CoVR: Learning Composed Video Retrieval from Web Video Captions

Lucas Ventura, Antoine Yang, Cordelia Schmid, Gül Varol

▶ **To cite this version:**

Lucas Ventura, Antoine Yang, Cordelia Schmid, Gül Varol. CoVR: Learning Composed Video Retrieval from Web Video Captions. 2023. hal-04327307

HAL Id: hal-04327307
https://hal.science/hal-04327307

Preprint submitted on 6 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

arXiv:2308.14746v1 [cs.CV] 28 Aug 2023

CoVR: Learning Composed Video Retrieval from Web Video Captions

Lucas Ventura[1,2] **Antoine Yang**[2] **Cordelia Schmid**[2] **Gül Varol**[1]
[1] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, France
[2] Inria, ENS, CNRS, PSL Research University, France
https://imagine.enpc.fr/~ventural/covr
lucas.ventura@enpc.fr

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. Our experiments further demonstrate that training a CoVR model on our dataset effectively transfers to CoIR, leading to improved state-of-the-art performance in the zero-shot setup on both the CIRR and FashionIQ benchmarks. Our code, datasets, and models are publicly available at imagine.enpc.fr/~ventural/covr.

Figure 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 people, with fireworks".

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

Preprint. Under review.

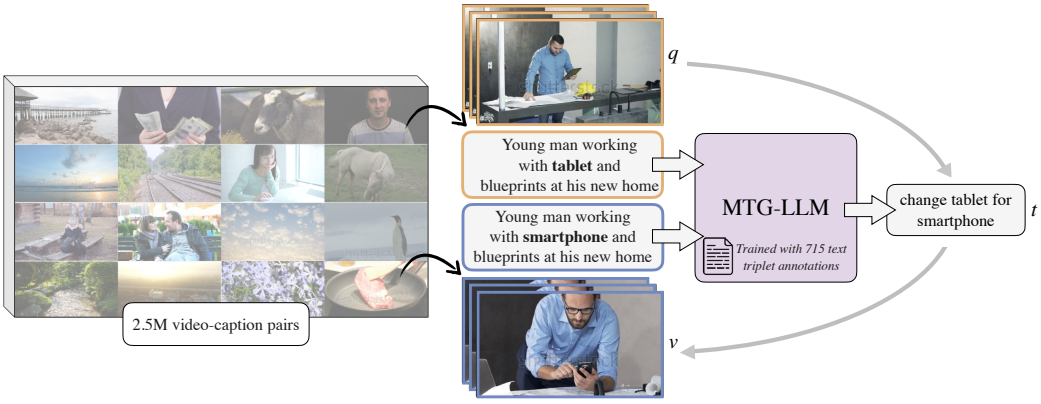

Figure 2: **Method overview:** We automatically mine similar caption pairs from a large video-caption database from the Web, and use our modification text generation language model (MTG-LLM) to describe the difference between the two captions. MTG-LLM is trained on a dataset of 715 triplet text annotations [8]. 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.

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 [34, 52], CoIR has emerged as a new task [66], and since then witnessed improvements of both models and benchmarks [6, 7, 21, 31, 41, 67]. 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) [63] 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).

We apply our triplet generation approach to the WebVid2M dataset [4] which contains 2.5M Web-scraped video-caption pairs. This results in the WebVid-CoVR training dataset with 1.6M CoVR triplets. By virtue of its automatic generation procedure, WebVid-CoVR is inherently noisy. To efficiently train on such large-scale and noisy data, we use a contrastive loss [65], adopting the HN-NCE variant from [51] to upsample the significance of hard negatives. We design a CoVR model based on the cross-modal BLIP [34] and use query scoring [5] 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 [41] and FashionIQ [67] benchmarks in the zero-shot setup. Finally, to foster research in CoVR, we repeat our generation procedure on a distinct subset of the WebVid10M dataset [4] 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. We apply this pipeline to the WebVid2M dataset and generate the WebVid-CoVR training dataset with 1.6M CoVR triplets. (ii) We show that training a CoVR model on WebVid-CoVR transfers well to the CoIR task, and achieves state-of-the-art

Table 1: **Existing datasets:** We compare our proposed WebVid-CoVR training dataset and its manually annotated test set WebVid-CoVR-Test with existing composed visual retrieval datasets. 📷 denotes image, 🎥 denotes video datasets. We contribute the largest training dataset 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 3).

Dataset	Type	#Triplets	#Visuals	#Unique words	Avg. text length	Domain
CIRR [41]	📷	36,554	21,185	7,129	59.51	Natural
FashionIQ [67]	📷	30,132	7,988	4,425	27.13	Fashion
CIRCO [6]	📷	1,020	-	-	-	Natural
LaSCo [31]	📷	389,305	121,479	13,488	30.70	Natural
SynthTriplets18M [21]	📷	18,000,000	-	-	-	Synthetic
WebVid-CoVR	🎥	1,648,789	130,775	19,163	23.36	Natural
WebVid-CoVR-Test	🎥	2,556	2,444	1,935	21.97	Natural

results on the CIRR and FashionIQ benchmarks 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 imagine.enpc.fr/~ventural/covr.

2 Related Work

Composed image retrieval (CoIR). CoIR [66] has been an active area of research in recent years [6, 7, 13, 21, 22, 27, 41, 54, 66, 67]. Most methods designed for this problem use manually annotated image-text-image triplets for training [7, 13, 41, 67]. Very recent works, such as Pic2Word [54] and SEARLE [6], explore zero-shot CoIR setups where no manually annotated CoIR triplet is used. These approaches build on CLIP [52] 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 CoIR.

Datasets for composed image retrieval. CIRR [41] and Fashion-IQ [67] are the two most widely used CoIR benchmarks. Both 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 concurrent works proposed larger, automatically generated CoIR datasets: LaSCo [31] and SynthTriplets18M [21]. However, these two datasets are currently not publicly available. The LaSCo dataset [31] is generated using the visual question answering annotations and the pairing between images and counterfactual images in the VQAv2 dataset [3]. 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 [21] uses the text-conditioned image editing framework InstructPix2Pix [8] 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., 38% vs 19% zero-shot R@1 on CIRR while generating fewer data). Lastly, our WebVid-CoVR dataset is not limited to still images and considers videos, while standing out as the largest composed retrieval dataset 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 [2, 12, 25, 30, 33, 35, 37, 44, 52, 55, 59, 75, 79] or video-caption pairs [1, 32, 36, 47, 48, 61, 68, 69, 76, 77, 78]. 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 [49], visual question answering [71] and visual dialog [39]. 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 [17, 18, 19, 40, 45, 46, 53, 68, 70, 72, 73]. We also make use of multiple video frames with query scoring similar to [5]. 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 CoVR 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.

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 [4] with 2.5M video-caption pairs, but this methodology could potentially be applied to other large datasets of video-text (or image-text) pairs.

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`zipf` frequency [58]. 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 12K distinct captions, resulting in 1.2M paired videos that we will use to generate the modification text.

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 B.4 of the Appendix. We use top-k sampling [16] for generating the tokens instead of maximum-likelihood-based 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 [63]. We then finetune this LLM for the MTG task on a manually annotated text dataset. In particular, we repurpose the editing dataset from InstructPix2Pix [8], which provides a modification text and a target caption for

[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 the supplementary material (Section C.2) with video queries.

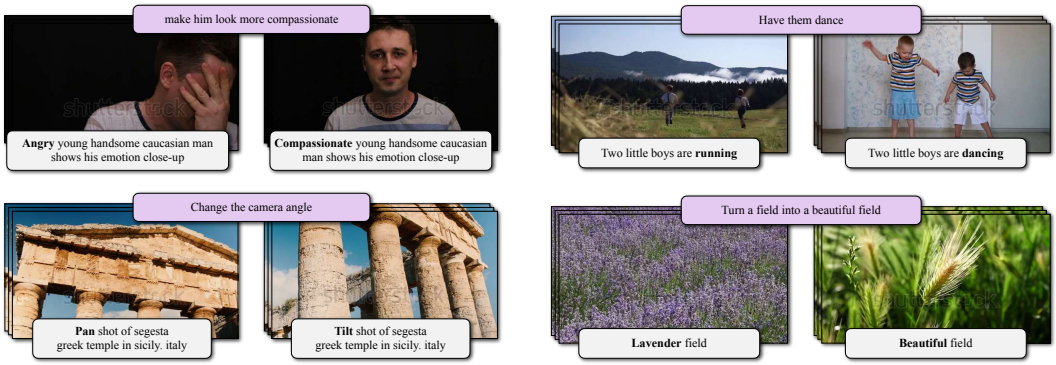

Figure 3: **Examples of generated CoVR triplets in WebVid-CoVR:** The middle frame of each video is shown with its corresponding caption, with the distinct word highlighted in bold. Additionally, the generated modification text is displayed on top of each pair of videos. The bottom right example illustrates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

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 B.4 of the Appendix.

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 [4], 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. 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 D.3 of the Appendix) 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 right 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 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 [4] 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.

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 model architecture. Our model architecture builds upon a pretrained image-text model, BLIP [34]. The BLIP model 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 is not pretrained for composed visual retrieval with both visual and text inputs. Therefore we adapt BLIP to the CoIR/CoVR task as follows.

We use the BLIP 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 image-grounded text encoder together, which outputs a multi-modal embedding $f(q,t) \in \mathbb{R}^d$ where d is the embedding dimension. 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 image embeddings to obtain thevideo embedding vector $h(v_k) \in \mathbb{R}^d$. The weights are obtained by computing the similarity between the corresponding frame and the modification text using the pretrained BLIP image and text encoders, respectively (similar to [5] in the context of text-to-video retrieval). Using pretrained and frozen BLIP embeddings allows us to precompute and store all the weights. Finally, 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 [51, 65], as it has been shown to be effective to learn strong multi-modal representations from large-scale noisy data [52]. We make several design choices to maximize its efficiency. 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 4.4 (Table 7).

Second, following HN-NCE [51], we use as negatives all target videos $v_{j \in \mathcal{B}}$ in the batch $\mathcal{B}$ and additionally increase the weight of most similar samples. Formally, given a training batch $\mathcal{B}$ of triplets (q_i, t_i, v_i), we minimize the following loss:

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

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_i and the target video embedding $\hat{v}_j$, and $w_{i,j}$ is set as in [51] with $\beta = 0.5$.

4 Experiments

In this section, 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), as well as transfer results of CoIR on standard image benchmarks (Section 4.3). Finally, we provide ablations on our key components (Section 4.4).

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.

CIRR [41] is a manually annotated CoIR dataset that contains open-domain natural images from NLVR2 [60]. It contains 36.5K queries annotated on 19K different images. CIRR includes two benchmarks: a standard one with the target search space as the entire validation corpus, 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 [67] 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

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 [7]. Moreover, using multiple frames to embed the target video brings further improvements over using the middle frame. The first row represents the baseline for random performance.

	Input modalities	Fusion	Backbone	#frames	R@1	R@5	R@10	R@50
Random	-	-	-	-	0.08	0.23	0.35	1.76
Not finetuned on WebVid-CoVR	Text	-	BLIP	-	19.68	37.09	45.85	65.14
	Visual	-	BLIP	15	34.90	59.23	68.04	85.95
	Visual + Text	Avg	CLIP	15	44.37	69.13	77.62	93.00
	Visual + Text	Avg	BLIP	15	45.46	70.46	79.54	93.27
Finetuned on WebVid-CoVR	Text	-	BLIP	-	23.67	45.89	55.13	77.03
	Visual	-	BLIP	15	38.89	64.98	74.02	92.06
	Visual + Text	MLP	CLIP	1	50.55	77.11	85.05	96.60
	Visual + Text	MLP	BLIP	1	50.63	74.80	83.37	95.54
	Visual + Text	CA	BLIP	1	51.80	78.29	85.84	97.07
	Visual + Text	CA	BLIP	15	**53.13**	**79.93**	**86.85**	**97.69**

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.

Evaluation metrics. Following standard evaluation protocols [41], 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 [63] that we finetune for one epoch with an initial learning rate of $3e{-}5$ for MTG. For our CoVR model, we use the BLIP with ViT-L [15] at 384 pixels finetuned for text-image retrieval on COCO and freeze the ViT for computational efficiency. We train our CoVR model on WebVid-CoVR for 4 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 B 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 [34] or CLIP [52] 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 [7] that concatenates the image and text embeddings from CLIP (or BLIP in [31]), and is referred to as late fusion by CASE [31]. 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 only the middle frame embedding or 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).

Table 3: **State-of-the-art comparison on the CIRR test set:** Our model benefits from training on WebVid-CoVR in the zero-shot setting, and in the finetuning setting where it performs competitively. † denotes results reported by [41]. For methods that pretrain specifically for composed retrieval, we indicate their pretraining data. CC3M denotes Conceptual Captions 3M [56].

Mode	Method	Pretraining Data	Recall@K				R_{subset}@K		
			K=1	K=5	K=10	K=50	K=1	K=2	K=3
Train (CIRR)	TIRG [66]†	-	14.61	48.37	64.08	90.03	22.67	44.97	65.14
	TIRG+LastConv [66]†	-	11.04	35.68	51.27	83.29	23.82	45.65	64.55
	MAAF [14]†	-	10.31	33.03	48.30	80.06	21.05	41.81	61.60
	MAAF-BERT [14]†	-	10.12	33.10	48.01	80.57	22.04	42.41	62.14
	MAAF-IT [14]†	-	9.90	32.86	48.83	80.27	21.17	42.04	60.91
	MAAF-RP [14]†	-	10.22	33.32	48.68	81.84	21.41	42.17	61.60
	ARTEMIS [13]	-	16.96	46.10	61.31	87.73	39.99	62.20	75.67
	CIRPLANT [41]†	-	19.55	52.55	68.39	92.38	39.20	63.03	79.49
	LF-BLIP [7, 31]	-	20.89	48.07	61.16	83.71	50.22	73.16	86.82
	CompoDiff [21]	SynthTriplets18M [21]	22.35	54.36	73.41	91.77	35.84	56.11	76.60
	Combiner [7]	-	33.59	65.35	77.35	95.21	62.39	81.81	92.02
	CASE [31]	-	48.00	79.11	87.25	**97.57**	75.88	**90.58**	**96.00**
	CASE [31]	LaSCo [31]	48.68	79.98	88.51	97.49	76.39	90.12	95.86
	CASE [31]	LaSCo [31]+COCO [38]	49.35	**80.02**	**88.75**	97.47	**76.48**	90.37	95.71
	CoVR-BLIP	-	48.84	78.05	86.10	94.19	75.78	88.22	92.80
	CoVR-BLIP	WebVid-CoVR	**49.69**	78.60	86.77	94.31	75.01	88.12	93.16
Zero Shot	Random†	-	0.04	0.22	0.44	2.18	16.67	33.33	50.00
	CompoDiff [21]	SynthTriplets18M [21]	19.37	53.81	72.02	90.85	28.96	49.21	67.03
	Pic2Word [54]	CC3M [56]	23.90	51.70	65.30	87.80	-	-	-
	CASE [31]	LaSCo [31]	30.89	60.75	73.88	92.84	60.17	80.17	90.41
	CASE [31]	LaSCo [31]+COCO [38]	35.40	65.78	**78.53**	**94.63**	64.29	82.66	**91.61**
	CoVR-BLIP	-	19.76	41.23	50.89	71.64	63.04	81.01	89.37
	CoVR-BLIP	WebVid-CoVR	**38.48**	**66.70**	77.25	91.47	**69.28**	**83.76**	91.11

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 cross-attention (CA) performs better than the MLP fusion. (v) Results with the BLIP backbone are marginally higher than those with CLIP. (vi) Using $N = 15$ target video frames further boosts the performance over using only the middle frame.

4.3 Transfer learning to composed image retrieval

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. We perform zero-shot CoIR by directly applying our model trained on our automatically generated WebVid-CoVR dataset to CoIR tasks and also explore finetuning our model on the training set of the downstream benchmark.

Tables 3 and 4 report results on CIRR and Fashion-IQ datasets, respectively. These results show that our model highly benefits from training on WebVid-CoVR, especially in the zero-shot setting, on both datasets. In addition, our model achieves state-of-the-art zero-shot performance on both CIRR and FashionIQ, and performs competitively in the finetuning setting on both benchmarks.

4.4 Ablation studies

In this section, we ablate the importance of several key aspects of our method by evaluating the performance of the model trained only on WebVid-CoVR.

Importance of data scale. In Table 5, 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 generated data underperforms.

Table 4: **State-of-the-art comparison on the FashionIQ validation set:** Our model benefits from training on WebVid-CoVR in the zero-shot setting, and in the finetuning setting. For methods that pretrain specifically for composed retrieval, we indicate their pretraining data. ‡ denotes results reported by [57].

Mode	Method	Pretraining Data	Dress R@10	Dress R@50	Shirt R@10	Shirt R@50	Toptee R@10	Toptee R@50	Average R@10	Average R@50
Train (FashionIQ)	JVSM [10]	-	10.70	25.90	12.00	27.10	13.00	26.90	11.90	26.60
	CIRPLANT [41]	-	17.45	40.41	17.53	38.81	61.64	45.38	18.87	41.53
	TRACE w/BER [24]	-	22.70	44.91	20.80	40.80	24.22	49.80	22.57	46.19
	VAL w/GloVe [11]	-	22.53	44.00	22.38	44.15	27.53	51.68	24.15	46.61
	MAAF [14]	-	23.80	48.60	21.30	44.20	27.90	53.60	24.30	48.80
	CurlingNet [74]‡	-	26.15	53.24	21.45	44.56	30.12	55.23	25.90	51.01
	RTIC-GCN [57]‡	-	29.15	54.04	23.79	47.25	31.61	57.98	28.18	53.09
	CoSMo[29]	-	25.64	50.30	24.90	49.18	29.21	57.46	26.58	52.31
	ARTEMIS[13]	-	27.16	52.40	21.78	43.64	29.20	53.83	26.05	50.29
	DCNet[27]	-	28.95	56.07	23.95	47.30	30.44	58.29	27.78	53.89
	SAC w/BERT[23]	-	26.52	51.01	28.02	51.86	32.70	61.23	29.08	54.70
	FashionVLP[20]	-	32.42	60.29	31.89	58.44	38.51	68.79	34.27	62.51
	LF-CLIP (Combiner) [7]	-	31.63	56.67	36.36	58.00	38.19	62.42	35.39	59.03
	LF-BLIP [7, 31]	-	25.31	44.05	25.39	43.57	26.54	44.48	25.75	43.98
	CASE [31]	LaSCo [31]	**47.44**	**69.36**	**48.48**	**70.23**	50.18	72.24	**48.79**	**70.68**
	CoVR-BLIP	-	43.51	67.94	48.28	66.68	51.53	73.60	47.77	69.41
	CoVR-BLIP	WebVid-CoVR	44.55	69.03	48.43	67.42	**52.60**	**74.31**	48.53	70.25
Zero Shot	Random	-	0.26	1.31	0.16	0.79	0.19	0.95	0.06	0.32
	Pic2Word [54]	CC3M [56]	20.00	40.20	26.20	43.60	27.90	47.40	24.70	43.70
	CoVR-BLIP	-	13.48	31.96	16.68	30.67	17.84	35.68	16.00	32.77
	CoVR-BLIP	WebVid-CoVR	**21.95**	**39.05**	**30.37**	**46.12**	**30.78**	**48.73**	**27.70**	**44.63**

Table 5: **Data size:** We experimentally validate the importance of the number of videos used for data generation and of filtering the generated data, evaluated by downstream performance on WebVid-CoVR-Test (test), CIRR (test), and FashionIQ (val). All models are trained for the same number of iterations on the generated data. Training batches are made up with distinct target videos.

Initial #videos	*Generated* #target videos	#triplets	Filtering	WebVid-CoVR-Test R@1	WebVid-CoVR-Test MeanR	CIRR R@1	CIRR MeanR	FashionIQ R@10	FashionIQ R@50
0	-	-	-	15.85	36.80	19.76	45.88	16.00	32.77
200k	4k	11k	✓	32.90	62.20	35.42	65.25	27.11	**46.96**
500k	14k	66k	✓	48.20	76.12	**38.84**	68.09	**28.05**	45.91
1M	38k	269k	✓	50.94	77.89	38.68	68.20	27.78	45.16
2.5M	130k	1.6M	✓	**53.13**	**79.93**	38.48	**68.48**	27.70	44.63
2.5M	211k	3.6M	✗	52.93	78.92	37.45	67.41	24.50	40.12

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 a simple rule-based baseline that uses several templates to generate the modification text given the two captions that differ by one word. Specifically, 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 B.3 of the Appendix. Additionally, we investigate the possibility of paraphrasing the rule-based modification texts using `GPT-3.5-turbo` from OpenAI [9] as a source of augmentation, by prompting *"Paraphrase the following sentence: `{Rule-base modification text}`"*. In preliminary analysis, we qualitatively observed that LLaMA [63] and LLaMA 2 [64] alternatives were overly verbose when used for paraphrasing; however, GPT-3.5 outputs were satisfactory.

In Table 6, 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 41 to 52 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.6 of the Appendix.

Table 6: **Modification text generation:** We compare our MTG-LLM (LLaMA 7B parameters) against both a rule-based MTG baseline and a paraphrased rule-based MTG baseline (using GPT-3.5-turbo from OpenAI). We observe important gains in the downstream performance of the model trained on the generated data.

	WebVid-CoVR				CIRR			
Model	R@1	R@5	R@10	R@50	R@1	R@5	R@10	R@50
Rule-based	41.39	68.58	77.74	93.74	15.66	38.36	51.69	78.92
Rule-based paraphrased	52.39	78.76	86.38	97.46	33.54	61.78	72.99	88.99
MTG-LLM	**53.13**	**79.93**	**86.85**	**97.69**	**38.48**	**66.70**	**77.25**	**91.47**

Table 7: **Ablations on training strategies:** Constructing batches of distinct target videos (and not CoVR triplets) and up-sampling hard negatives both benefit the downstream CoVR/CoIR performance.

		WebVid-CoVR-Test				CIRR			
Iteration	HN-NCE [51]	R@1	R@5	R@10	R@50	R@1	R@5	R@10	R@50
Triplets	✓	49.80	78.33	**87.09**	97.38	35.35	63.40	73.98	89.64
Videos	✗	49.02	76.06	84.62	96.79	35.57	63.45	74.53	90.72
Videos	✓	**53.13**	**79.93**	86.85	**97.69**	**38.48**	**66.70**	**77.25**	**91.47**

Training strategies. In Table 7, we first show the benefit 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 [51] also slightly benefits the performance.

5 Conclusions, Limitations, and Societal Impacts

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 our dataset may not depict some visible changes due to the way we generate triplets (i.e., without looking at images, but only considering caption pairs). Moreover, our modification text generation model 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 be captured with our approach: "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.

Societal impact. 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 our model on different datasets with harmful content. These risks include but are not limited to: surveillance applications such as searching for a specific person in videos and gathering sensitive information, and looking up violent and graphic videos. For our WebVid-CoVR dataset release, we provide a dataset on our project page, and refer to Section A for further analysis about removal of inappropriate content. We note that anyone utilizing our dataset must also adhere to the terms of use stipulated by WebVid [4].

Acknowledgments and Disclosure of Funding

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] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *NeurIPS*, 2021. 3

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 3

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *ICCV*, 2015. 3

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 2, 4, 5, 10, 15

[5] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. A CLIP-hitchhiker's guide to long video retrieval. *arXiv:2205.08508*, 2022. 2, 3, 6

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

[7] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Effective conditioned and composed image retrieval combining CLIP-based features. In *CVPR*, 2022. 2, 3, 7, 8, 9

[8] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2, 3, 4, 20, 21

[9] Tom Brown et al. Language models are few-shot learners. In *NeurIPS*, 2020. 9

[10] Yanbei Chen and Loris Bazzani. Learning joint visual semantic matching embeddings for language-guided retrieval. In *ECCV*, 2020. 9

[11] Yanbei Chen, Shaogang Gong, and Loris Bazzani. Image search with text feedback by visiolinguistic attention learning. In *CVPR*, 2020. 9

[12] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Universal image-text representation learning. In *ECCV*, 2020. 3

[13] Ginger Delmas, Rafael Sampaio de Rezende, Gabriela Csurka, and Diane Larlus. ARTEMIS: Attention-based retrieval with text-explicit matching and implicit similarity. In *ICLR*, 2022. 3, 8, 9

[14] Eric Dodds, Jack Culpepper, Simao Herdade, Yang Zhang, and Kofi Boakye. Modality-agnostic attention fusion for visual search with text feedback. *arXiv:2007.00145*, 2020. 8, 9

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 7

[16] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *ACL*, 2018. 4

[17] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. CLIP2Video: Mastering video-text retrieval via image clip. *arXiv:2106.11097*, 2021. 3

[18] Zijian Gao, Jingyu Liu, Sheng Chen, Dedan Chang, Hao Zhang, and Jinwei Yuan. CLIP2TV: an empirical study on transformer-based methods for video-text retrieval. *arXiv:2111.05610*, 2021. 3

[19] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. BridgeFormer: Bridging video-text retrieval with multiple choice questions. In *CVPR*, 2022. 3

[20] Sonam Goenka, Zhaoheng Zheng, Ayush Jaiswal, Rakesh Chada, Yue Wu, Varsha Hedau, and Pradeep Natarajan. FashionVLP: Vision language transformer for fashion retrieval with feedback. In *CVPR*, 2022. 9

[21] Geonmo Gu, Sanghyuk Chun, Wonjae Kim, HeeJae Jun, Yoohoon Kang, and Sangdoo Yun. CompoDiff: Versatile composed image retrieval with latent diffusion. *arXiv:2303.11916*, 2023. 2, 3, 8

[22] Xiaoxiao Guo, Hui Wu, Yu Cheng, Steven Rennie, Gerald Tesauro, and Rogério Schmidt Feris. Dialog-based interactive image retrieval. In *NeurIPS*, 2018. 3

[23] Surgan Jandial, Pinkesh Badjatiya, Pranit Chawla, Ayush Chopra, Mausoom Sarkar, and Balaji Krishnamurthy. SAC: Semantic attention composition for text-conditioned image retrieval. In *WACV*, 2022. 9

[24] Surgan Jandial, Ayush Chopra, Pinkesh Badjatiya, Pranit Chawla, Mausoom Sarkar, and Balaji Krishnamurthy. TRACE: Transform aggregate and compose visiolinguistic representations for image search with text feedback. *arXiv:2009.01485*, 2020. 9

[25] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 3

[26] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019. 18

[27] Jongseok Kim, Youngjae Yu, Hoeseong Kim, and Gunhee Kim. Dual compositional learning in interactive image retrieval. *AAAI*, 2021. 3, 9

[28] Shinonome AI Lab. https://huggingface.co/datasets/shinonomelab/cleanvid-15m_map. 15, 16, 18

[29] Seungmin Lee, Dongwan Kim, and Bohyung Han. CoSMo: Content-style modulation for image retrieval with text feedback. In *CVPR*, 2021. 9

[30] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: ClipBERT for video-and-language learning via sparse sampling. In *CVPR*, 2021. 3

[31] Matan Levy, Rami Ben-Ari, Nir Darshan, and Dani Lischinski. Data roaming and early fusion for composed image retrieval. *arXiv:2303.09429*, 2023. 2, 3, 7, 8, 9

[32] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *CVPR*, 2022. 3

[33] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 3

[34] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2, 6, 7, 22

[35] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 3

[36] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for video+language omni-representation pre-training. In *EMNLP*, 2020. 3

[37] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020. 3

[38] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 8, 22

[39] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv:2304.08485*, 2023. 3

[40] Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. TS2-Net: Token shift and selection transformer for text-video retrieval. In *ECCV*, 2022. 3

[41] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *ICCV*, 2021. 2, 3, 6, 7, 8, 9

[42] Steven Loria. `textblob.readthedocs.io`. 15

[43] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 18

[44] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 3

[45] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval. *arXiv:2104.08860*, 2021. 3

[46] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-CLIP: End-to-end multi-grained contrastive learning for video-text retrieval. In *ACMMM*, 2022. 3

[47] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 3

[48] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 3

[49] Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2action: Cross-modal supervision for action recognition. In *CVPR*, 2020. 3

[50] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 22

[51] Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. Filtering, distillation, and hard negatives for vision-language pre-training. In *CVPR*, 2023. 2, 6, 10

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 3, 6, 7

[53] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned CLIP models are efficient video learners. In *CVPR*, 2023. 3

[54] Kuniaki Saito, Kihyuk Sohn, Xiang Zhang, Chun-Liang Li, Chen-Yu Lee, Kate Saenko, and Tomas Pfister. Pic2Word: Mapping pictures to words for zero-shot composed image retrieval. *CVPR*, 2023. 3, 8, 9

[55] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks Track*, 2022. 3

[56] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 8, 9

[57] Minchul Shin, Yoonjae Cho, ByungSoo Ko, and Geonmo Gu. RTIC: Residual learning for text and image composition using graph convolutional network. *arXiv:2104.03015*, 2021. 9

[58] Robyn Speer. rspeer/wordfreq: v3.0.2. `https://doi.org/10.5281/zenodo.7199437`, September 2022. Zenodo. 4

[59] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *ICLR*, 2019. 3

[60] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *ACL*, 2019. 6

[61] Yuchong Sun, Hongwei Xue, Ruihua Song, Bei Liu, Huan Yang, and Jianlong Fu. Long-form video-language pre-training with multimodal temporal contrastive learning. In *NeurIPS*, 2022. 3

[62] Son Nguyen Thanh. https://pypi.org/project/better-profanity/. 15

[63] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv:2302.13971*, 2023. 2, 4, 7, 9

[64] Hugo Touvron et al. LLaMA 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023. 9

[65] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. 2, 6

[66] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval - an empirical odyssey. In *CVPR*, 2019. 2, 3, 8

[67] Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogério Feris. Fashion IQ: A new dataset towards retrieving images by natural language feedback. In *CVPR*, 2021. 2, 3, 6

[68] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. VideoCLIP: Contrastive pre-training for zero-shot video-text understanding. In *EMNLP*, 2021. 3

[69] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *CVPR*, 2022. 3

[70] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. CLIP-ViP: Adapting pre-trained image-text model to video-language representation alignment. *arXiv*, 2022. 3

[71] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, 2021. 3

[72] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. TACo: Token-aware cascade contrastive learning for video-text alignment. In *ICCV*, 2021. 3

[73] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *ICLR*, 2022. 3

[74] Youngjae Yu, Seunghwan Lee, Yuncheol Choi, and Gunhee Kim. CurlingNet: Compositional learning between images and text for fashionIQ data. *arXiv:2003.1229*, 2020. 9

[75] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv:2111.11432*, 2021. 3

[76] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. MERLOT Reserve: Neural script knowledge through vision and language and sound. In *CVPR*, 2022. 3

[77] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. MERLOT: Multimodal neural script knowledge models. In *NeurIPS*, 2021. 3

[78] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, 2023. 3

[79] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. In *AAAI*, 2020. 3

APPENDIX

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

A Dataset statistics and analysis

In this section, we provide analysis on our WebVid-CoVR. A detailed datasheet can be found at the project webpage.

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 [42] 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 [4] provided by [28]. Finally, using the `better-profanity` library [62], 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 [28]. 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.4, 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.

B Implementation details

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

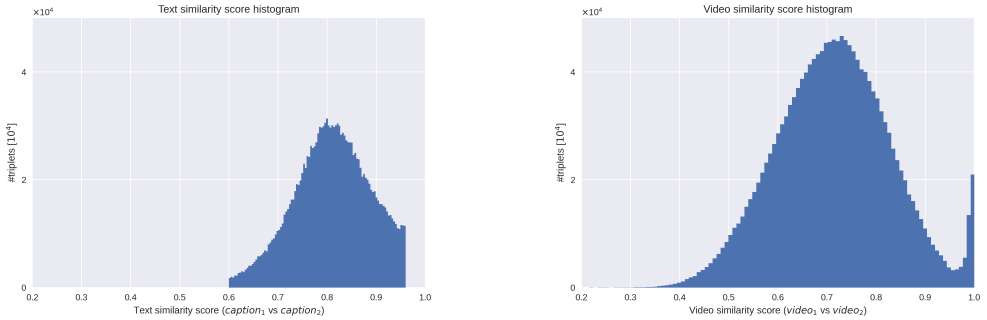

Figure A.1: **Text/video similarity of the caption/video pairs:** Distribution of text similarity scores between caption pairs $(caption_1, caption_2)$ (left) and video similarity scores between video pairs $(video_1, video_2)$ (right), using CLIP embeddings and cosine similarity.

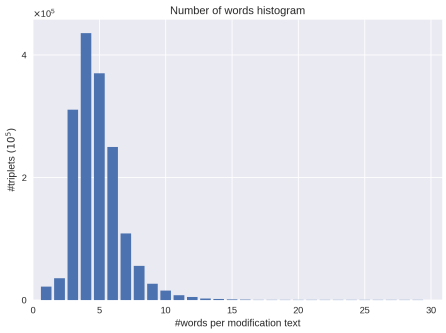

Figure A.2: **Histogram of the number of words in the generated modification text:** Most modification texts have between 3 and 8 words.

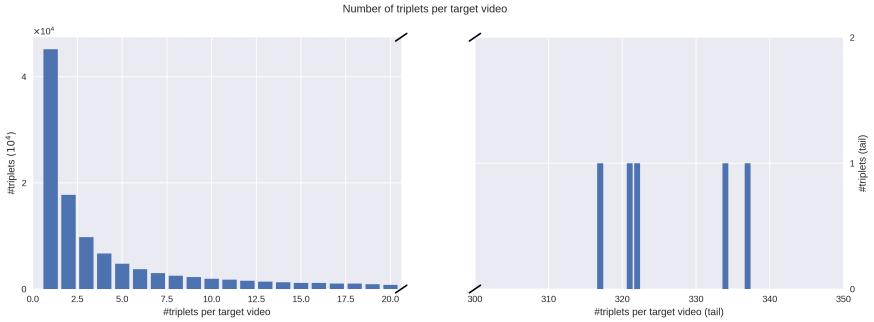

Figure A.3: **Distribution of number of triplets per target video:** We display the histogram depicting the number of triplets associated with each target video in the WebVid-CoVR dataset. Most target videos have 1 or 2 triplets and certain videos exhibit a high number of triplets (zoomed in to the tail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

B.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.

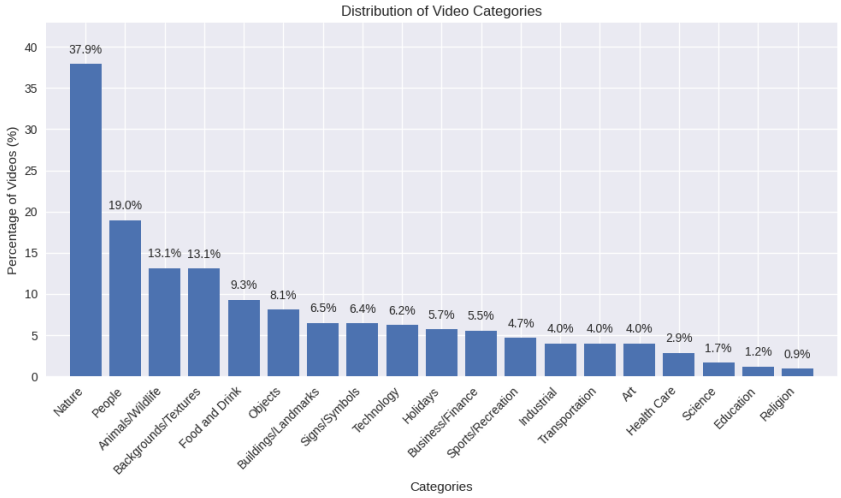

Figure A.4: **Distribution of video categories:** We plot the distribution of categories for videos in WebVid-CoVR, as provided by [28] as WebVid metadata. Note that 50% of our WebVid-CoVR videos are present in this metadata collection. Looking at the distribution, we observe that around 40% and 20% of WebVid-CoVR are videos of Nature and People, respectively.

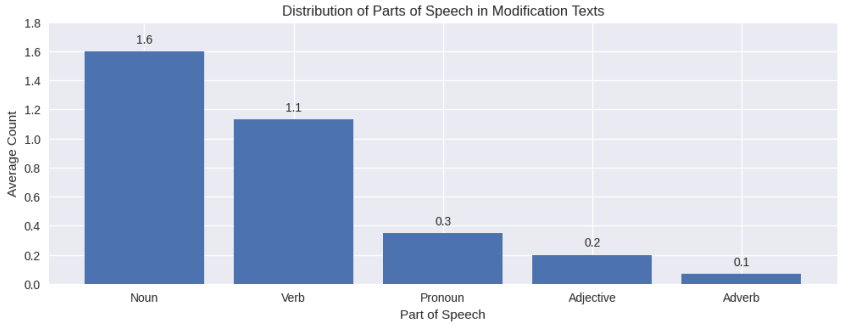

Figure A.5: **Distribution of parts of speech in modification texts:** Distribution of nouns, verbs, pronouns, adjectives, and adverbs in the modification text using part-of-speech (POS) tagging. On average, there are more than one noun and one verb per modification text.

- **Caption similarity search:** To identify captions with one-word differences, we employed the *faiss* library [26] 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 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.

B.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 [43]. For our MTG-LLM, we finetune for one epoch with a batch size of 128 and a learning of $3e{-}5$ that is warmed up linearly for the first 100 steps

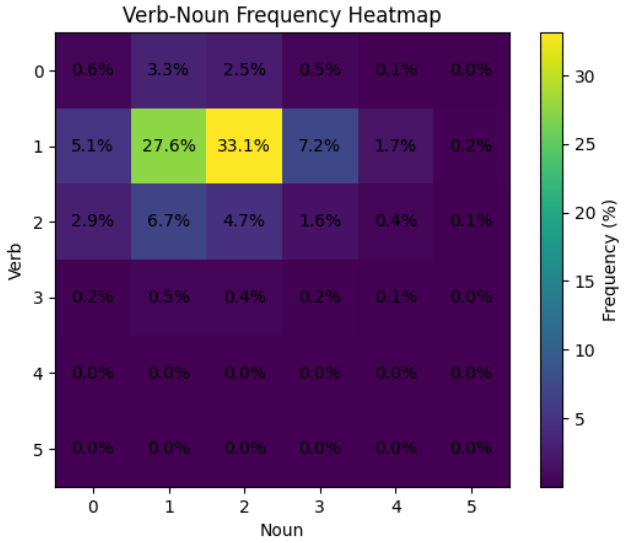

Figure A.6: **Verb-noun heatmap:** This heatmap illustrates the percentage of modification texts containing specific combinations of verbs and nouns. Each cell represents the frequency of a particular verb-noun combination, and the values are presented as percentages. The color intensity indicates the 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.

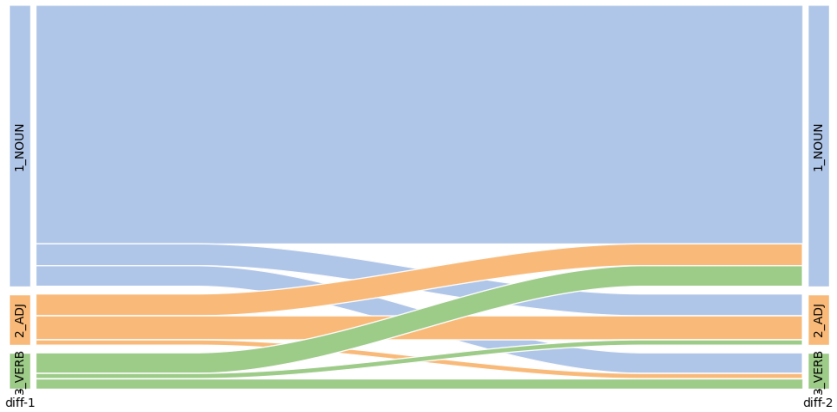

Figure A.7: **Transition of POS tags across the difference words between the two captions:** The visualization primarily focuses on nouns, adjectives, and verbs, which constitute a significant proportion of modifications at 87% (comprising 65% nouns, 13% adjectives, and 9% verbs). The remaining words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

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.

B.3 List of rule-based templates

In the ablation studies (Section 4.4 of the main paper), we introduced a rule-based MTG baseline. Here, in Table A.1, we show the templates used for the rules. We refer to Section D.2 (Table A.6) for qualitative comparison with our finetuned MTG-LLM.

Table A.1: **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`$_2$

B.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 [8]. 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 [8] 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.2.

C Additional experiments

We provide additional experiments, reporting CoVR results obtained by training on data generated with prompting (i.e., without finetuning) the LLM (Section C.1), results when changing the visual query from an image to a video (Section C.2), and varying the pretrained BLIP model (Section C.3).

C.1 Prompting versus finetuning the MTG-LLM

Here, we justify why we finetuned Llama as opposed to simply prompting it without any training. For prompting, 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
```

Then, we concatenate our two captions for which we wish to generate a modification text. Table A.3 shows that finetuning the MTG-LLM for generating the training data is much more effective than

Table A.2: **Added examples to the MTG-LLM training:** We add the below 15 examples to the set of 700 text triplets from [8].

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

Table A.3: **Prompting versus finetuning LLM:** We compare our finetuned model (MTG-LLM) to a prompting baseline (see Section C.1) and observe important gains in the downstream performance of the model trained on the generated data.

Model	WebVid-CoVR-Test				CIRR			
	R@1	R@5	R@10	R@50	R@1	R@5	R@10	R@50
Prompting	51.33	76.68	85.13	96.71	34.94	63.04	74.02	89.83
Finetuning	**53.13**	**79.93**	**86.85**	**97.69**	**38.48**	**66.70**	**77.25**	**91.47**

Table A.4: **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. To keep the computational complexity low, we only 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)	53.17	79.93	86.85	97.69
Video (5 uniform frames)	53.91	79.85	87.09	97.42

Table A.5: **Variants of pretrained BLIP backbones:** We compare the BLIP model without finetuning (base), BLIP finetuned on Flickr30k, and BLIP finetuned on COCO (the one used in the rest of the paper) [34]. For this experiment, we finetune the models on WebVid-CoVR using the cross-attention layers of BLIP as the fusion method, and 15 frames for the target video as in the last row of Table 2.

Backbone	R@1	R@5	R@10	R@50
BLIP Base	50.74	78.91	86.23	97.34
BLIP Flickr30k	52.50	79.46	86.70	**97.77**
BLIP COCO	**53.13**	**79.93**	**86.85**	97.69

prompting it without finetuning, as measured by CoVR performance on WebVid-CoVR-Test and CoIR performance on CIRR. This is also consistent with our qualitative observations: we found that the LLM struggles to perform the modification text generation without finetuning (see Table A.6 in the next section).

C.2 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 5 equally-spaced frames and compute visual embeddings for each frame using the BLIP image encoder. We then average the per-frame embeddings and forward it through the BLIP cross-attention layers to 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.4, using 5 query frames leads to similar performance to using the middle frame.

C.3 Variants of pretrained BLIP backbones

All experiments in this paper are performed with the BLIP model [34] finetuned on COCO [38]. Here, we include experiments when changing this backbone with other pretrained BLIP variants. Specifically, we use the BLIP model without COCO finetuning (BLIP base), and the BLIP model finetuned on Flickr30k [50]. For this experiment (as in the last row of Table 2 of the main paper), we use pretrained cross-attention layers of BLIP as our multimodal combined representation, and finetune them on WebVid-CoVR with 15-frame target video embeddings. In Table A.5, we observe that the BLIP model finetuned on COCO has the highest performance.

D Qualitative analysis

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

D.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, Colorful 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 ingrident of medicine
  - Oxazepam - male doctor with mobile phone opens and touches hologram active ingrident of medicine
  - Blue forget-me-nots
  - Blue galaxy

D.2 Qualitative comparison of MTG approaches

In Section 4.4 of the main paper and Section C.1, 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.6.

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.6, 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. 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.

D.3 Training triplet examples

Figures A.8, A.9, and A.10 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.11, 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.12 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.

Table A.6: **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

D.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.13 and Figure A.14, 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.14, 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").

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

In Figure A.15, 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).

D.6 Qualitative CoIR results on the CIRR benchmark

In Figure A.16, 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.

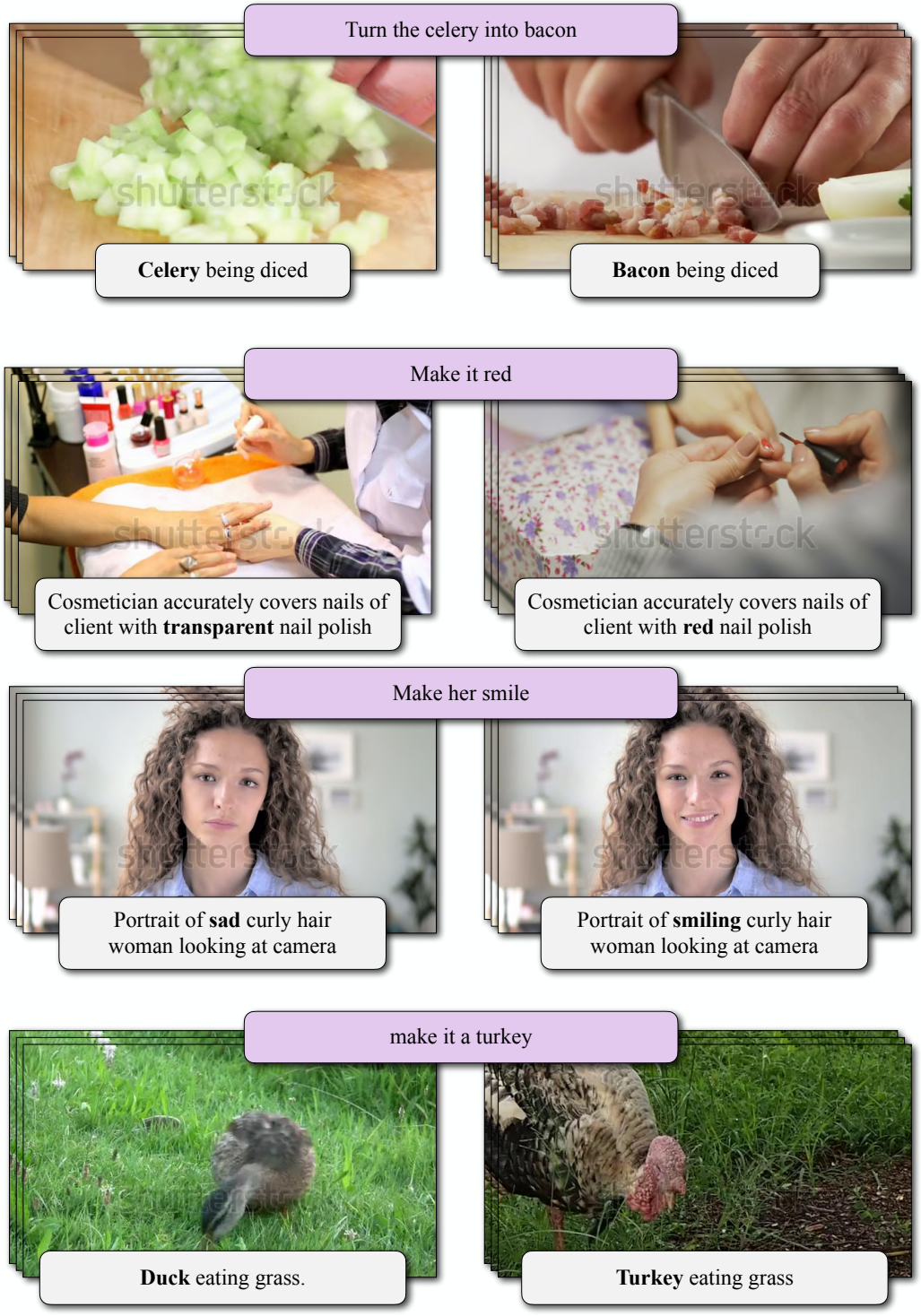

Figure A.8: **Examples of generated triplets:** We illustrate triplet samples (one per row) generated using our automatic dataset creation methodology. Each sample consists of two videos with their corresponding captions (at the bottom of each video) and the generated modification text using our MTG-LLM (in purple).

Figure A.9: **Examples of generated triplets (ctd)**

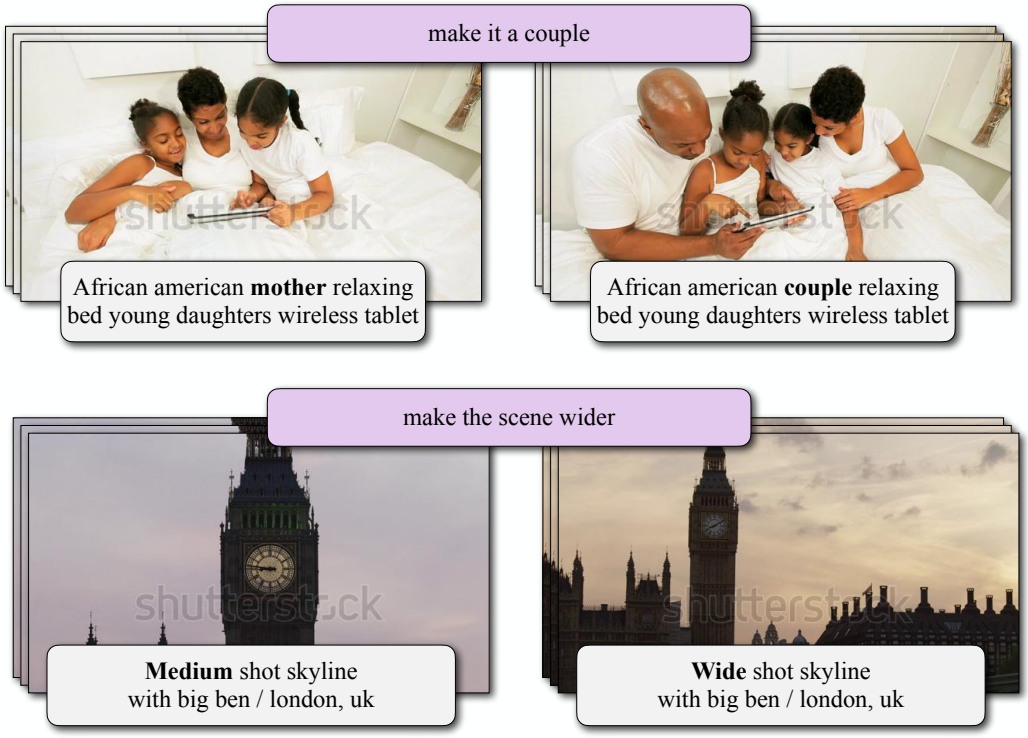

Figure A.10: **Examples of generated triplets (ctd)**

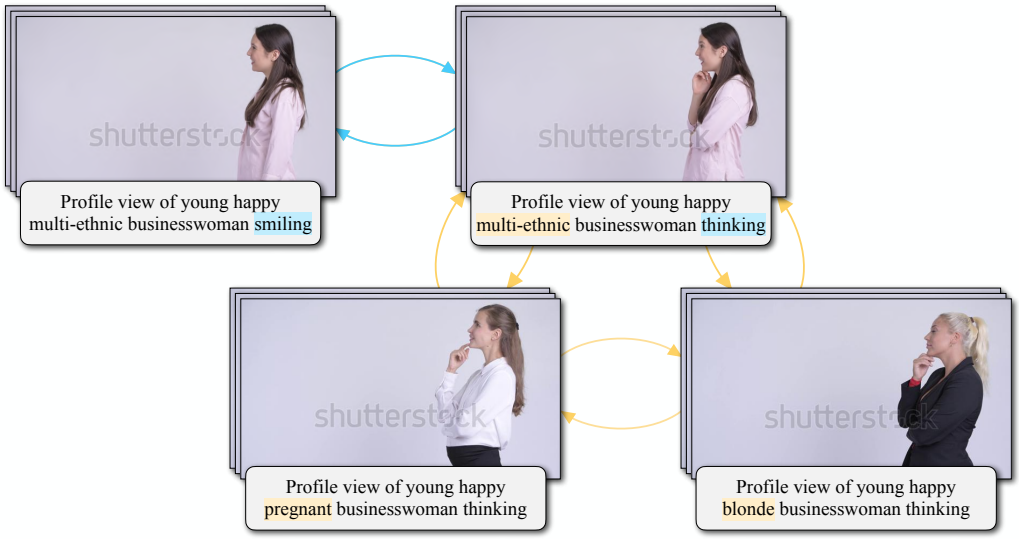

Figure A.11: **Generated triplets from multiple similar captions:** We can train with as many triplets as pairs of captions with one word difference by generating modification texts using our trained MTG-LLM: *she is thinking*, *Have her look happy*, *Make the businesswoman pregnant*, *make her blonde*, *make her multi-ethnic*, *Make the woman pregnant*, etc.

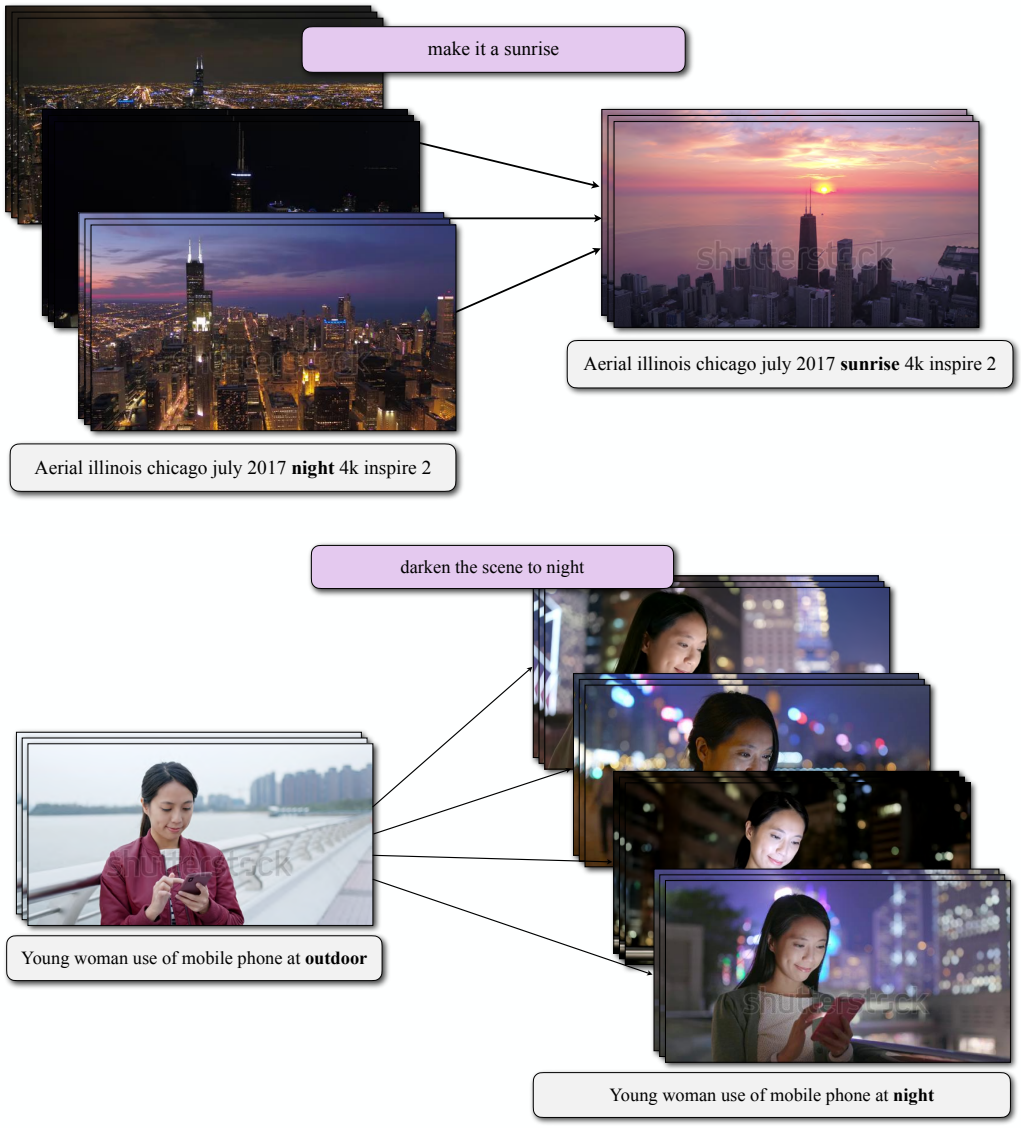

Figure A.12: **Generated triplets with multiple videos:** In cases where there are several videos with the same caption, we can generate multiple triplets by leveraging the multiple videos. It can be seen as a way of data augmentation.

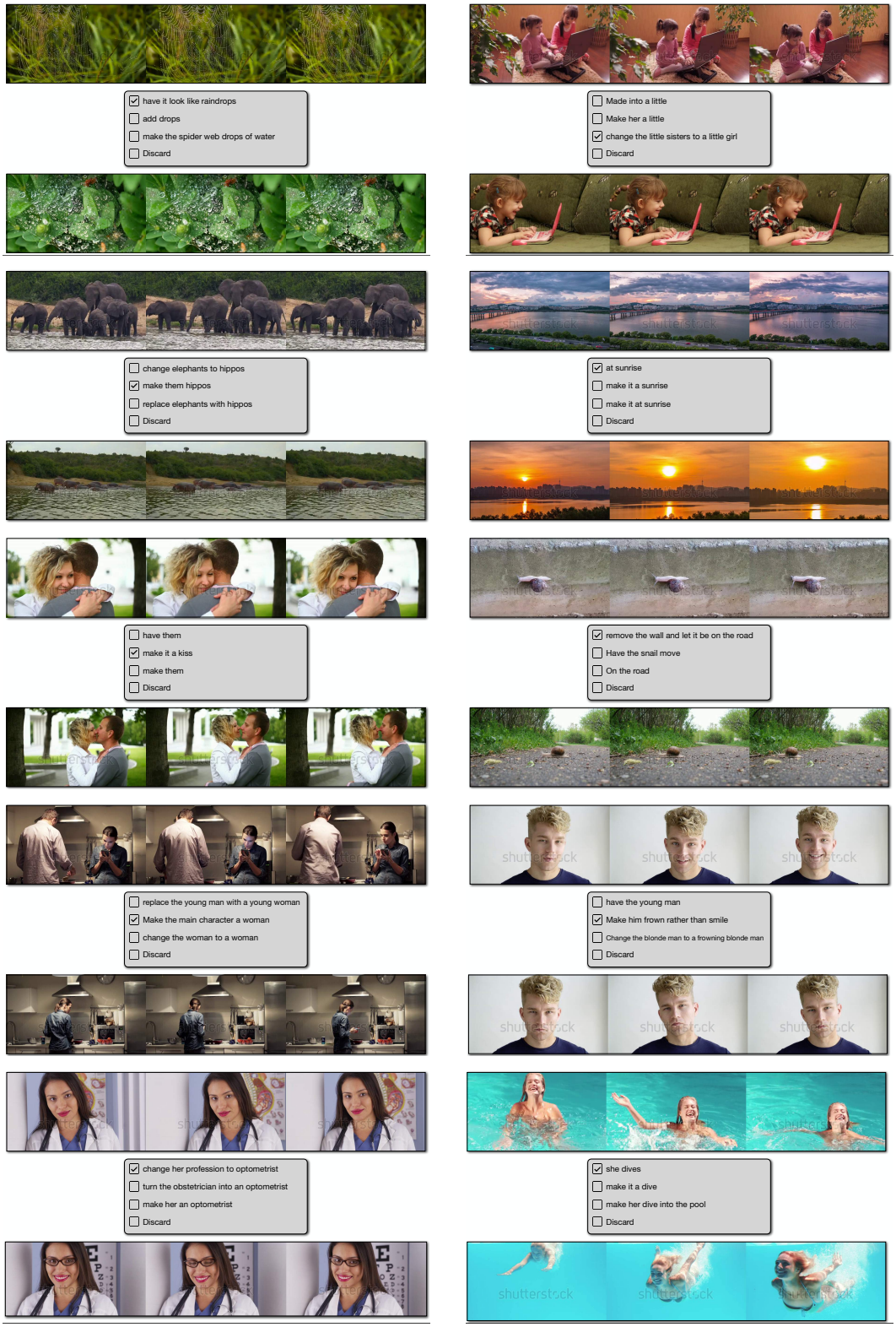

Figure A.13: **Manual annotation examples (kept):** We show samples from WebVid-CoVR-Test which are automatically mined triplets that are marked as correct during the annotation process. Each sample consists of two videos and a set of modification text options (in between each video pair). The chosen modification text is indicated by a checkmark.

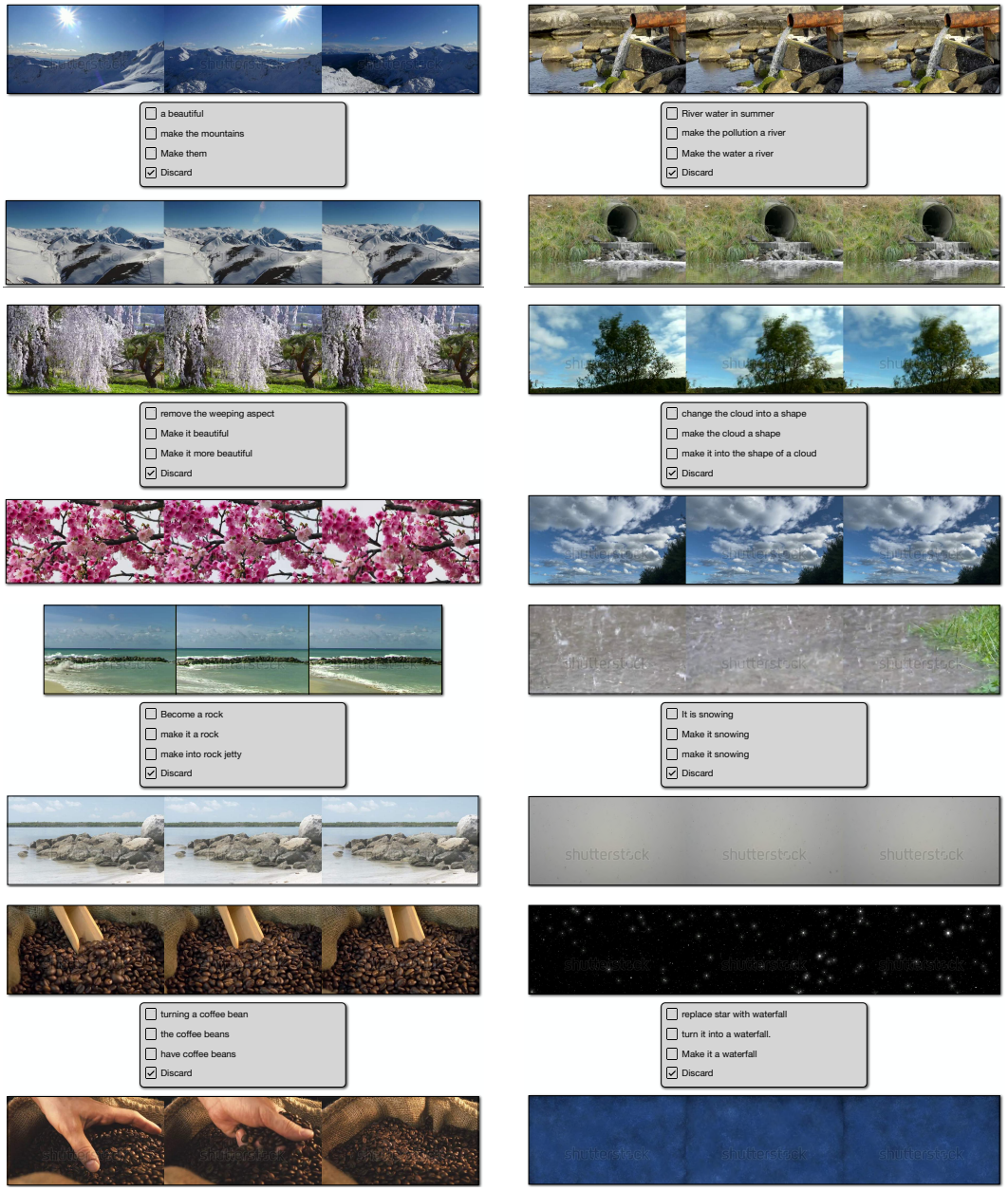

Figure A.14: **Manual annotation examples (discarded):** We show automatically mined triplets that are discarded during the annotation process. Discarded texts include videos that are too similar (bottom left), too dissimilar (bottom right), or have bad modification texts (top left).

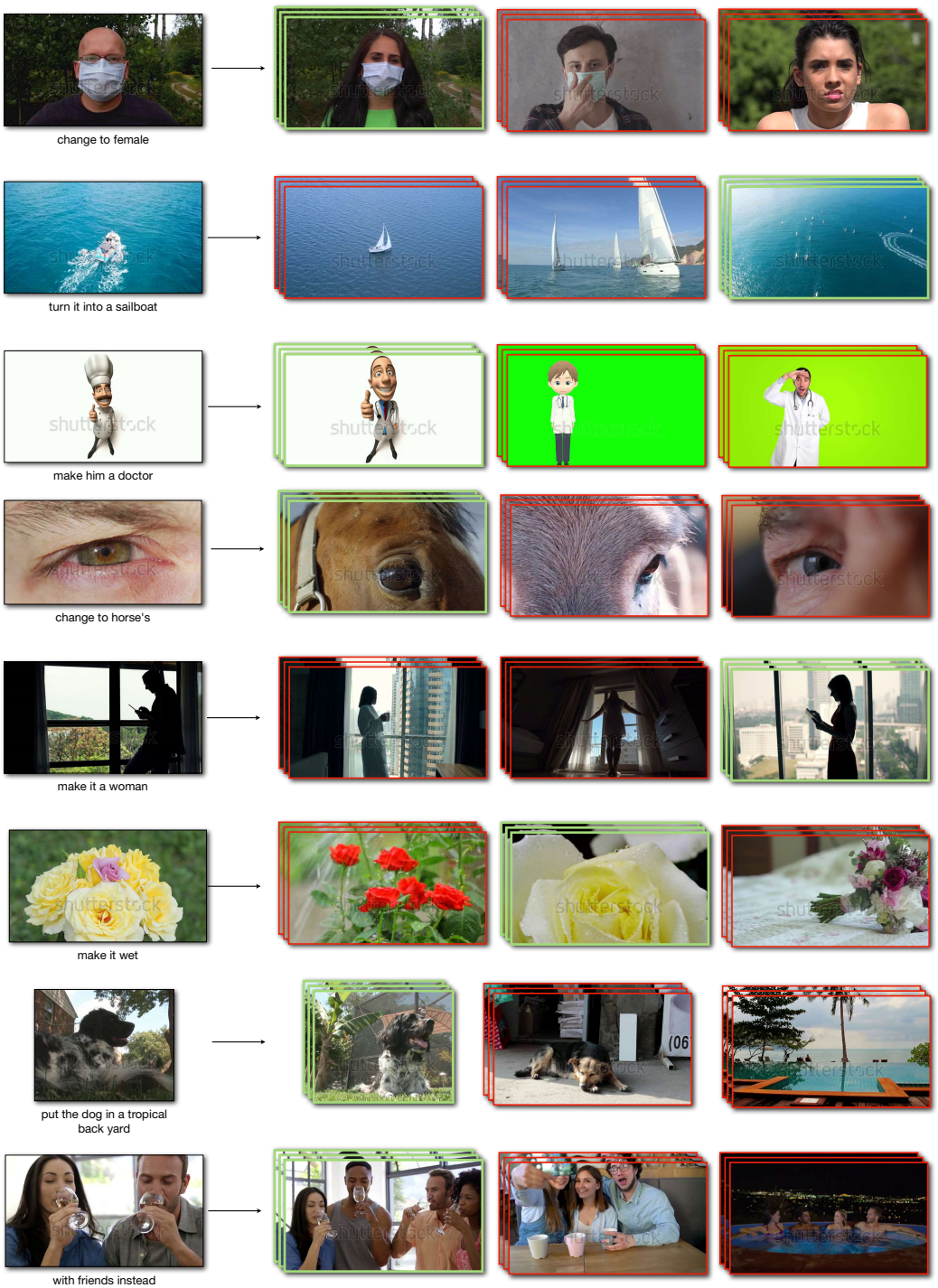

Figure A.15: **Qualitative CoVR results on WebVid-CoVR-Test:** We display the input image and modification text queries on the left, along with the top 3 retrieved videos by our model on the right. Ground-truth is denoted with a green border.

Figure A.16: **Qualitative CoIR results on CIRR:** 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.).