Topic A: Composed Image Retrieval

May 19, 2025

Brahim SAADI
ENS Paris-Saclay
Paris
brahim.saadi@ens-paris-saclay.fr

Dani BOUCH
ENS Paris-Saclay
Paris
dani.bouch@ens-paris-saclay.fr

Abstract

Composed Image Retrieval (COIR) aims to identify a target image by leveraging a reference image alongside textual instructions that specify the desired changes. Building upon this foundation, Ventura et al. [1] introduce Composed Video Retrieval (CoVR), an extension that integrates both text and video queries to enhance the retrieval capabilities within video databases. Addressing the challenges of traditional COIR methods, the authors present an automated dataset creation process and introduce a new framework for (CoVR). In our work, we first replicate the baseline results from [1] to validate their approach. We then propose two key enhancements: a dynamic weighting scheme that effectively fuses text, image, and multi-modal embeddings, and modifications to the dataloader that incorporate advanced sampling strategies, including hard negative sampling, filtered triplet sampling, and a combination of both. We were able to achieve a slight improvement in performence compared to our established baseline.

1. Introduction

Composed Image Retrieval (COIR) seeks to locate a target image in a database when provided with both a visual example and a complementary text query describing how the target differs from the reference. As illustrated in Figure 1, this dual-modality approach allows users to express fine-grained modifications (e.g., color change, scene context, or additional attributes) that may be challenging to convey with text alone. By combining the expressive power of visual examples with concise textual refinements, COIR has emerged as a compelling paradigm for various applications, including product search, content curation, and visual exploration. Professionals in media-related fields can accelerate their editing workflows by pinpointing relevant variants of a base reference, while casual users may explore different angles of a tourist scene or experiment with aesthetic choices.

Figure 1. **Overview of Composed Image Retrieval (COIR) [1]**

2. Model Architecture: CoVR-BLIP2

The **CoVR-BLIP2** framework [1], initially developed for *Composed Video Retrieval (CoVR)*, as illustrated in Figure 2, CoVR-BLIP2 integrates a BLIP-2 image encoder with a Q-Former module to combine visual and textual inputs into a unified embedding space. For CoVR tasks, each video frame is independently encoded and then aggregated, typically using a weighted mean, to form a single *video embedding* $h(v)$. On the query side, a *reference image* q and a *modification text* t are merged to generate a multi-modal embedding $f(q,t)$. The training objective is designed to align $f(q,t)$ with $h(v)$, enhanced by a caption retrieval loss for improved textual grounding.

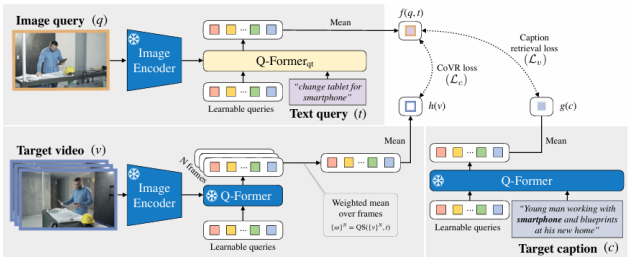


Figure 2. **Architecture of CoVR-BLIP2. [1]**

This approach can be seamlessly extended to CoIR by treating the target as a single image rather than multiple frames. Instead of aggregating embeddings from individual frames, the model encodes a single image v, maintaining the contrastive alignment between $f(q,t)$ and $h(v)$ without requiring frame-level fusion.

3. Main Contributions

3.1. Reproducing Paper's Baseline Results

Before delving into novel experiments, our first step was to replicate one of the baseline results from the *CoVR-2* paper. Utilizing a pretrained BLIP-2 checkpoint (fine-tuned on COCO) provided in the official repository, we conducted a zero-shot evaluation on the CIRR dataset without any additional training. From this quick test, we obtained a Recall@1 of 46.33. This replication both verified our local setup and ensured we could accurately reproduce the official environment prior to implementing our own modifications.

3.2. Establishing Our Baseline

We set out to establish our own baseline. Since our resources were limited—we only had a single **GPU** and less memory—we chose to lower our **batch size** to 64 and train for only **two epochs**. We still used the BLIP-2 checkpoint but now trained it on the CIRR dataset, using a **NVIDIA L4 GPU**. Because of these changes, our performance is slightly lower than the paper's published result at different Recalls@k—but it's still in the same ballpark. As shown in Table 1, this new result serves as our baseline going forward. It lets us iterate much faster while still being comparable to the original setup in[1] . We can also see an qualitative example in Section 5 *Evaluation example on CIRR dataset with the baseline model (Section 5.1)*.

Method	K=1	K=5	K=10	K=50
Reported Result	**50.87**	**80.80**	**88.84**	**98.00**
Our Model	48.58	79.28	87.81	97.81

Table 1. Results on the CIRR dataset comparing the Reported Result and Our Model.

3.3. Embedding Weighting

A primary challenge in Composed Image Retrieval (COIR) is effectively balancing the influence of the text and image modalities. Depending on the query, certain tasks may rely more heavily on textual modifications (e.g., changing color), while others depend predominantly on visual aspects (e.g., altering shape). The original **CoVR-BLIP2** framework utilizes a single multi-modal embedding $f(q,t)$ to represent the combined query. However, this approach may inadvertently lose information specific to either modality when the query heavily favors one over the other. To address this, we propose learning a weighted combination of three distinct embeddings: text-only embedding, an image-only embedding, and the existing multi-modal embedding. We directly have access to $f(q,t)$ and need to extract the text (f_{text}) and image (f_{img}) embeddings as follows:

- **Text Embedding (f_{text}):** We obtain the text embedding by extracting the **CLS token** from the **BLIP-2** text encoder and projecting it using a linear layer:

$$f_{\text{text}} = \text{Normalize}(\text{Linear}(\text{CLS}))$$

- **Image Embedding (f_{img}):** The image embedding is derived by aggregating the image features from the Q-Former's output and then projecting them via another linear layer:

$$f_{\text{img}} = \text{Normalize}(\text{Linear}(\text{Q-Former Output}))$$

To enhance the effectiveness of **Composed Image Retrieval (COIR)**, we explored various strategies for combining the text-only embedding f_{text}, the image-only embedding f_{img}, and the multi-modal embedding $f(q,t)$. Our objective was to identify the most effective method for balancing these embeddings to improve retrieval performance. Specifically, we experimented with: averaging the text and image embeddings as $(f_{\text{text}} + f_{\text{img}})/2$, averaging all three embeddings as $(f_{\text{text}} + f_{\text{img}} + f(q,t))/3$, utilizing a simple two-layer **MLP** to determine the weights, and employing a deeper four-layer **MLP** for more complex weighting.

For the MLP approaches, the network learns weights w_1, w_2, and w_3 to combine the embeddings as follows:

$$\text{Combined\_Embedding} = w_1 \cdot f(q,t) + w_2 \cdot f_{\text{img}} + w_3 \cdot f_{\text{text}}$$

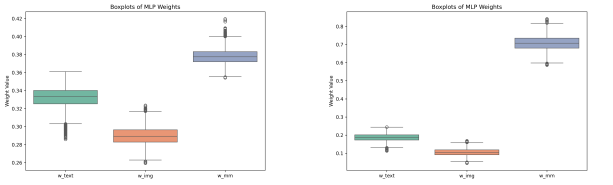
Method	K=1	K=5	K=10	K=50
Our Model	**48.58**	**79.28**	**87.81**	97.81
Text+Image Avg	33.85	69.33	81.59	96.46
Text+Img+MM Avg	45.90	78.05	86.94	97.76
MLP (2 Layers)	46.62	77.95	86.94	**97.97**
MLP (4 Layers)	48.07	78.82	87.22	97.90

Table 2. Comparison of results on the CIRR dataset for various embedding combination methods.

We found that excluding the multi-modal embedding hurts performance. Once we bring it back, results improve. The MLP approaches let the model dynamically adjust how much weight to give to text, image, and multi-modal embeddings, leading to slightly higher recall compared to simple averaging.

This leads us to examine how the MLP actually distributes its weights on real CIRR data. As shown in Figure 3a and Figure 3b, both the two-layer and four-layer MLP tend to assign higher weights to the multi-modal embedding. Interestingly, the deeper MLP leans even more heavily on it. This signals a stable weighting scheme that focuses on the multi-modal embedding. A potential solution would involve ensuring that the image and text embeddings are closely aligned while maintaining the multi-modal

embedding's alignment with both, thereby preserving valuable information across modalities. This idea is explored in [2], which proposes a Dynamic Weighted Combiner to mitigate these issues. Due to lack of time, we weren't able to implement this but instead experimented with a simpler variation, as detailed in Section 5 *Consistency Loss Experiment (Section 5.2)*.



(a) **Weight Distribution for 2-Layer MLP** (b) **Weight Distribution for 4-Layer MLP**

Figure 3. Weight distributions assigned by the two-layer and four-layer MLPs. Both models tend to prioritize the multi-modal embedding, with the deeper MLP showing a stronger preference.

3.4. Dataloader Modification

Another area of improvement we experimented with was modifying the dataloader. The **CIRR** dataset provides "member groups"—sets of images that are visually or contextually related. By organizing related images within the same batch, we encourage the model to learn finer discriminations.

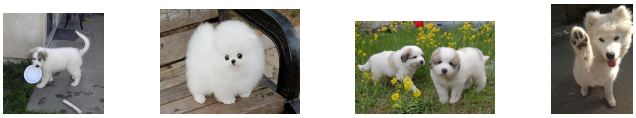


Figure 4. **Hard Negative Examples**

In this experiment, we implemented Hard Negative Sampling [3] by ensuring that each batch contains all images from the same member groups. This approach enhances the model's ability to distinguish subtle differences between closely related images. Conversely, we also explored Filtered Triplet Sampling (FTS), which prevents multiple member groups from appearing within the same batch, thereby simplifying the contrastive learning objective. To leverage the strengths of both methods, we introduced a combined sampling strategy defined by the equation:

$$\text{Combined\_Sampling} = \alpha \cdot \text{HN} + (1 - \alpha) \cdot \text{FTS}$$

where α is a coefficient that controls the probability of selecting between Hard Negative Sampling and Filtered Triplet Sampling for each batch.

Method	K=1	K=5	K=10	K=50
Our Baseline	48.58	**79.28**	87.81	97.81
HN Sampling	46.34	76.60	85.40	96.84
Filtered Sampling	47.76	62.41	78.31	97.86
0.4 HN + 0.6 FTS	48.05	78.88	**87.85**	**97.89**
0.5 HN + 0.5 FTS	48.43	78.45	84.22	96.86
0.6 HN + 0.4 FTS	**49.83**	78.41	82.85	95.02

Table 3. Comparison of results on the CIRR dataset between different sampling strategies.

Analyzing the results in Table 3, we observe that higher values of α, which assign more weight to Hard Negative Sampling (HN), lead to improved Recall@k for smaller values of k. Conversely, lower α values, emphasizing Filtered Triplet Sampling (FTS), enhance Recall@k at larger k values. A possible imporvement would be to finetune this hyperparameter alpha to find the best possible value.

We also observed that training with Hard Negative Sampling using the Hard Negative Contrastive Loss (HNCE) led to unstable training dynamics. To mitigate this, we employed Cross Entropy Loss (CL) with the hard negative loader which improves performence for Recall@k at all values of k and it also leads to a more stable training Section 5 *Dataloader effect on training curves: (Section 5.3)*. The performance comparisons are presented in Table 4.

Method	K=1	K=5	K=10	K=50
HN HNCE loss	46.34	76.60	85.40	96.84
HN Cross Entropy loss	**47.81**	**78.84**	**87.37**	**97.98**

Table 4. Comparison of results on the CIRR dataset using Hard-Negative Sampling with Hard Negative Contrastive Loss (HNCE) and Cross Entropy Loss (CL).

4. Conclusion

Throughout this project, we encountered significant challenges, including limited computational resources, tight deadlines, and the complexity of handling large datasets and models. Despite these obstacles, we successfully navigated and leveraged the existing official implementation codebase and were able to reproduce some of the paper's results. We also complemented that by experimenting with embedding weighting and modifying the dataloader where we were able to achieve slight improvements over our baseline results. Looking ahead, we'd like to continue some of the experiments we started and attempt to implement better dynamic weighting methods and look into more complex dataloading techniques.

References

[1] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. Covr-2: Automatic data construction for composed video retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11409–11421, December 2024. 1, 2

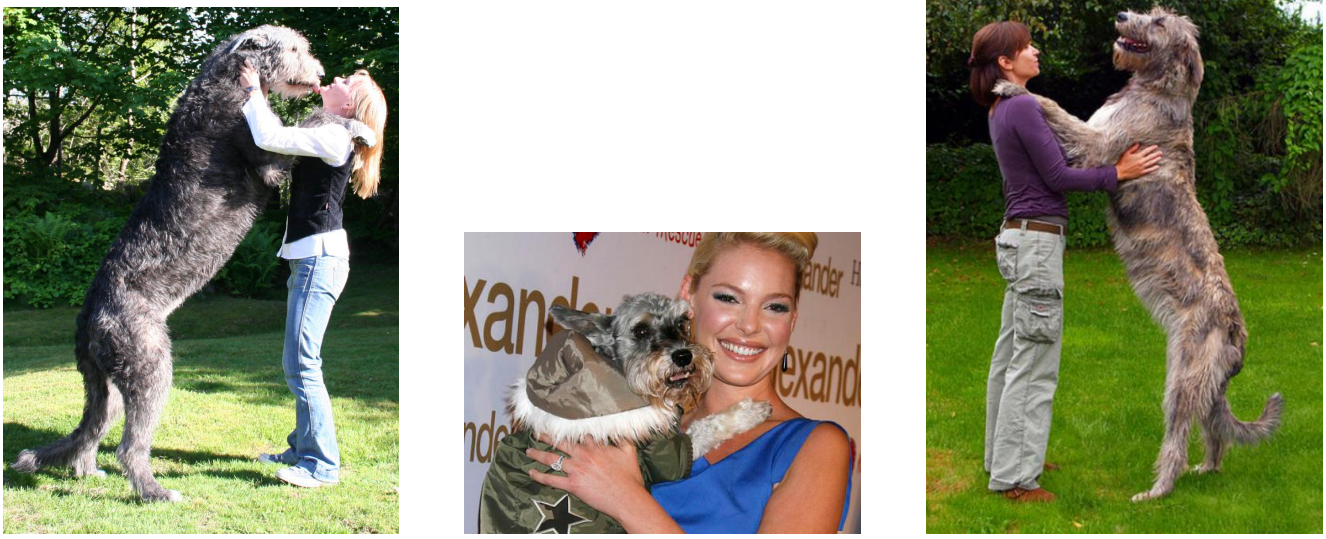
[2] Fuxiang Huang, Lei Zhang, Xiaowei Fu, and Suqi Song. Dynamic weighted combiner for mixed-modal image retrieval, 2023. 3

[3] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples, 2021. 3

5. Appendix

5.1. Evaluation example on CIRR dataset example using our Baseline Model

Figure 5. Modificiation caption: Remove all but one dog and add a woman hugging it



1. 2. 3.

Figure 6. Top 3 Retrieved Images

5.2. Consistency Loss Experiment

To ensure consistency between different modalities, we introduced a self-distillation loss using KL divergence. The total loss is defined as:

$$L_{\text{total}} = L_{\text{HardNegNCE}} + \lambda L_{\text{distill}}$$

where

$$L_{\text{distill}} = \text{KL}(f_{\text{text}} \parallel f(q,t)) + \text{KL}(f_{\text{img}} \parallel f(q,t))$$

For $\lambda = 0.05$, we obtained the following results:

Method	K=1	K=2	K=5	K=10	K=50
HNCE + Distillation Loss	45.06	59.277	76.096	86.0	97.566
HNCE Loss (Our baseline)	48.58	79.28	87.81	97.81	97.81

Table 5. Comparison of results on the CIRR dataset using Hard Negative Contrastive Loss (HNCE) and HNCE with Distillation Loss.

The results show poor performence due to the simplistic method of the experiment using only KL divergence and due to the fact the parameter λ would need finetuning and maybe experementing with more complex loss functions.

5.3. Dataloader effect on training curves:

We can see that the loss starts off much higher with hard negative sampling because of harder batches each time and that the training is unstable.

Figure 7. Training curves comparing Filtered Sampling and Hard Negative Sampling.

Compared to HNCE, Cross Entropy loss achieves more stable training.

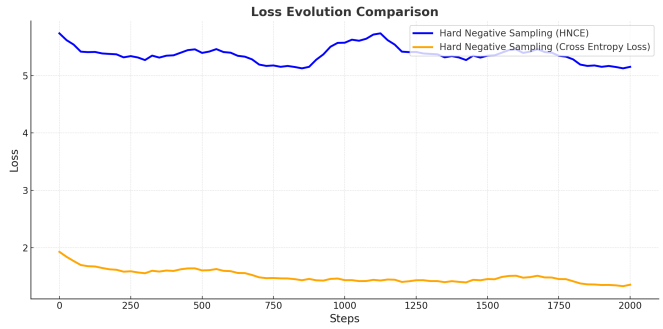


Figure 8. Training curves comparing Hard Negative Contrastive Loss (HNCE) and Cross Entropy Loss (CL).