentation, by prompting *"Paraphrase the following sentence:* `{Rule-base modification text}`". In preliminary analysis, we qualitatively observed that LLaMA [10] and LLaMA 2 [71] alternatives were overly verbose when used for paraphrasing; however, GPT-3.5 outputs were satisfactory.

In Table 8, we show that our MTG-LLM generates better modification texts than the rule-based baseline, by evaluating the results of the model trained on the generated data. Paraphrasing the rule-based examples significantly boosts the performance (from 39 to 57.9 R@1), while still being worse than our MTG-LLM, especially on the CIRR benchmark. Note that the paraphrasing comes with the cost of running an expensive LLM ($43 cost for this experiment for 1 paraphrasing per modification text on the entire dataset). On the other hand, our MTG-LLM finetuning only requires 715 text examples. Qualitative examples comparing MTG-LLM and rule-based are provided in Table A.11 of the Appendix.

Training strategies. In Table 10, we first show the benefit

TABLE 8
Modification text generation: We compare our finetuned model MTG-LLM (LLaMA 7B parameters) against (a) a rule-based MTG baseline, (b) a paraphrased rule-based MTG baseline (using GPT-3.5-turbo from OpenAI), and (c) simply prompting the frozen LLaMA LLM. We observe important gains in the downstream performance of the model trained on the generated data.

Model	WebVid-CoVR-Test R@1	R@5	R@10	R@50	CIRR R@1	R@5	R@10	R@50
Rule-based	39.08	67.33	78.13	93.82	12.02	34.75	47.06	75.35
Rule-based & GPT-paraphrased	57.94	81.85	89.79	97.93	32.89	61.98	73.04	90.34
Prompting LLaMA	56.46	82.08	89.32	97.85	34.27	64.29	75.76	91.61
MTG-LLM	**59.82**	**83.84**	**91.28**	**98.24**	**41.42**	**72.58**	**82.55**	**96.31**

TABLE 9
Data size: We measure the importance of the number of videos used for data generation and of filtering the generated data, by evaluating on WebVid-CoVR-Test, CIRR, and FashionIQ. All models are trained for the same number of iterations on the generated data.

Initial #videos	*Generated* #triplets	Filtering	WebVid-CoVR-Test R@1	MeanR	CIRR R@1	MeanR	FashionIQ R@10	R@50	CIRCO mAP@5
0	-	-	16.55	36.15	18.60	47.82	9.75	21.09	4.83
200k	11k	✓	51.53	77.92	40.72	70.82	34.51	55.82	24.00
500k	66k	✓	55.13	80.59	40.22	70.67	36.04	57.10	25.86
1M	269k	✓	57.32	82.13	40.55	71.08	36.76	57.13	26.92
2.5M	1.6M	✓	**59.82**	**83.30**	**41.42**	**73.22**	36.81	56.70	27.84
2.5M	3.6M	✗	58.65	82.57	40.84	71.11	**37.19**	**57.19**	**29.11**

TABLE 10
Training strategies: Iterating on batches of distinct target videos (instead of triplets) and up-sampling hard negatives both benefit the CoVR/CoIR performance.

Iteration	HN-NCE [15]	WebVid-CoVR-Test R@1	R@5	R@10	R@50	CIRR R@1	R@5	R@10	R@50
Triplets	✓	53.79	81.69	88.97	97.89	37.49	66.63	77.11	91.28
Videos	✗	55.24	81.34	89.48	98.24	38.70	68.84	78.41	92.72
Videos	✓	**59.82**	**83.84**	**91.28**	**98.24**	**41.42**	**72.58**	**82.55**	**96.31**

on WebVid-CoVR of training by iterating on target videos instead of CoVR triplets. This is to avoid having the same target video appearing multiple times in a training batch, hence increasing the number of correct negatives that are used in the contrastive loss. Furthermore, up-sampling hard negatives adopting the HN-NCE loss formulation from [15] also slightly benefits the performance.

4.7 Qualitative analysis

In this section, we provide qualitative examples of our WebVid-CoVR and CC-CoIR triplets. Figure 5, shows examples of triplets generated using our automatic dataset creation. These examples demonstrate the effectiveness of our approach in generating coherent modification texts for paired videos. For more examples, we refer to Section E.5 of the Appendix.

5 Conclusions and Limitations

In this work, we studied the new task of CoVR by proposing a simple yet effective methodology to create automatic training data. Our results on several benchmarks (including our manually curated video benchmark, as well as existing image benchmarks) suggest that, while noisy, such an automated and scalable approach can provide effective CoVR model training. One potential limitation of our method is that the generated modification text may not depict some visible changes due to not considering the image pair, but only their captions. Moreover, our modification text is suboptimal due to only inputting one-word difference caption pairs (i.e., focusing only on one change, and not considering multi-word differences). For example, the following modification with multiple changes from the CIRR dataset would not exist in our data "close up of a similar dog, but it is swimming on its own with a tennis ball in its mouth". Future work can incorporate visually-grounded modification generation and multiple modifications between query and target video pairs.

Ethics Statement

Our model constitutes a generic multi-modal search tool, but is not intended for a specific application. While there are helpful use cases such as online shopping, traveling, and personal development (i.e., how-to), there may be potential privacy and harmful risks when training the model on datasets with inappropriate content. The risks include surveillance applications such as searching for a specific person, and looking up violent and graphic videos. For our WebVid-CoVR data release, we refer to Section A for further analysis about removal of inappropriate content. We note that our dataset users must also adhere to the terms of use stipulated by WebVid [12].

Fig. 5. **Examples of generated triplets:** We illustrate triplet samples generated using our automatic dataset creation methodology (left: WebVid-CoVR, right: CC-CoIR). Each sample consists of two videos/images with their corresponding captions (at the bottom of each video/image) and the generated modification text using our MTG-LLM (in purple).

ACKNOWLEDGEMENTS

This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011014223 made by GENCI. The authors would like to acknowledge the research gift from Google, the ANR project CorVis ANR-21-CE23-0003-01, Antoine Yang's Google PhD fellowship, and thank Mathis Petrovich, Nicolas Dufour, Charles Raude, and Andrea Blazquez for their helpful feedback.

REFERENCES

[1] J. Li, D. Li, C. Xiong, and S. C. H. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *ICML*, 2022. 1, 3, 5, 6, 19

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, 2021. 1, 2, 3, 5, 6, 15, 16

[3] N. Vo, L. Jiang, C. Sun, K. Murphy, L.-J. Li, L. Fei-Fei, and J. Hays, "Composing text and image for image retrieval - an empirical odyssey," in *CVPR*, 2019. 1, 2, 8

[4] A. Baldrati, L. Agnolucci, M. Bertini, and A. Del Bimbo, "Zero-shot composed image retrieval with textual inversion," *ICCV*, 2023. 1, 2, 6, 8

[5] A. Baldrati, M. Bertini, T. Uricchio, and A. D. Bimbo, "Effective conditioned and composed image retrieval combining CLIP-based features," in *CVPR*, 2022. 1, 2, 6, 7, 8

[6] G. Gu, S. Chun, W. Kim, H. Jun, Y. Kang, and S. Yun, "Compodiff: Versatile composed image retrieval with latent diffusion," in *TMLR*, 2024. 1, 2, 3, 8, 9

[7] M. Levy, R. Ben-Ari, N. Darshan, and D. Lischinski, "Data roaming and early fusion for composed image retrieval," in *AAAI*, 2024. 1, 2, 3, 6, 8

[8] Z. Liu, C. Rodriguez-Opazo, D. Teney, and S. Gould, "Image retrieval on real-life images with pre-trained vision-and-language models," in *ICCV*, 2021. 1, 2, 6, 8

[9] H. Wu, Y. Gao, X. Guo, Z. Al-Halah, S. Rennie, K. Grauman, and R. Feris, "Fashion IQ: A new dataset towards retrieving images by natural language feedback," in *CVPR*, 2021. 1, 2, 6

[10] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," *arXiv:2302.13971*, 2023. 1, 4, 6, 9

[11] T. Brooks, A. Holynski, and A. A. Efros, "InstructPix2Pix: Learning to follow image editing instructions," in *CVPR*, 2023. 2, 3, 4, 9, 16, 17, 18

[12] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in time: A joint video and image encoder for end-to-end retrieval," in *ICCV*, 2021. 2, 3, 4, 10, 13, 20

[13] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *ACL*, 2018. 2, 3, 7, 8, 16

[14] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv:1807.03748*, 2018. 2, 5

[15] F. Radenovic, A. Dubey, A. Kadian, T. Mihaylov, S. Vandenhende, Y. Patel, Y. Wen, V. Ramanathan, and D. Mahajan, "Filtering, distillation, and hard negatives for vision-language pre-training," in *CVPR*, 2023. 2, 5, 6, 10

[16] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *ICML*, 2023. 2, 3, 5, 6, 19

[17] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "A CLIP-hitchhiker's guide to long video retrieval," *arXiv:2205.08508*, 2022. 2, 3, 5

[18] X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. S. Feris, "Dialog-based interactive image retrieval," in *NeurIPS*, 2018. 2

[19] G. Delmas, R. S. de Rezende, G. Csurka, and D. Larlus, "ARTEMIS: Attention-based retrieval with text-explicit matching and implicit similarity," in *ICLR*, 2022. 2, 8

[20] J. Kim, Y. Yu, H. Kim, and G. Kim, "Dual compositional learning in interactive image retrieval," *AAAI*, 2021. 2, 8

[21] K. Saito, K. Sohn, X. Zhang, C.-L. Li, C.-Y. Lee, K. Saenko, and T. Pfister, "Pic2Word: Mapping pictures to words for zero-shot composed image retrieval," *CVPR*, 2023. 2, 8

[22] K. Zhang, Y. Luan, H. Hu, K. Lee, S. Qiao, W. Chen, Y. Su, and M.-W. Chang, "MagicLens: Self-supervised image retrieval with open-ended instructions," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 21–27 Jul 2024, pp. 59 403–59 420. 2

[23] S. Sun, F. Ye, and S. Gong, "Training-free zero-shot composed image retrieval with local concept reranking," *arXiv:2312.08924*, 2023. 2, 8

[24] G. Gu, S. Chun, W. Kim, , Y. Kang, and S. Yun, "Language-only training of zero-shot composed image retrieval," in *CVPR*, 2024. 2, 8, 9

[25] S. Karthik, K. Roth, M. Mancini, and Z. Akata, "Vision-by-language for training-free compositional image retrieval," in *ICLR*, 2024. 2, 8

[26] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *ICCV*, 2015. 3

[27] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," in *NeurIPS*, 2022. 3

[28] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "UNITER: Universal image-text representation learning," in *ECCV*, 2020. 3

[29] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," in *NeurIPS*, 2021. 3

[30] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *ECCV*, 2020. 3

[31] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *ICML*, 2021. 3

[32] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VL-BERT: Pre-training of generic visual-linguistic representations," in *ICLR*, 2019. 3

[33] C. Schuhmann, R. Beaumont, R. Vencu, C. W. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman *et al.*, "LAION-5B: An open large-scale dataset for training next generation image-text models," in *NeurIPS Datasets and Benchmarks Track*, 2022. 3

[34] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong, "Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text," *NeurIPS*, 2021. 3

[35] D. Li, J. Li, H. Li, J. C. Niebles, and S. C. Hoi, "Align and prompt: Video-and-language pre-training with entity prompts," in *CVPR*, 2022. 3

[36] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, "HERO: Hierarchical encoder for video+language omni-representation pre-training," in *EMNLP*, 2020. 3

[37] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips," in *ICCV*, 2019. 3

[38] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *CVPR*, 2020. 3

[39] H. Xu, G. Ghosh, P.-Y. Huang, D. Okhonko, A. Aghajanyan, F. Metze, L. Zettlemoyer, and C. Feichtenhofer, "VideoCLIP: Contrastive pre-training for zero-shot video-text understanding," in *EMNLP*, 2021. 3

[40] Y. Sun, H. Xue, R. Song, B. Liu, H. Yang, and J. Fu, "Long-form video-language pre-training with multimodal temporal contrastive learning," in *NeurIPS*, 2022. 3

[41] H. Xue, T. Hang, Y. Zeng, Y. Sun, B. Liu, H. Yang, J. Fu, and B. Guo, "Advancing high-resolution video-language representation with large-scale video transcriptions," in *CVPR*, 2022. 3

[42] Y. Zhao, I. Misra, P. Krähenbühl, and R. Girdhar, "Learning video representations from large language models," in *CVPR*, 2023. 3

[43] A. Nagrani, C. Sun, D. Ross, R. Sukthankar, C. Schmid, and A. Zisserman, "Speech2action: Cross-modal supervision for action recognition," in *CVPR*, 2020. 3

[44] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, "Just ask: Learning to answer questions from millions of narrated videos," in *ICCV*, 2021. 3

[45] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *NeurIPS*, 2023. 3

[46] H. Fang, P. Xiong, L. Xu, and Y. Chen, "CLIP2Video: Mastering video-text retrieval via image clip," *arXiv:2106.11097*, 2021. 3

[47] Z. Gao, J. Liu, S. Chen, D. Chang, H. Zhang, and J. Yuan, "CLIP2TV: an empirical study on transformer-based methods for video-text retrieval," *arXiv:2111.05610*, 2021. 3

[48] Y. Ge, Y. Ge, X. Liu, D. Li, Y. Shan, X. Qie, and P. Luo, "BridgeFormer: Bridging video-text retrieval with multiple choice questions," in *CVPR*, 2022. 3

[49] Y. Liu, P. Xiong, L. Xu, S. Cao, and Q. Jin, "TS2-Net: Token shift and selection transformer for text-video retrieval," in *ECCV*, 2022. 3

[50] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, and T. Li, "CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval," *arXiv:2104.08860*, 2021. 3

[51] Y. Ma, G. Xu, X. Sun, M. Yan, J. Zhang, and R. Ji, "X-CLIP: End-to-end multi-grained contrastive learning for video-text retrieval," in *ACMMM*, 2022. 3

[52] H. Rasheed, M. U. Khattak, M. Maaz, S. Khan, and F. S. Khan, "Fine-tuned CLIP models are efficient video learners," in *CVPR*, 2023. 3

[53] H. Xue, Y. Sun, B. Liu, J. Fu, R. Song, H. Li, and J. Luo, "CLIP-vip: Adapting pre-trained image-text model to video-language alignment," in *ICLR*, 2023. 3

[54] J. Yang, Y. Bisk, and J. Gao, "TACo: Token-aware cascade contrastive learning for video-text alignment," in *ICCV*, 2021. 3

[55] L. Yao, R. Huang, L. Hou, G. Lu, M. Niu, H. Xu, X. Liang, Z. Li, X. Jiang, and C. Xu, "FILIP: Fine-grained interactive language-image pre-training," in *ICLR*, 2022. 3

[56] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," in *ACL*, 2018. 4

[57] L. Ventura, A. Yang, C. Schmid, and G. Varol, "CoVR: Learning composed video retrieval from web video captions," *AAAI*, 2024. 5, 8, 9

[58] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi, "A corpus for reasoning about natural language grounded in photographs," in *ACL*, 2019. 6

[59] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva: Exploring the limits of masked visual representation learning at scale," 2022. 6

[60] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014. 8, 19

[61] Y. Yu, S. Lee, Y. Choi, and G. Kim, "CurlingNet: Compositional learning between images and text for fashionIQ data," *ICCV Workshop*, 2019. 8

[62] Y. Chen and L. Bazzani, "Learning joint visual semantic matching embeddings for language-guided retrieval," in *ECCV*, 2020. 8

[63] S. Jandial, A. Chopra, P. Badjatiya, P. Chawla, M. Sarkar, and B. Krishnamurthy, "TRACE: Transform aggregate and compose visiolinguistic representations for image search with text feedback," *arXiv:2009.01485*, 2020. 8

[64] Y. Chen, S. Gong, and L. Bazzani, "Image search with text feedback by visiolinguistic attention learning," in *CVPR*, 2020. 8

[65] E. Dodds, J. Culpepper, S. Herdade, Y. Zhang, and K. Boakye, "Modality-agnostic attention fusion for visual search with text feedback," *arXiv:2007.00145*, 2020. 8

[66] M. Shin, Y. Cho, B. Ko, and G. Gu, "RTIC: Residual learning for text and image composition using graph convolutional network," *arXiv:2104.03015*, 2021. 8

[67] S. Lee, D. Kim, and B. Han, "CoSMo: Content-style modulation for image retrieval with text feedback," in *CVPR*, 2021. 8

[68] S. Jandial, P. Badjatiya, P. Chawla, A. Chopra, M. Sarkar, and B. Krishnamurthy, "SAC: Semantic attention composition for text-conditioned image retrieval," in *WACV*, 2022. 8

[69] S. Goenka, Z. Zheng, A. Jaiswal, R. Chada, Y. Wu, V. Hedau, and P. Natarajan, "FashionVLP: Vision language transformer for fashion retrieval with feedback," in *CVPR*, 2022. 8

[70] T. Brown et al., "Language models are few-shot learners," in *NeurIPS*, 2020. 9

[71] H. Touvron et al., "LLaMA 2: Open foundation and fine-tuned chat models," *arXiv:2307.09288*, 2023. 9
[72] S. Loria, textblob.readthedocs.io, 2013. 13
[73] S. A. Lab, https://huggingface.co/datasets/shinonomelab/cleanvid-15m_map, 2023. 13, 14, 15
[74] S. N. Thanh, https://pypi.org/project/better-profanity/, 2018. 13
[75] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, 2019. 16
[76] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019. 17
[77] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*. Springer Berlin Heidelberg, 2003. 19

APPENDIX

This document provides WebVid-CoVR dataset statistics (Section A), CC-CoIR dataset statistics (Section B), implementation details (Section C), additional experiments (Section D), and additional qualitative examples (Section E). We also provide the code and dataset together with a datasheet, and an illustrative video on our project page at imagine.enpc.fr/~ventural/covr.

A WebVid-CoVR dataset statistics and analysis

In this section, we provide analysis on our WebVid-CoVR. A detailed datasheet can be found as a separate file.

Filtering inappropriate content and vulgar language. We take several measures to detect semi-automatically any inappropriate content, and remove such instances from our dataset. To achieve this, we use a combination of tools (such as negative sentiment and profanity detectors) and apply them on modification texts and video captions.

We conduct a sentiment analysis on the modification texts using the `TextBlob` library [72] to identify instances of negative sentiment. We find that less than 0.5% of the dataset (about 2k instances) exhibits negative sentiment. Upon manual review, we identify false positives in this categorization, including examples such as "make it an evil pumpkin" or "Change him into a frustrated businessman". The instances detected as negative sentiment are reviewed and 260 of them are removed from the dataset. We ensure that the dataset does not include any videos marked for mature content, by checking the metadata of WebVid [12] provided by [73]. Finally, using the `better-profanity` library [74], we identify approximately 2k video captions that are marked for profanity. Upon manual inspection, we find that there were a large number of videos displaying computer-generated visuals with those words. We also notice false positives (e.g., misinterpretation due to context), such as the animal cock being incorrectly identified as profanity. The videos detected to contain profanity in their captions are reviewed and excluded from the dataset.

Distribution of caption and video embedding similarities. As explained in Section 3.1 of the main paper, we filter caption pairs with CLIP text embedding similarity ≥ 0.96 and caption pairs with CLIP text embedding similarity ≤ 0.6, and for each caption pair, we choose the 10 video pairs with the highest CLIP visual similarity computed at the middle frame of the videos. We also note that our cosine similarities are normalized between [0, 1]. Here, we further show the distribution of text embedding similarity in caption pairs and visual embedding similarity in video pairs in Figure A.1. The distribution of video similarity scores exhibits two distinct peaks. The first peak corresponds to a score of approximately 0.7 and includes video pairs that are significantly dissimilar. The second peak corresponds to a score close to 1.0 and represents video pairs with highly similar visual content.

Number of words in modification texts. Figure A.2 further provides the histogram of the number of words in the generated modification text. We observe that the majority of texts contain 3-8 words.

Number of triplets per target video. In Section 3.2 of the main paper, we provided several statistics about our WebVid-CoVR dataset, e.g., on average, a target video is associated with 12.7 triplets. However, in Figure A.3, when visualizing the distribution of triplets associated with each target video, we see that the histogram reveals that the majority of target videos are associated to only 1 or 2 triplets. The histogram exhibits a long tail, i.e., a small subset of target videos have a considerably larger number of triplets associated. These videos have captions such as "Mountain landscape", "Water stream", and "Water river", leading to numerous one-word difference captions associated with them.

Video categories. We plot the distribution of video categories in Figure A.4. These categories are found using the WebVid metadata provided by [73]. We find 50% of WebVid-CoVR videos in this metadata collection. Note more than one category can be associated with a single video (e.g., Nature and Animals/Wildlife for a video of a fish in the ocean).

Distribution of part-of-speech (POS) tags. We conducted POS tagging on the modification texts within the WebVid-CoVR dataset to analyze their distribution. The resulting analysis reveals the average counts of different parts of speech per modification text, including Nouns, Verbs, Pronouns, Adjectives, and Adverbs. We plot the distribution in Figure A.5, and see that, on average, a modification text contains 1.6 nouns and 1.1 verbs, emphasizing the prevalent use of nouns and verbs in the dataset's modifications. The most frequently encountered words within each category's top 3 are as follows: Noun: *symbol, water, forest.* Verb: *make, turn, change.* Pronoun: *it, them, her.* Adjective: *green, more, black.* Adverb: *instead, more, then.* We also include a visualization of the verb-noun frequency heatmap in Figure A.6, which provides insights into the distribution of verb-noun count combinations across modification texts in our dataset. From the heatmap, we observe that over 60% of the sentences exhibit a pattern of having one verb paired with one or two nouns.

We also conducted an analysis using POS tagging on the video *captions*. Figure A.7 visually illustrates the transition of POS tags across the difference words in Caption 1 and Caption 2. We observe a predominant pattern of noun-to-noun changes in our caption pairs.

Source of noise. As mentioned in Section 3.2 of the main paper, about 22% of the automatic collection can be considered as noisy, because this was the percentage of discarded triplets when manually curating the WebVid-CoVR test set. We expect a similar noise ratio in the training set. To inspect the noise in detail, we manually went over the triplet examples that were marked as unsuitable (therefore discarded) when annotating the test set. We marked whether the reason for discarding falls within any of the following categories, and computed the following percentages (normalized by the number of discarded triplets).

- 35%: The generated modification text does not describe the visual difference. Primarily attributed to either the quality of the video captions or the output generated by the MTG-LLM.
- 28%: Paired videos are visually too similar.
- 15%: Paired videos are visually too different.
- 13%: At least one of the videos is difficult to understand/low quality.
- 9%: Captions are too similar (e.g., one-word difference does not change the meaning: "On the chairlift" and "Ride the chairlift").

While the first category of errors is the largest, it is important to also note that our strict standards for the test set necessitated the discarding of many triplets that could potentially be useful for training.

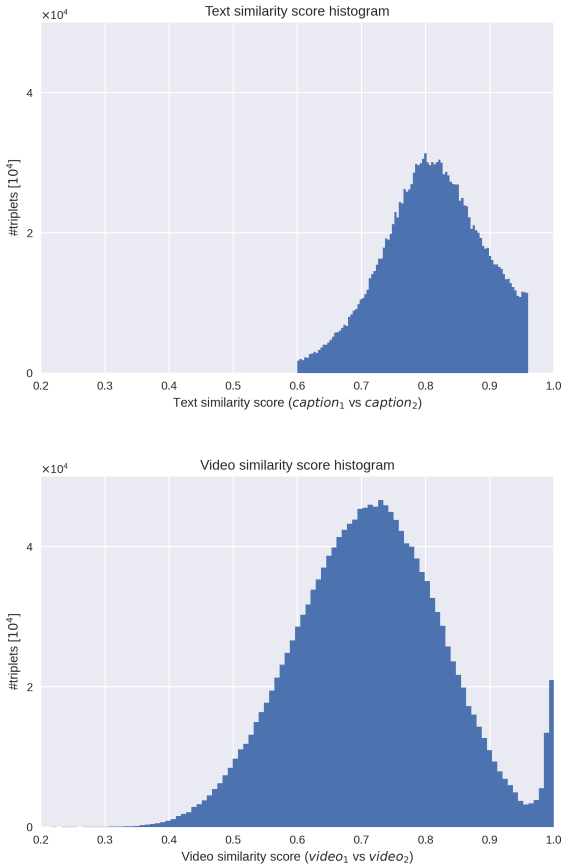

Fig. A.1. **Text/video similarity of the caption/video pairs:** Distribution of text similarity scores between caption pairs $(caption_1, caption_2)$ (left) and video similarity scores between video pairs $(video_1, video_2)$ (right), using CLIP embeddings and cosine similarity.

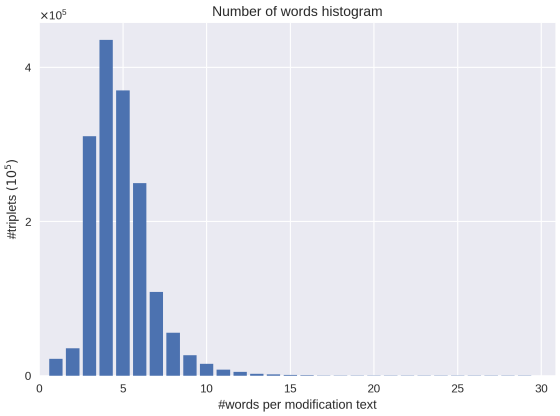

Fig. A.2. **Histogram of the number of words in the generated modification text:** Most modification texts have between 3 and 8 words.

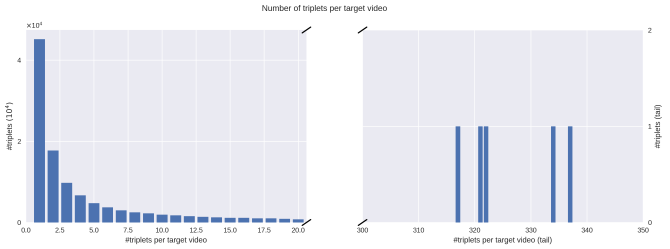

Fig. A.3. **Distribution of number of triplets per target video:** We display the histogram depicting the number of triplets associated with each target video in the WebVid-CoVR dataset. Most target videos have 1 or 2 triplets and certain videos exhibit a high number of triplets (zoomed in to the tail on the right plot), e.g., some target videos are present in over 300 triplets, highlighting the variability in modification texts.

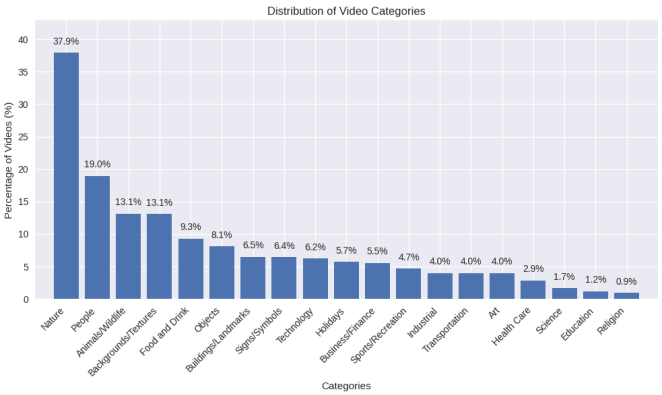

Fig. A.4. **Distribution of video categories:** We plot the distribution of categories for videos in WebVid-CoVR, as provided by [73] as WebVid metadata. Note that 50% of our WebVid-CoVR videos are present in this metadata collection. Looking at the distribution, we observe that around 40% and 20% of WebVid-CoVR are videos of Nature and People, respectively.

WebVid-CoVR dataset overlap with zero-shot CoIR evaluation datasets. To contextualize the zero-shot performance, we analyze the potential overlap between our WebVid-CoVR dataset and the three CoIR datasets. We compute the CLIP [2] embeddings for all target videos in the CC-CoIR training set and calculate their similarity with the *target* images in the test sets of each CoIR dataset. We then evaluate the overlap by setting similarity thresholds at 0.7, 0.8, and 0.9. We define overlap as occurring when at least one sample in the test set has a similarity score above the threshold. We note that for simplicity, we only consider target images.

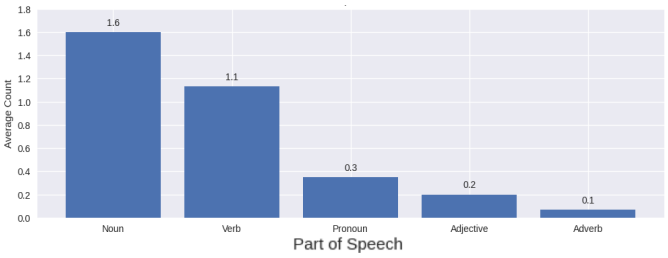

Fig. A.5. **Distribution of parts of speech in modification texts:** Number of nouns, verbs, pronouns, adjectives, and adverbs in the modification text using part-of-speech (POS) tagging. On average, there are more than one noun and one verb per modification text.

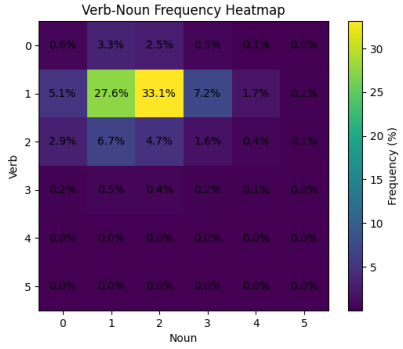

Fig. A.6. **Verb-noun heatmap:** This heatmap illustrates the percentage of modification texts containing specific combinations of verbs and nouns. Each cell represents the frequency of a particular verb-noun combination, and the values are presented as percentages. The color intensity indicates the relative frequency of occurrence. We observe that over 60% of the sentences exhibit a pattern of having one verb paired with one or two nouns.

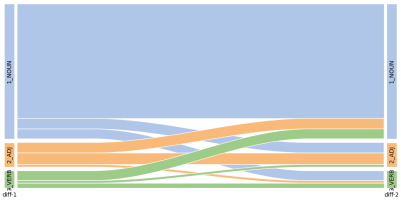

Fig. A.7. **Transition of POS tags across the difference words between the two captions:** The visualization primarily focuses on nouns, adjectives, and verbs, which constitute a significant proportion of modifications at 87% (comprising 65% nouns, 13% adjectives, and 9% verbs). The remaining words fall into categories where the POS tagger was unable to classify the word (12%) or adverbs (<1%).

The results, summarized in Table A.1, show that no target videos from the WebVid-CoVR dataset exhibit a similarity score higher than 0.9 with any of the CoIR datasets, indicating zero overlap at the highest threshold. However, as the threshold decreases, the overlap increases, specially for CIRCO, which shows a 12.6% overlap at a 0.8 threshold.

B CC-CoIR DATASET STATISTICS

In this section, we provide analysis on our CC-CoIR. We start with 3.3M caption-image pairs, with 2M distinct captions. As explained in Section 3.1 of the main paper, we mine paired images by searching for captions that differ by a single

TABLE A.1
Overlap between WebVid-CoVR training data and zero-shot CoIR benchmarks: We present the percentage overlap between target videos in our WebVid-CoVR training dataset and the target images in the test sets of three CoIR datasets at different CLIP cosine similarity thresholds (0.7, 0.8, and 0.9). Overlap is defined as the presence of at least one target image in the test set with a similarity score above the specified threshold. The results indicate no overlap at the highest threshold (0.9). Note that this analysis focuses on the overlap of target images, not triplets.

Threshold	CIRR	FashionIQ Dress	FashionIQ Shirt	FashionIQ Toptee	CIRCO
0.7	45.4%	13.6%	13.8%	10.9%	79.0%
0.8	1.6%	0.3%	0.0%	0.0%	12.6%
0.9	0.0%	0.0%	0.0%	0.0%	0.0%

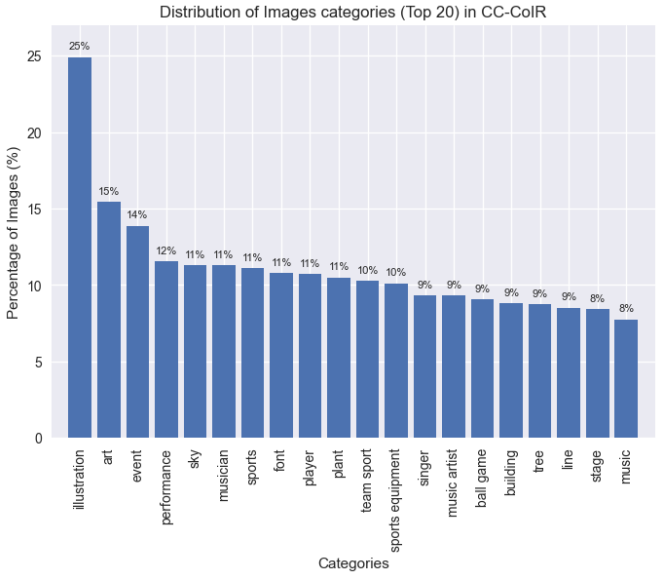

Fig. A.8. **Distribution of image categories (Top 20) in CC-CoIR:** We plot the distribution of categories for images in CC-CoIR, as provided by [13]. Note that 67% of our CC-CoIR images have one or more categories in this metadata collection. Looking at the distribution, we observe that around 25% and 15% of WebVid-CoVR are videos of illustration and art, respectively.

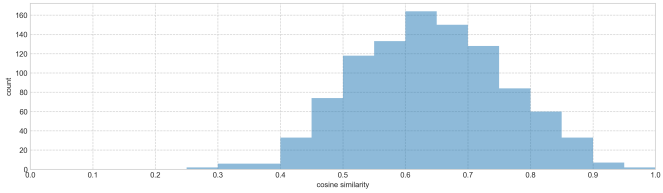

Fig. A.9. **Cosine similarity histogram between target image and generated image:** We plot the distribution of cosine similarity scores between target images and images generated using InstructPix2Pix [11] based on the corresponding source image and modification text. The results suggest that the majority of target images exhibit a similarity to the generated images.

TABLE A.2
Overlap between CC-CoIR training data and zero-shot CoIR benchmarks: We repeat the overlap analysis with CC-CoIR (as is similarly done for WebVid-CoVR in Table A.1). The results indicate minimal overlap at the highest threshold (0.9), with fewer than 2% similarity across all datasets, suggesting limited direct overlap.

Threshold	CIRR	FashionIQ Dress	FashionIQ Shirt	FashionIQ Toptee	CIRCO
0.7	34.0%	6.9%	10.8%	9.6%	59.1%
0.8	5.2%	0.2%	0.3%	0.2%	20.7%
0.9	0.2%	0.0%	0.0%	0.0%	1.6%

word, excluding punctuation marks. This process allows us to identify a vast pool of 1.2M distinct caption pairs with 281k distinct captions, resulting in 3.3M triplets after filtering.

CC-CoIR image categories. Figure A.8 illustrates the distribution of the top 20 categories derived from our CC-CoIR dataset. We find 67% of images within our CC-CoIR dataset possess one or more associated categories. It's worth mentioning that a single video may be associated with multiple categories simultaneously. For instance, a video featuring a singer may be categorized under both "Singer" and "Music Artist". It is important to note that while only the top 20 are displayed, the complete dataset encompasses over 10,000 distinct categories. This highlights the wide variety of visual content present within our collection.

Quantifying noise in CC-CoIR. Here, we attempt to quantify how well the modification text describes the transformation of the query image into the target image. While this task is challenging, one possible approach involves leveraging the InstructPix2Pix [11] model. We input the source image and corresponding modification text into the model to generate an image that reflects the intended transformation. The results, summarized in Figure A.9, reveal that the majority of target images have a cosine similarity "close" to that of the generated images. This suggests that the modification text generally aligns with the visual changes captured in the target images. However, a qualitative examination of the generated images indicates that they cannot always serve as ground truth, suggesting that these results should be interpreted with caution.

CC-CoIR dataset overlap with zero-shot CoIR evaluation datasets. As previously done for WebVid-CoVR, we now analyze the overlap between our CC-CoIR dataset and the CoIR datasets using CLIP [2] embeddings. The results, summarized in Table A.2, reveal that fewer than 2% of the target images in the CC-CoIR training set have a similarity score higher than 0.9 in any of the CoIR datasets. When the threshold is relaxed to 0.8, the overlap increases, with CIRCO showing 20.7% and CIRR showing 5.2% of samples with a similarity score above the threshold.

C Implementation Details

We describe the dataset generation computation time (Section C.1), further training details (Section C.2), provide the templates we use for our rule-based baseline (Section C.3), details about our MTG-LLM finetuning and inference (Section C.4), and prompt to our prompting experiment (Section C.5.

C.1 Dataset generation computation time

We outline the detailed computation time for each step of the dataset generation. The computation times below are obtained using a **single** NVIDIA RTX A6000, but it is important to note that most of the processes can be parallelized, which would significantly reduce the wallclock time required. In practice, we used 2 GPUs.

- **Text embedding extraction:** We extracted text embeddings from 2 million distinct captions out of a total of 2.4 million video-caption pairs. This process completed in less than 2 hours.
- **Caption similarity search:** To identify captions with one-word differences, we employed the *faiss* library [75] to select the 100 closest captions, avoiding the need to compare each caption against the entire set of 2 million captions. This optimization significantly reduced the search time, resulting in 2.5 hours.
- **Text similarity filtering:** Thanks to the precomputed text embeddings, the text similarity filtering step

TABLE A.3
Rule-based templates: For our rule-based MTG baseline, we randomly choose one of the below templates during training.

Remove `txt_diff`$_1$
Take out `txt_diff`$_1$ *and add* `txt_diff`$_2$
Change `txt_diff`$_1$ *for* `txt_diff`$_2$
Replace `txt_diff`$_1$ *with* `txt_diff`$_2$
Replace `txt_diff`$_1$ *by* `txt_diff`$_2$
Replace `txt_diff`$_1$ *with* `txt_diff`$_2$
Make the `txt_diff`$_1$ *into* `txt_diff`$_2$
Add `txt_diff`$_2$
Change it to `txt_diff`$_2$

incurred no additional time overhead. All the text filtering processes were completed in less than 5 minutes, even on a large pool of 1.2 million captions.

- **Video similarity computation:** To filter by video similarity, we extracted the middle frame from approximately 135,000 videos and computed CLIP embeddings. This step takes approximately 3 hours.
- **MTG-LLM model finetuning:** Finetuning for 715 examples takes less than 10 minutes. Note that the time required to finetune the MTG-LLM model is independent of the number of CoVR triplets we generate.
- **Modification text generation:** This is the most time-consuming stage of the pipeline. It takes around 24 hours to process the 1.6 million caption pairs.

C.2 Training details

Here, we provide implementation details in addition to Section 4.1 of the main paper. In terms of the optimization algorithm, we utilize AdamW [76]. For our MTG-LLM, we finetune for one epoch with a batch size of 128 and a learning of $3\mathrm{e}{-5}$ that is warmed up linearly for the first 100 steps and then kept constant. For our CoVR model, keeping the visual backbone frozen largely improves the efficiency of the training process: an epoch on the CIRR dataset takes 4 minutes with a frozen backbone and 25 minutes with a finetuned backbone, while leading to similar performance. During the training process, we employ several image data augmentations. These transformations include a random resized crop, where the input image is resized to a resolution of 384×384. Additionally, we apply a random horizontal flip and random adjustments to contrast, brightness, sharpness, translation, and rotation. We use a weight decay of 0.05 and an initial learning rate of $1\mathrm{e}{-5}$ that is decayed to 0 following a cosine schedule over 10 epochs.

C.3 List of rule-based templates

In the ablation studies (Section 4.6 of the main paper), we introduced a rule-based MTG baseline. Here, in Table A.3, we show the templates used for the rules. We refer to Section E.2 (Table A.11) for qualitative comparison with our finetuned MTG-LLM.

C.4 Generating a modification text from paired captions with MTG-LLM

As described in Section 3.1 of the main paper, we use top-k sampling at inference for the MTG-LLM. Specifically, we use $k = 200$ and $temperature = 0.8$. We further give details about the text input-output format for the MTG-LLM. At training, we form the input prompt by concatenating captions and target and adding delimiters and stop sequences similar to InstructPix2Pix [11]. In detail, given a caption pair $(caption_1, caption_2)$ and a corresponding target $Target$, we concatenate them and add a separator in the following way: $caption_1${`separator`}$caption_2$`\n&&\n`$Target$, where `separator` is `\n&&\n`.

For instance, the model takes as input:

```
Clouds in the sky\n&&\nAirplane in the
    sky \n\n### Response:
```

and is trained to generate the response:

```
Clouds in the sky\n&&\nAirplane in the
    sky \n\n### Response: Add an
    airplane
```

At inference, we simply leave the response empty, and let the model autoregressively generate a modification text.

As mentioned in Section 3.1 of the main paper, we add 15 manually prepared text triplets to the existing 700 text triplets from [11] used for training. The motivation is to address specific CoVR cases not present in the original set of triplets, such as *"remove clouds and reveal only sky"* given input captions *"Clouds timelapse"* and *"Sky timelapse"*. We show these 15 samples in Table A.4.

The caption pairs from Table A.4 originate from failure cases of the initial iteration of our MTG-LLM on WebVid-CoVR. We manually corrected some of the failures by typing modification texts without looking at the corresponding videos, i.e., in a way we would want a text-only model to behave, by focusing on the changed words. We note that some cases are ambiguous, especially when the captions are not long or precise enough. For example, 'walking swan' and 'white swan' may not necessarily result in 'change color to white', but there is no way to know without looking at the visual pair, which itself is an interesting area for future work.

C.5 Details of the LLaMA prompt

In the ablation studies (Section 4.6 of the main paper), we justified why we finetuned LLaMA as opposed to simply prompting it without any training. Here, we show how we determine the prompt for the aforementioned experiment. Specifically, we prepend few-shot examples of pairs of captions and desired generated texts, before adding the two captions in question. In particular, we use the following sentence:

```
Clouds in the sky&&Airplane in the sky->
    Add an airplane\n
Aerial view of forest&&Aerial view
    autumn forest -> Change season to
    autumn\n
Clouds timelapse&&Sky timelapse -> remove
    clouds and reveal only sky\n
Aerial view of a sailboat anchored in
    the mediterranean sea.&&Aerial view
    of two sailboat anchored in the
    mediterranean sea.-> Add one
    sailboat\n
```

TABLE A.4
Added examples to the MTG-LLM training: We add the below 15 examples to the set of 700 text triplets from [11].

Caption_1	Clouds in the sky
Caption_2	Airplane in the sky
Target output	Add an airplane
Caption_1	Woman with the tablet computer sitting in the city.
Caption_2	Woman with tablet computer sitting in the park.
Target output	In the park
Caption_1	Walking swan
Caption_2	White swan
Target output	Change color to white
Caption_1	Child playing on beach, sea waves view, girl spinning on coastline in summer 4k
Caption_2	Child playing on beach, sea waves view, girl running on coastline in summer 4k
Target output	Make her spin
Caption_1	Aerial view of forest
Caption_2	Aerial view autumn forest
Target output	Change season to autumn
Caption_1	Palm tree in the wind
Caption_2	Palm trees in the wind
Target output	Add more palm trees
Caption_1	Schoolgirl talking on the phone
Caption_2	Girl talking on the phone
Target output	Make her older
Caption_1	Clouds timelapse
Caption_2	Sky timelapse
Target output	remove clouds and reveal only sky
Caption_1	Aerial view of a sailboat anchored in the mediterranean sea, vathi, greece.
Caption_2	Aerial view of two sailboat anchored in the mediterranean sea, vathi, greece.
Target output	Add one sailboat
Caption_1	France flag waving in the wind. realistic flag background. looped animation background.
Caption_2	Italian flag waving in the wind. realistic flag background. looped animation background.
Target output	Swap the flag for an italian one
Caption_1	Woman jogging with her dog in the park
Caption_2	Woman playing with her dog in the park.
Target output	Stop jogging and make them play
Caption_1	Oil Painting Reproductions of by humans william-glackens
Caption_2	Oil Painting Reproductions of zombies by william-glackens
Target output	Replace the humans with zombies
Caption_1	The girl who loved the sea by banafria
Caption_2	The girl, wearing a hat, who loved the sea by banafria
Target output	Put a hat on her
Caption_1	famous painting Paris, a Rainy Day of Gustave Caillebotte
Caption_2	famous painting Paris, a Sunny Day of Gustave Caillebotte
Target output	Change it to more pleasant weather
Caption_1	Bee on purple flower
Caption_2	Bee on a flower
Target output	Change color of the flower

Then, we concatenate our two captions for which we wish to generate a modification text. The previous results in (Table 8 of the main paper, are also consistent with our qualitative observations: we found that the LLM struggles to perform the modification text generation without finetuning (see Table A.11 in the next section).

D Additional Experiments

We provide additional experiments, reporting CoVR results when changing the visual query from an image to a video (Section D.1), effect of backbones (Section D.2), incorporating visual similarity between videos (Section D.3), results when filtering dynamic or static videos (Section D.4) effect of the modification text length (Section D.5), and optimal number of frames (Section D.6).

D.1 Video query for CoVR

As noted in Section 3 of the main paper, we focus on image queries in this paper. This was because querying with an image has arguably more applications for realistic search scenarios. Here, we explore the setup of using a *video* as the visual query instead of an image query. We can do this since our dataset consists of video-text-video triplets. To encode a query video, we sample 15 equally-spaced frames and compute visual embeddings for each frame using the BLIP-2 image encoder. We then average the per-frame embeddings and forward it through the BLIP-2 cross-attention layers to

TABLE A.5
Querying with a video: We report results on WebVid-CoVR-Test by using multiple frames from the query *video*. Recall that the rest of the paper investigates the setup where the middle video frame is used as an *image* query. We use 5 query video frames (uniformly sampled throughout the video). The number of target video frames remains unchanged as 15. The performance is similar to the image query setup, with marginal increase.

Visual query	R@1	R@5	R@10	R@50
Image (middle frame)	**59.82**	83.84	**91.28**	98.24
Video	59.55	**84.19**	90.85	**98.32**

TABLE A.6
Variants of pretrained BLIP-2 backbones: We compare the BLIP-2 model without finetuning (base) and BLIP-2 finetuned on COCO (the one used in the rest of the paper) [1]. For this experiment, we finetune the models on WebVid-CoVR using the cross-attention layers of BLIP-2 as the fusion method.

Backbone	R@1	R@5	R@10	R@50
BLIP-2 Base	59.66	**84.04**	90.92	**98.32**
BLIP-2 COCO	**59.82**	83.84	**91.28**	98.24

obtain a multimodal query embedding $f(q, t)$. Note that we keep the target video representation fixed to 15 frames with weighted embedding averaging as described in Section 3.3 of the main paper. As seen in Table A.5, using 15 query frames leads to similar performance to using the middle frame.

D.2 Effect of backbones

Variants of pretrained BLIP-2 models. All experiments in this paper are performed with the BLIP-2 model [16] finetuned on COCO [60]. Here, we include experiments when changing this backbone with ViT-L without COCO finetuning (BLIP-2 base). For this experiment (as in the last row of Table 2 of the main paper), we use pretrained cross-attention layers of BLIP-2 as our multimodal combined representation, and finetune them on WebVid-CoVR. In Table A.6, we observe that the BLIP-2 model fine-tuned with COCO has a similar performance to BLIP-2 Base, with a slight improvement on R@1.

Effect on CIR benchmarks. As shown in Table 2 of the main paper, the BLIP-2 backbone with cross-attention layer finetuning achieves the highest performance on the WebVid-CoVR-Test dataset. In Table A.7, we extend the comparison to additional CIR benchmarks, evaluating frozen CLIP with average fusion (no training), CLIP with MLP fusion, and BLIP and BLIP-2 with cross-attention layer finetuning. The results show that MLP fusion fails to generalize effectively across these datasets when trained with WebVid-CoVR, while the BLIP-2 backbone continues to deliver the best results across all benchmarks.

D.3 Incorporating visual similarity between videos

When constructing the WebVid-CoVR dataset, we rely solely on caption similarity to identify similar video pairs for generating triplets, as discussed in Section 3.1. Here we provide an additional analysis on the effect of incorporating visual similarity in a similar fashion as we did with the caption similarity.

TABLE A.7
CLIP vs BLIP performance on CIR benchmarks: The table compares frozen CLIP with average fusion, CLIP with MLP fusion, and BLIP and BLIP-2 with cross-attention layer finetuning. Results demonstrate that while MLP fusion struggles to generalize across datasets when trained with WebVid-CoVR, the BLIP-2 backbone consistently outperforms CLIP across all benchmarks.

Backbone	Pretraining Data	CIRR R@1	FashionIQ R@10	CIRCO mAP@5
CLIP	-	10.65	17.63	3.63
CLIP + MLP	WV-CC-CoVIR	11.39	19.10	5.67
BLIP	WV-CC-CoVIR	38.48	27.70	21.43
BLIP-2	WV-CC-CoVIR	**43.74**	**38.15**	**28.29**

TABLE A.8
Effect of visual similarity: We observe worse performance on CoIR zero-shot benchmarks as we increase the visual similarity threshold in our training data. We train each model for the same number of iterations.

Threshold	Data (%)	CIRR R@1	FashionIQ R@10 mean
0.00 (None)	100%	**41.30**	**37.01**
0.55	92%	40.72	36.13
0.65	71%	38.07	35.71
0.70	55%	35.78	35.03

Specifically, we train variants of our model where we filter the WebVid-CoVR training triplets to only keep those whose video pair similarity is above a certain threshold. We measure the similarity using CLIP features on the middle frames, as in Section 3.1. The results in Table A.8 demonstrate that increasing the visual similarity threshold consistently decreases the downstream performance, while also discarding a large portion of the training data.

This suggests that relying solely on caption similarity is reasonable, and that incorporating visual similarity more strictly can be detrimental. A potential reason is that enforcing visual similarity constraints could bias the model to ignore the input text modification. Overall, our results indicate that the automatically mined video pairs with caption similarity already exhibit sufficient visual consistency for training the CoVR task.

D.4 Dynamic vs static content

The videos in our WebVid-CoVR dataset contain both dynamic sequences with motion, as well as static content. To analyze the prevalence, we compute the optical flow using the Gunnar Farneback's algorithm [77] and empirically choose a magnitude threshold of 1 to distinguish between videos with static and dynamic elements. The magnitude value is obtained by averaging the Euclidean norms of motion vectors in both horizontal and vertical directions across the computed video frames. We identify that around 25% of the triplets contain static target videos, which represents approximately 21% of the overall target videos.

As an additional method to distinguish between static and dynamic triplets, we analyze the part of speech of the word that changes in the target video caption. Specifically, we examine whether the modified word is a noun, verb, or another part of speech. We find that 65% of the changes

TABLE A.9
Performance on WebVid-CoVR-Test when training on dynamic vs static video triplets: We employ two methods to classify videos as static or dynamic: one based on the optical flow of the target video, and another based on the type of word (noun or verb) modified in the target caption. The results indicate that training on the full dataset, which includes both videos with temporal information and not, yields the highest performance.

	Data (%)	R@1	R@5	R@10	R@50
Static target videos	25%	54.77	79.85	87.75	97.50
Dynamic target videos	75%	58.80	83.10	90.61	**98.24**
Nouns change	65%	58.06	82.43	89.98	97.97
Verbs change	9%	53.44	77.82	85.64	96.99
All	100%	**59.82**	**83.84**	**91.28**	**98.24**

involved nouns, 9% involve verbs, and the remaining 26% correspond to other parts of speech (such as adjectives).

We train models omitting either only the static portion or only the dynamic portions. The results in Table A.9 show that training on both dynamic and static triplets is beneficial, with a minor decrease in performance if we omit static videos during training (while maintaining the same iteration count). This may be because image training data can still be complementary to video training [12].

Overall, we demonstrate that WebVid-CoVR contains both static and dynamic videos, hence posing an advantage over image datasets by providing more diversity.

Ideally, leveraging dynamic videos would imply capturing not just static states but also the transitions between them, which is a key distinction between composed video retrieval and composed image retrieval. For instance, when given an image of 'wheat flour' and a text query like 'change this to dough,' an image search will likely retrieve the end result, such as 'dough.' In contrast, composed video retrieval could potentially capture the entire process, showing 'how' the transformation from 'flour' to 'dough' occurs, providing a richer and more informative result. We leave this potential application to future research, which will benefit from better datasets and improved modeling techniques.

D.5 Effect of modification text length

We analyze the impact of the modification text length, by experimenting with generating multiple candidates per caption pair and selecting the longest modification text, increasing the average length from 23.36 to 33.35 characters. However, as shown in Table A.10, this decreases performance on all three datasets (WebVid-CoVR-Test, CIRR, and FashionIQ). We hypothesize that since our triplets represent a single modification, longer texts tend to be more verbose without improving quality.

This shows that while the generated modification texts are not as long on average as the ones from CIRR, our generated texts still provide useful training signal, as evidenced by the state-of-the-art performance of models trained on our dataset when transferred to standard CoIR tasks (CIRR and FashionIQ in Table 4 and CIRCO in Table 5). Moreover, the average number of characters of the modification text in WebVid-CoVR and the widely used FashionIQ CoIR benchmark are comparable (see Table 1 of the main paper: 23.36 vs 27.13, respectively).

TABLE A.10
Increasing the average modification text length in WebVid-CoVR by selecting the longest of multiple generated candidates per caption pair degrades downstream performances on WebVid-CoVR-Test, CIRR, and FashionIQ.

Modification text avg #chars	WebVid-CoVR-Test R@1	WebVid-CoVR-Test MeanR	CIRR R@1	CIRR MeanR	FashionIQ R@10	FashionIQ R@50
(ours) 23.36	**59.82**	**83.30**	**41.42**	**73.22**	**36.81**	**56.70**
(longest) 33.35	58.84	82.72	35.64	67.60	33.20	52.13

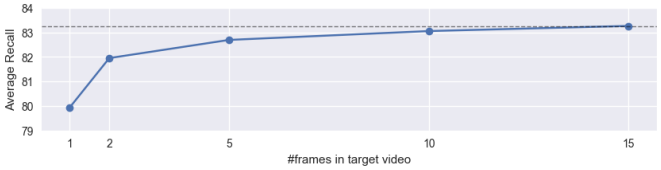

Fig. A.10. **Training with more target video frames** consistently improves performance on WebVid-CoVR-Test.

D.6 Optimal number of frames

In the main paper, we sample 15 frames from WebVid-CoVR videos during training. Here, we experiment with incrementally increasing the number of frames for training and testing. We report the average recall on WebVid-CoVR-Test and observe a steady increase in performance with more frames (see Figure A.10). However, it's worth noting that the performance is already quite high with just one frame, likely due to the static image bias inherent in the WebVid dataset, where videos are typically short in duration.

E Qualitative Analysis

In this section, we provide examples of caption filtering (Section E.1), qualitative comparison between different MTG approaches (Section E.2), qualitative examples of our WebVid-CoVR triplets (Section E.3), samples from our manual test set annotation process (Section E.4), qualitative CoVR results on WebVid-CoVR-Test (Section E.5) and CoIR results on CIRR (Section E.6).

E.1 Examples of filtered captions

As described in Section 3.1 of the main paper, we employ a filtering process to select paired captions that facilitate the generation of meaningful training data. In this section, we provide examples of the filtered captions.

Filtering template captions. Upon analyzing the paired captions, we observed that a significant portion of the pairs originated from a small set of template captions. Out of 1.2M distinct caption pairs, approximately 719k (60%) were generated from these template captions. The following examples showcase some of these template captions:

- **Abstract:** *Abstract color movement tunnel, Abstract color nature background, Abstract color smoke flowing on white background, Abstract colorful paint ink spread explode, Abstract colorful pattern background, Abstract colorful red cement wall background or texture. the camera moves up, Abstract colorful satin background animation, Abstract colorful shiny bokeh background., Abstract colorful smoke on black background*, etc

- **Background:** *Abstract background, Animated backgrounds, Animation, background., Aquarium background, Artistic background, Aurora background, Balloons background, Basketballs background, Beach background, Bluebell background, Bright background, Brush background, Bubbles background, Bubbly background, Celebrate background, Celebratory background, Cg background, Christmas background, Christmas background, Circles background, Color background, Colored background, Colorfull background,*, etc.
- **Concept:** *Brazil high resolution default concept, Brazil high resolution dollars concept, Businessman with advertising hologram concept, Businessman with algorithm hologram concept, Businessman with automation hologram concept, Businessman with bitcoin hologram concept, Businessman with branding hologram concept, Businessman with public relations hologram concept, Close up of an eye focusing on a freelance concept on a futuristic screen., Coins fall into piggy bank painted with flag of ghana. national banking system or savings related conceptual 3d animation, Communication concept, Communication network concept., Communication team concept, Concept of connection, Concept of dancing at disco party. having fun with friends., Concept of education, Concept of geography, Cyber monday concept*, etc
- **Flag:** *Flag of america, Flag of andorra, Flag of aruba, Flag of austria, Flag of azerbaijan, Flag of bahrain, Flag of belarus, Flag of belize, Flag of black, Flag of bolivia, Flag of brazil, Flag of bulgaria, Flag of cameroon, Flag of canada,* etc.

Filtering caption pairs with high or low similarity. To ensure the generation of meaningful modifications, we further refine the selection of caption pairs by filtering out those with excessively high or low similarity. Caption pairs with highly similar meanings may result in trivial or unnoticeable modifications. Conversely, pairs with significant dissimilarity can lead to large visual differences that are difficult to describe accurately. We show below some of the filtered captions based on the CLIP text embedding cosine similarity.

- **High similarity:** 10% of the pairs have CLIP text similarity above 0.96.
  - Close-up of a tree with green leaves and sunlight
  - Close-up of a tree with green leaves and sunshine
  - Businessman speaking on the phone
  - Businessman talking on the phone
  - Boat on a sea
  - Boat on the sea
- **Low similarity:** 2% of the pairs have CLIP text similarity below 0.60.
  - Leaves close-up
  - Peacock, close-up
  - Moon jellyfish
  - Moon night
  - Close up of a lynx
  - Close up of a milkshake

Exclusion of digit differences and out-of-vocabulary words. In order to maintain the high quality and coherence of the generated modification text, we apply additional filtering criteria. Specifically, we exclude caption pairs where the differences between captions are numerical digits (often representing dates) or involve out-of-vocabulary words (using the python libraries wordfreq and enchant) that may hinder the generation process.

- **Difference between the captions is a digit:** Approximately 2% of the pairs.
  - 23.09.2015 navigation on the moscow river
  - 07.08.2015 navigation on the moscow river.
  - Light leaks element 190
  - Light leaks element 215
  - Pure silver, shape of granules of pure silver each one is unique 44 (2)
  - Pure silver, shape of granules of pure silver each one is unique 95 (2)
- **Difference in one of the captions has an out-of-vocabulary word:** Approximately 7% of the pairs.
  - Businessman writing on hologram desk tech word- bitcoin
  - Businessman writing on hologram desk tech word- crm
  - Mitomycin-c - male doctor with mobile phone opens and touches hologram active ingredient of medicine
  - Oxazepam - male doctor with mobile phone opens and touches hologram active ingredient of medicine
  - Blue forget-me-nots
  - Blue galaxy

E.2 Qualitative comparison of MTG approaches

In Section 4.6 of the main paper and Section C.5, we show that finetuning our MTG-LLM works better than a rule-based approach and than few-shot prompting of the LLM. In this section, we provide a qualitative comparison of three different methods for generating modification text: (i) rule-based, (ii) prompting-based, and (iii) our MTG-LLM finetuning. We present examples of paired captions and the corresponding modification texts generated by each method in Table A.11.

Rule-based method. The rule-based method relies on predefined rules to generate modification text. We illustrate an example limitation in the last row of Table A.11, where the difference text is simply a preposition (i.e., 'of' vs 'above'), and the modification text becomes 'Remove of'. The rule-based method performs well when the modifications follow a specific pattern, but it may struggle with more complex modifications (e.g., 'tree' vs 'trees' should generate 'add more trees' for plurality).

Prompting LLM. The prompting-based method involves using a pretrained language model without finetuning.

TABLE A.11
Comparison between modification text generation approaches: We provide qualitative examples for a pair of captions, and three methods to generate modification text: (i) rule-based, (ii) prompting-based, (iii) our MTG-LLM finetuning. Rule-based method is limited, for example in the case where the difference text is a preposition (last row), whereas the prompting-based method is prone to hallucinating (e.g., 'remove iceberg', 'change the pose of the runner'). Our approach tends to be the most robust across cases.

Caption_1	*Happy* girl dancing
Caption_2	*Beautiful* girl dancing
Rule-based	Make the *Happy* into *Beautiful*
Prompting LLM	Change girl
MTG-LLM	Make her more *beautiful*
Caption_1	Black *bird*
Caption_2	black *bear*
Rule-based	Add *bear*
Prompting LLM	Remove animal
MTG-LLM	Turn it into a *bear*
Caption_1	*Autumn* landscape in the mountains.
Caption_2	*Winter* landscape in the mountains
Rule-based	Remove *Autumn*
Prompting LLM	Change scene to winter.
MTG-LLM	Change to *winter*
Caption_1	Flying over the *ice*
Caption_2	Flying over the *mountains*
Rule-based	Take out *ice* and add *mountains*
Prompting LLM	Remove iceberg
MTG-LLM	Move the *ice* to the *mountains*
Caption_1	*Dandelion* field
Caption_2	*Rice* field
Rule-based	Change *dandelion* for *rice*
Prompting LLM	Change the season
MTG-LLM	Turn it into a *rice* field.
Caption_1	*Happy* woman
Caption_2	*Running* woman
Rule-based	Change it to *running*
Prompting LLM	Remove the girl and change the pose of the runner
MTG-LLM	Make her a runner
Caption_1	Woman working on cellphone at *night*
Caption_2	Woman working on cellphone at *outdoor*
Rule-based	Replace *night* with *outdoor*
Prompting LLM	Change the woman and add one cellphone
MTG-LLM	Make it an *outdoor*
Caption_1	Barber cuts the hair of the client with *clipper*.
Caption_2	Barber cuts the hair of the client with *trimmer*
Rule-based	Take out *clipper* and add *trimmer*
Prompting LLM	Add or change the hairdresser's tool
MTG-LLM	Change to a *trimmer*
Caption_1	Aerial shot *of* a lake.
Caption_2	Aerial shot *above* a lake
Rule-based	Remove *of*
Prompting LLM	Add a lake to the sequence
MTG-LLM	make it an aerial shot *of* a lake

However, this method is prone to hallucinations and may generate modification text that does not accurately represent the intended difference. For example, in the second example, the prompting LLM suggests removing the term 'animal' instead of replacing 'bird' with 'bear'.

MTG-LLM (Our approach). Our MTG-LLM approach utilizes a large language model finetuned on a manually annotated dataset specifically for modification text generation. It tends to be the most robust across different cases.

E.3 Training triplet examples

Figures A.11, A.12, and A.13 all show examples of triplets generated using our automatic dataset creation. These examples demonstrate the effectiveness of our approach in generating coherent modification texts for paired videos. This capability serves as a form of data augmentation and increasing the diversity in the training set. In Figure A.14, we show that the dataset is not composed by pairs only, as there are many captions that have many relations between them. Furthermore, in Figure A.15 we show cases where a single caption is associated with multiple videos. This scenario allows us to generate multiple triplets by leveraging the diverse visual content captured in different videos. The triplets shown in the aforemention figures exhibit a wide range of variations, encompassing different themes such as emotions, food, actions, camera edits, gender changes, and time of the day.

E.4 Manual test set annotation

In this section, we further describe the process of manually annotating the test set for our WebVid-CoVR-Test CoVR benchmark, previously discussed in Section 3.2 of the main paper. The annotation process involves presenting the annotator with generated modification texts from three different runs of MTG-LLM, along with three frames each from the query and target videos. The annotator's task is to evaluate the quality of the modification texts and the suitability of the videos for the CoVR task.

A total of 3.1k triplets were shown for annotation. In Figure A.16 and Figure A.17, we present 10 examples that were considered correct during the annotation, along with the chosen modification texts (marked with a checkmark). These examples demonstrate successful modification texts and appropriate video content for the CoVR task.

On the other hand, in Figure A.17, we show 8 examples that were discarded during the annotation. These examples were rejected either because the modification texts were incorrect or because the videos were deemed unsuitable for the CoVR task due to being either too similar (e.g., bottom left, both videos are showing the same coffe with almost no modification) or too incoherent (e.g., top right example "Make the water a river").

E.5 Qualitative CoVR results on WebVid-CoVR-Test

In Figure A.18, we show qualitative CoVR results on our manually verified WebVid-CoVR-Test set. We observe that top ranked video frames have high visual and semantic similarity with the queries even when not corresponding to the ground truth (marked with a green border).

E.6 Qualitative CoIR results on the CIRR benchmark

In Figure A.19, we demonstrate qualitative CoIR results of our models trained only on WebVid-CoVR (ZS) and the one further finetuned on CIRR training set (Sup.), tested on the CIRR test set. We observe promising retrieval quality for both models.

Fig. A.11. **Examples of generated triplets:** We illustrate triplet samples (one per row) generated using our automatic dataset creation methodology. Each sample consists of two videos with their corresponding captions (at the bottom of each video) and the generated modification text using our MTG-LLM (in purple).

Fig. A.12. **Examples of generated triplets (ctd)**

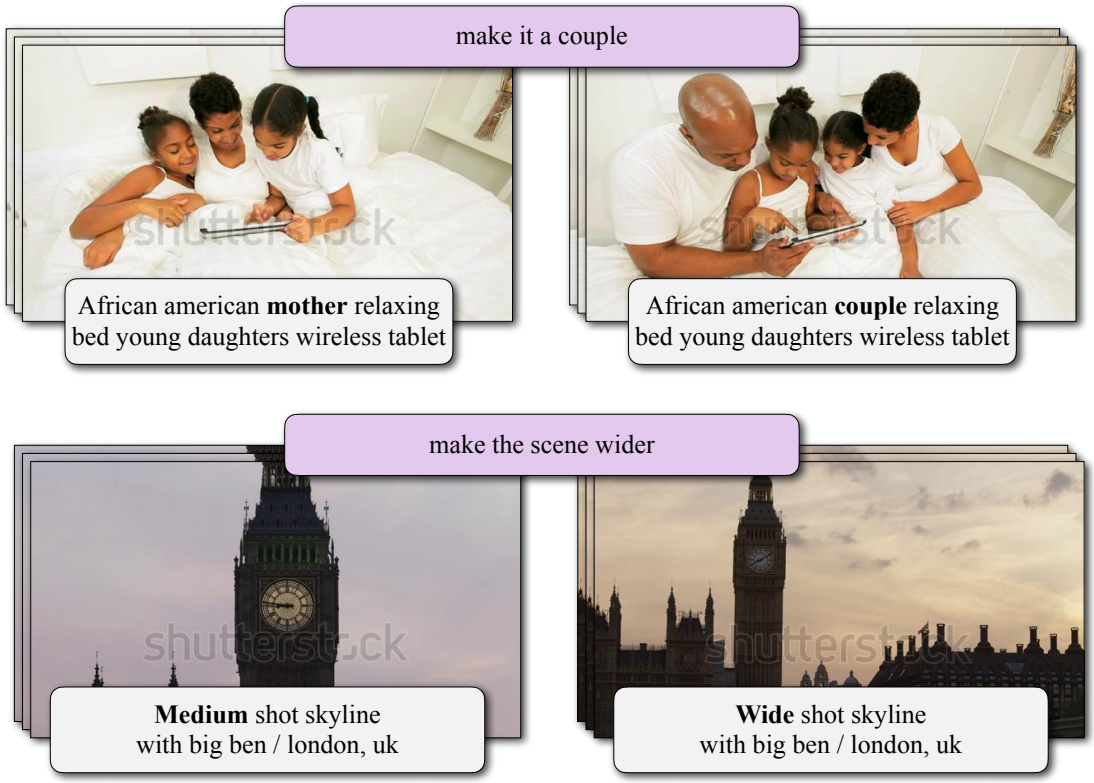

Fig. A.13. **Examples of generated triplets (ctd)**

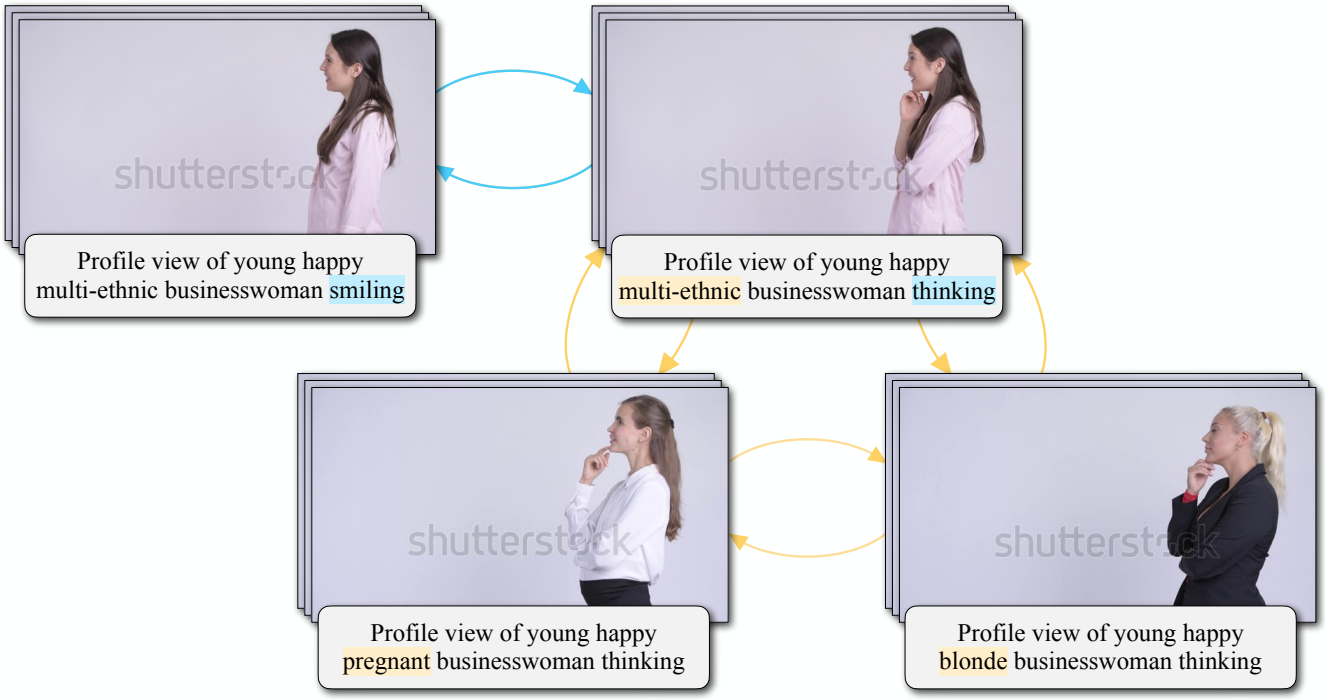

Fig. A.14. **Generated triplets from multiple similar captions:** We can train with as many triplets as pairs of captions with one word difference by generating modification texts using our trained MTG-LLM: *she is thinking*, *Have her look happy*, *Make the businesswoman pregnant*, *make her blonde*, *make her multi-ethnic*, *Make the woman pregnant*, etc.

Fig. A.15. **Generated triplets with multiple videos:** In cases where there are several videos with the same caption, we can generate multiple triplets by leveraging the multiple videos. It can be seen as a way of data augmentation.

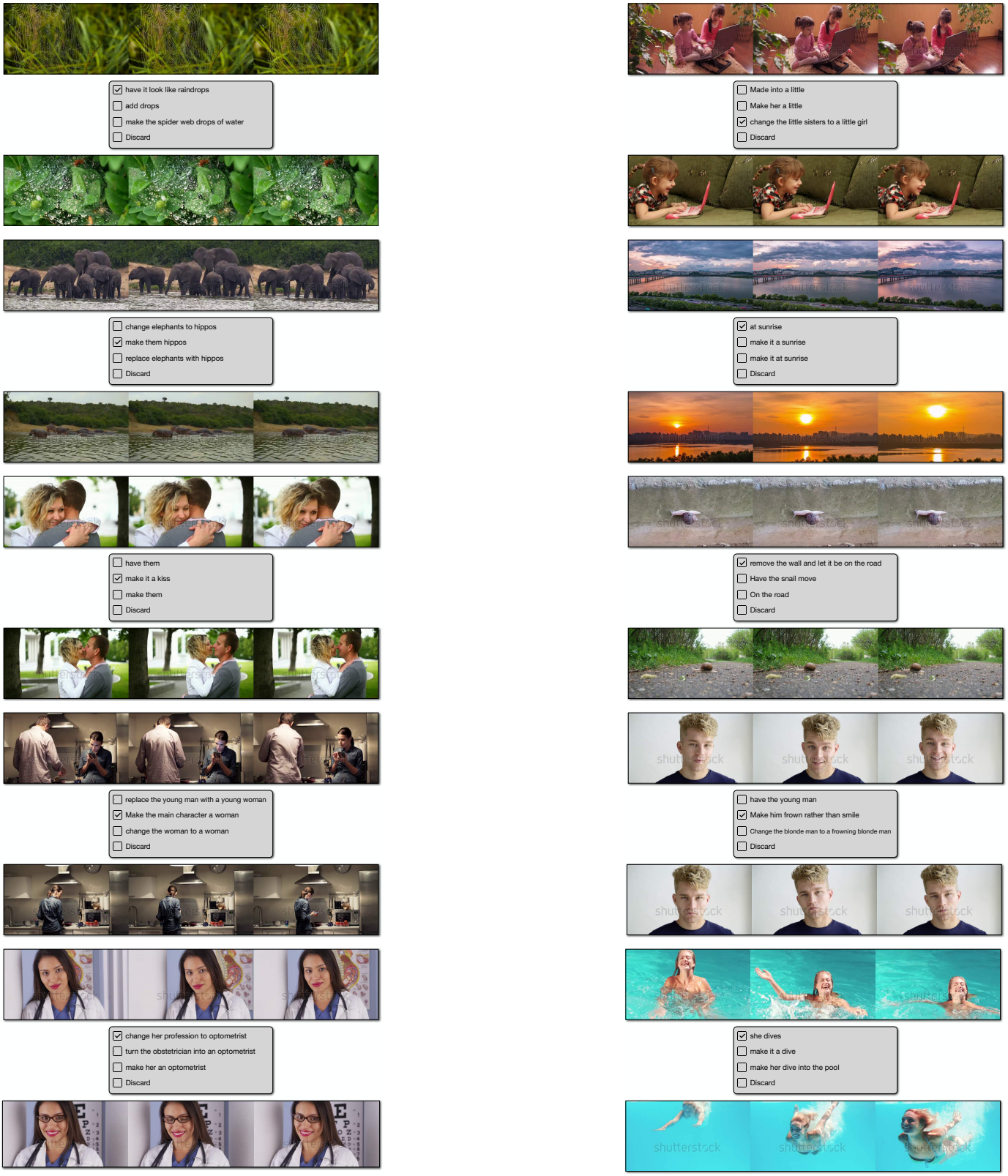

Fig. A.16. **Manual annotation examples (kept):** We show samples from WebVid-CoVR-Test which are automatically mined triplets that are marked as correct during the annotation process. Each sample consists of two videos and a set of modification text options (in between each video pair). The chosen modification text is indicated by a checkmark.

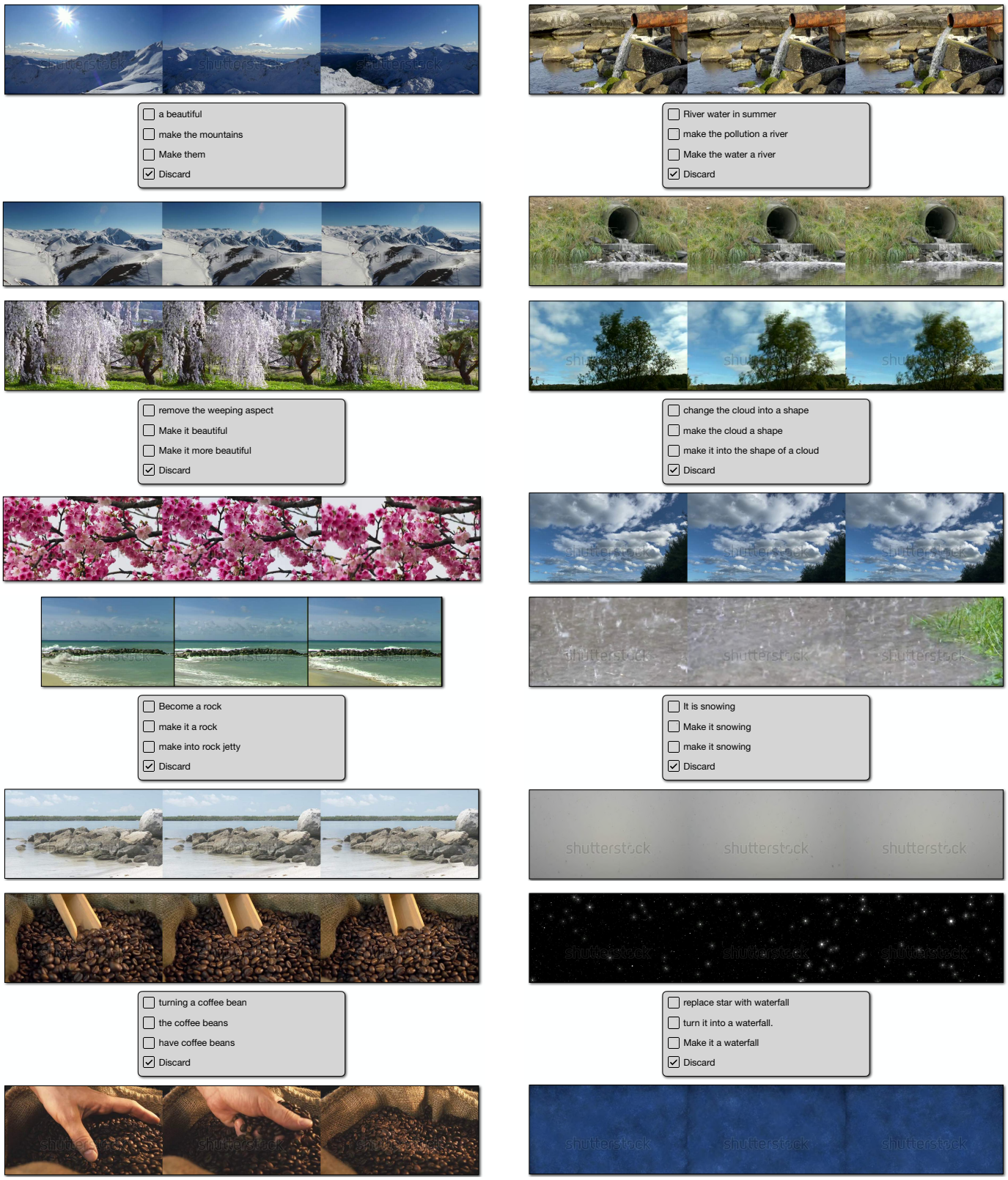

Fig. A.17. **Manual annotation examples (discarded):** We show automatically mined triplets that are discarded during the annotation process. Discarded texts include videos that are too similar (bottom left), too dissimilar (bottom right), or have bad modification texts (top left).

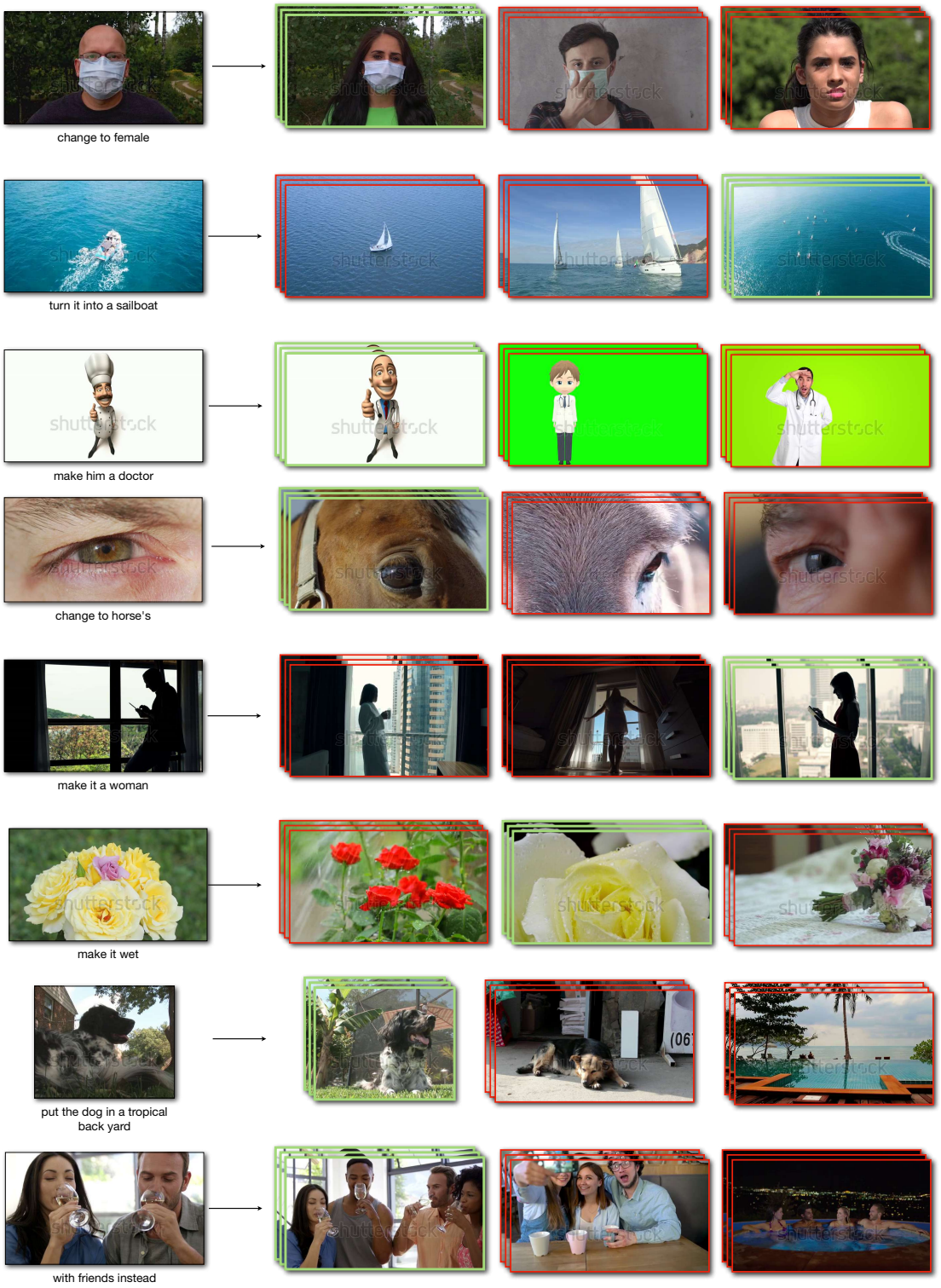

Fig. A.18. **Qualitative CoVR results on WebVid-CoVR-Test:** We display the input image and modification text queries on the left, along with the top 3 retrieved videos by our model on the right. Ground-truth is denoted with a green border.

Fig. A.19. **Qualitative CoIR results on CIRR test set:** Given a query image and a modification text, we show our top retrieved videos of our zero-shot (ZS) model trained with WebVid-CoVR and the model finetuned on CIRR ground-truth supervision (Sup.).