Composed Image Retrieval using Contrastive Learning and Task-oriented CLIP-based Features

ALBERTO BALDRATI, Università degli Studi di Firenze - MICC, Italy and Università di Pisa, Italy
MARCO BERTINI, Università degli Studi di Firenze - MICC, Italy
TIBERIO URICCHIO, Università degli Studi di Macerata, Italy
ALBERTO DEL BIMBO, Università degli Studi di Firenze - MICC, Italy

Given a query composed of a reference image and a relative caption, the Composed Image Retrieval goal is to retrieve images visually similar to the reference one that integrates the modifications expressed by the caption. Given that recent research has demonstrated the efficacy of large-scale vision and language pre-trained (VLP) models in various tasks, we rely on features from the OpenAI CLIP model to tackle the considered task. We initially perform a task-oriented fine-tuning of both CLIP encoders using the element-wise sum of visual and textual features. Then, in the second stage, we train a Combiner network that learns to combine the image-text features integrating the bimodal information and providing combined features used to perform the retrieval. We use contrastive learning in both stages of training. Starting from the bare CLIP features as a baseline, experimental results show that the task-oriented fine-tuning and the carefully crafted Combiner network are highly effective and outperform more complex state-of-the-art approaches on FashionIQ and CIRR, two popular and challenging datasets for composed image retrieval. Code and pre-trained models are available at https://github.com/ABaldrati/CLIP4Cir

CCS Concepts: • **Computing methodologies** → *Image representations*; *Visual content-based indexing and retrieval.*

Additional Key Words and Phrases: multimodal retrieval, combiner networks, vision language model

ACM Reference Format:
Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. 2023. Composed Image Retrieval using Contrastive Learning and Task-oriented CLIP-based Features. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1 (August 2023), 23 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Content-Based Image Retrieval (CBIR) is a fundamental task in multimedia and computer vision which has undergone a continuous evolution since its early years [46], moving from the use of engineered features like SIFT to CNNs [33, 56]. It has been applied to many different specialized domains like artworks and cultural heritage [4, 12], commerce [17, 54], surveillance [2], nature [24, 25]. In the basic form, the query is composed of only an image, of which features are computed and compared with the ones extracted by a database of images.

We can extend CBIR systems to improve their effectiveness by adding additional information to the query image. For example, interactive image retrieval systems extend CBIR systems by adding some form of user feedback, e.g. to provide some measure of relevance [5]. In composed image retrieval, the visual query is extended

Authors' addresses: Alberto Baldrati, alberto.baldrati@unifi.it, Università degli Studi di Firenze - MICC, Firenze, Italy, Università di Pisa, Pisa, Italy; Marco Bertini, marco.bertini@unifi.it, Università degli Studi di Firenze - MICC, Firenze, Italy; Tiberio Uricchio, tiberio.uricchio@unimc.it, Università degli Studi di Macerata, Macerata, Italy; Alberto Del Bimbo, alberto.delbimbo@unifi.it, Università degli Studi di Firenze - MICC, Firenze, Italy.

1551-6857/2023/8-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Fig. 1. The left portion of the illustration depicts a specific case of composed image retrieval in the fashion domain, where the user imposes constraints on the character attribute of a t-shirt. Meanwhile, the right part showcases an example where the user asks to alter objects and their cardinality within a real-life image.

to an image-language pair [37] where a short textual description, typically expressed in natural language, may request constraints and desired changes or add specifications on some attributes of the retrieved results [26]. Figure 1 illustrates two examples of this task. In both queries, a user selects a reference image and then provides additional requests in the form of text, e.g. asking to change details, texture, color, or shape features of the reference image. Composed image retrieval systems find applications in various domains such as web search, e-commerce, and surveillance. However, developing solutions for this task can be challenging due to the need for incorporating feedback and user intent while addressing the semantic gap between image and text content.

Very recently, researchers proved that deep neural networks combining visual and language modalities like CLIP [41], ALIGN [28], and the more recent method proposed in [9], trained using an image-caption alignment objective on large-scale internet data, can obtain impressive zero-shot transfer on a myriad of downstream tasks like image classification, text-based image retrieval, and object detection.

In this work, we show that features obtained from vision and language pretrained (VLP) models – we employed CLIP-based features – can be effectively used to implement a composed image retrieval system where user feedback is provided as natural language input to provide additional (or contrasting) requirements concerning those embedded in the visual features of the image used to query the system. Firstly, we apply the system to the fashion domain, performing experiments on the challenging FashionIQ dataset [51]. Then, to study the generalization capabilities to a broader image domain, we perform experiments on the newly introduced CIRR dataset [37]. Experiments show that the proposed approach obtains state-of-the-art results on both datasets.

To summarize, we highlight our main contributions as follows:

- We propose a novel task-oriented fine-tuning scheme for adapting vision-language models to the composed image retrieval task. The aim of such a task-oriented adaptation scheme is to reduce the mismatch between the large-scale pre-training and the downstream task.
- We propose a novel two-stage approach that combines task-oriented fine-tuning with the training of a Combiner network which can perform a fine-grained merging of the multimodal features. This two-stage approach achieves state-of-the-art results on two standard and challenging datasets: FashionIQ and CIRR.
- We address the issue of using the CLIP model with images having a high aspect ratio since the CLIP visual encoder can input only square pictures. We propose a novel preprocess pipeline suited for image retrieval tasks that helps to reduce content information loss compared to the standard CLIP preprocess pipeline.
- To provide further insight into the workings of our proposed system, we perform several qualitative experiments. The first experiment aims to demonstrate how our approach affects the feature distribution in the embedding spaces and the impact of pairwise feature distances on retrieval performance. Additionally, we report visualization experiments utilizing the gradCAM technique [44] to gain a deeper understanding of the image portions that are most significant during retrieval.

2 RELATED WORKS

Traditional CBIR does not use user feedback or its intent to refine results. However, within interactive and composed CBIR, much work has been done to improve retrieval performance by incorporating user's feedback in terms of relevance to the query [42] or by considering relative [30] and absolute attributes [20, 55]. The limiting expressiveness of attributes was successively addressed in [19, 48] by considering purely textual feedback, allowing richer expressiveness. Nonetheless, the performance of the textual model can limit the system in understanding and reacting appropriately.

Visual and language pre-training

Models like GPT-2, BERT [15] and GPT-3 [6] have shown that large amounts of text combined with recent improvements in attention mechanisms enable learning of powerful features that integrate vast knowledge. Adding images to the learning process, CLIP [41] has very recently shown that it is feasible to perform multimodal zero-shot learning, obtaining remarkable feature generalization of both images and text. CLIP is a deep neural network trained to predict the association between text snippets and paired images. Unlike standard vision models trained on specific datasets that are typically good at only one task, this new class of models learns associations between images and natural language supervision that are widely available on the internet. They are not directly optimized for a benchmark and yet can perform consistently well on different tasks. CLIP effectiveness is still subject of study [1], with first applications to art [13], image generation [11] and zero-shot video retrieval [18], event classification [32], visual commonsense reasoning [49]. Our work builds upon CLIP and further explores its potential in the composed image retrieval task, applying the proposed approach to a specific domain, i.e. fashion, and also to general images. ALIGN [28] uses a dual-encoder architecture to learn the alignment of visual and language representations of image and text pairs using a contrastive loss in a noisy dataset. The extremely large scale of such a dataset, composed of 1 billion pairs, twice the size of the CLIP training dataset, makes up for its noise and leads to state-of-the-art representations even using such a simple learning scheme. Differently from CLIP and ALIGN, the authors in [9] propose a data-efficient contrastive distillation method that learns from a training dataset that is 133× smaller than the one used by CLIP (400 million pairs), using a ResNet50 image encoder and DeCLUTR text encoder.

Composed image retrieval

In the growing area of image retrieval with user feedback that combines images and text, our work relates to two recently introduced datasets that address the composed image retrieval task: *i)* FashionIQ, a fashion image retrieval with text [51], and with *ii)* the very recent composed image retrieval of generic images introduced in [37]. In [8], a transformer that can be seamlessly plugged into a CNN to selectively preserve and transform the visual features conditioned on language semantics is presented. Text Image Residual Gating (TIRG) [48] combines image and text features using gating and residual features. The authors of [45] leverage skip connections by combining them with graph neural networks, resulting in improved performance. The authors of [31] employ two different neural network modules to address image style and content. In [29], the authors present a Correction Network which explicitly models the difference between the reference and target image in the embedding space. In [37], a new dataset (CIRR) for composed image retrieval on real-world images is proposed, along with a novel transformer-based model that uses rich pre-trained vision-and-language knowledge, called CIRPLANT, to modify visual features conditioned on natural language. CIRPLANT leverages visual-and-language pre-trained models in composed image retrieval: the OSCAR model [34] is carefully adapted to the task with promising results. In [16], the authors proposed the Modality-Agnostic Attention Fusion (MAAF) model to tackle the composed image retrieval task. The model treats the convolutional spatial image features and learned text embeddings as modality-agnostic tokens and passes them to a Transformer for further processing. In [36], the authors

propose a Multi-Grained Fusion (MGF) module which fuses features at different stages. ComposeAE [3] is an autoencoder-based model that learns the composition of image and text features for retrieving images by adopting a deep metric learning (DML) approach instead of fusing them by passing through a few fully connected layers. CurlingNet, proposed in [52], measures the semantic differential relationships between images concerning a conditioning query text. The main components are two networks: the first one, called the Delivery filter, delivers the source image to the candidate cluster according to a given query in embedding space, while the second one, called the Sweeping filter, checks the attributes highlighted in the query and learns the path from the center of valid target candidates to the target image. In [53], the composed image retrieval task is extended to a multi-turn conversation. The authors proposed a system that utilizes ComposeAE [3] to combine image and text at each turn. The combined representation is then fed into a recurrent network, following the turn order, for further processing. In [26], the authors present the SAC (Semantic Attention Composition) framework, which consists of two modules: the Semantic Feature Attention (SFA) module finds the salient regions of the image w.r.t. the text, and then the Semantic Feature Modification (SFM) module determines how to change the relevant parts of the image compositing coarse and fine salient image features computed by SFA with text embeddings.

The proposed method starts with the hypothesis of having a unified embedding space for images and text achieved through the Vision-Language model CLIP. In the first stage, we fine-tune both CLIP encoders to adapt them to the composed image retrieval task. Next, using the task-adapted embedding spaces, we train a Combiner network to merge the multimodal features. In contrast to fashion-oriented approaches like [8, 31], our method does not rely on spatial features. Instead, we argue that when considering images of a broader domain, the semantics hold greater significance than local visual aspects.

3 THE PROPOSED METHOD

The proposed approach addresses the multimodal task of composed image retrieval. The input query consists of a reference image I_q (e.g., an image of a black shirt with a cartoon lion) and a relative caption T_q that includes a descriptive request from the user about the image (e.g., "has dog print and is dark grey color"). The goal is to retrieve target images that satisfy similarity constraints imposed by both the input components (e.g., an image of a dark grey shirt with a dog print, as shown in Fig. 3). For a successful retrieval, the system should understand the semantics of the image and the meaning of the text, integrate the multi-domain information, and then use the fused representation to retrieve the relevant images.

In contrast to previous works like [8, 29, 31, 45] that build from different image and textual models, we start from the hypothesis of having a unified embedding of images and text, obtained through using the CLIP model [41]. CLIP is a vision-language model trained to align images and their corresponding text captions in a unified embedding space. It consists of an image encoder ψ_I and a text encoder ψ_T. Given an image I, the image encoder extracts a feature representation $\psi_I(I) \in \mathbb{R}^d$, where d is the size of the CLIP embedding space. Similarly, for a given text caption T, the text encoder extracts a feature representation $\psi_T(T) \in \mathbb{R}^d$. CLIP learns to map similar concepts expressed in images and text to similar feature representations. For instance, given the image of a cat I_c and the text T_c *"a photo of a cat"*, the way CLIP is trained should guarantee that $\psi_I(I_c) \approx \psi_T(T_c)$.

We argue that, even though having a unified embedding space is a good starting point, it is not exactly what we need in the task we are considering. In composed image retrieval, the goal is to move from the reference to the target image point in the image embedding space with the aid of textual information. Hence, instead of utilizing a unified image-text embedding space, our approach involves creating two separate embedding spaces that can be combined through a sum operation. Formally, given an image of a black dress I_x and the corresponding text T_y ("is blue"). Let I_z represent the image of a blue dress. Our aim is to shape the embedding spaces such that $\psi_I(I_x) + \psi_T(T_y) \approx \psi_I(I_z)$. When this equation is satisfied, we can affirm that the textual embedding space exhibits strong "additivity properties" in relation to the image space, or equivalently, the embedding spaces are additive.

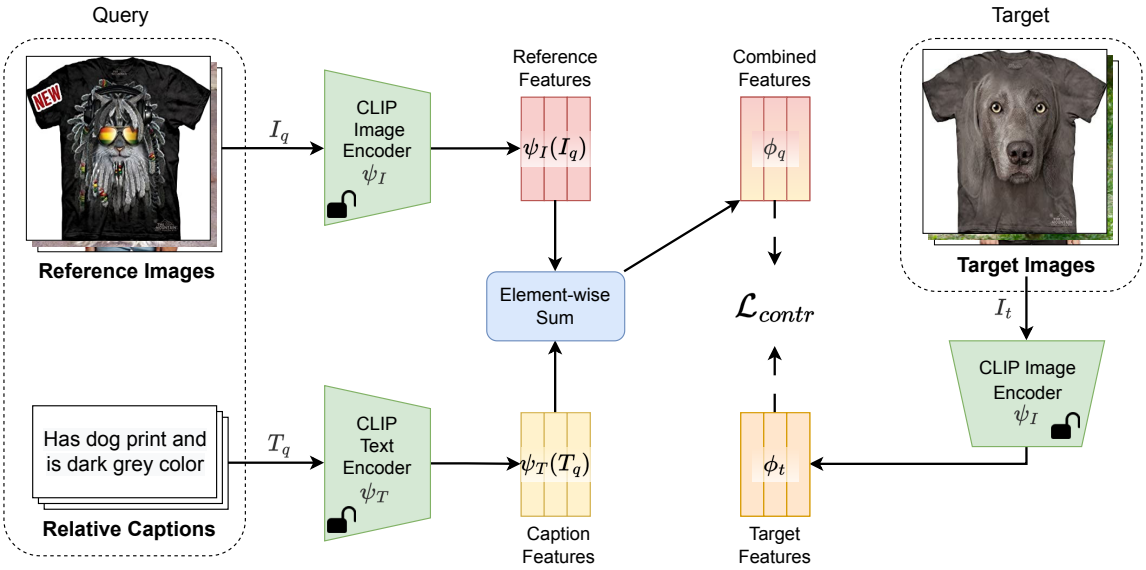


Fig. 2. First stage of training. In this stage, we perform a task-oriented fine-tuning of CLIP encoders to reduce the mismatch between the large-scale pre-training and the downstream task. We start by extracting the image-text query features and combining them through an element-wise sum. We then employ a contrastive loss to minimize the distance between combined features and target image features in the same triplet and maximize the distance from the other images in the batch. We update the weights of both CLIP encoders.

Ideally, the embeddings of the relative caption should correspond to the displacement vector from the query image to the target image features, i.e. $\psi_T(T_y) \approx \psi_I(I_z) - \psi_I(I_x)$.

We propose a two-stage approach to address the task of composed image retrieval by taking full advantage of the capabilities of CLIP's features. In the first stage, we tackle the objective mismatch between the large-scale pretraining of CLIP and the downstream task: we propose a novel fine-tuning scheme tailored to improving the additivity properties of the embedding spaces. In the second stage, starting from the task-oriented features, we train from scratch a Combiner neural network that learns to perform a fine-grained combination of image-text features. Although we train the Combiner network from scratch, we design its structure to take full advantage of the first stage of training (see Section 3.2 for more details). During both stages the training is performed using triplets (I_q, T_q, I_t), where $q = (I_q, T_q)$ is the query and I_t is the target image that we aim to retrieve given q.

At inference time, given a query (I_q, T_q), we utilize the fine-tuned CLIP encoders and the trained Combiner network to generate the combined features. Subsequently, following the standard image-to-image retrieval approach, we compute the cosine distances between the combined features and the database of index image features. The results are then sorted based on their similarity.

3.1 Task-oriented fine-tuning

In this stage, we adapt both CLIP's encoders to composed image retrieval reducing the mismatch between the large-scale pre-training and the downstream task. Given a query consisting of a reference image I_q and a relative caption T_q, we extract their feature representations using the CLIP image encoder ψ_I and text encoder ψ_T respectively. This results in $\psi_I(I_q) \in \mathbb{R}^d$ and $\psi_T(T_q) \in \mathbb{R}^d$, where d denotes the size of the CLIP embedding space. To combine the query features, we perform an element-wise sum, resulting in $\phi_q = \psi_I(I_q) + \psi_T(T_q)$.

Our objective is to minimize the distance between the query combined features ϕ_q and the target image features $\phi_t = \psi_I(I_t)$ belonging to the same triplet and, at the same time, maximize the distance from the other target images in the same batch. To this end, following [31, 45, 48], we employ a batch-based contrastive loss:

$$\mathcal{L}_{contr} = \frac{1}{B}\sum_{i=1}^{B} -\log \frac{\exp\{\tau * \kappa(\phi_q^i, \phi_t^i)\}}{\sum_{j=1}^{B} \exp\{\tau * \kappa(\phi_q^i, \phi_t^j)\}} \tag{1}$$

Here, $\kappa(\cdot)$ denotes the cosine similarity, τ is a temperature parameter that controls the range of the logits, and B is the number of images in a batch. We update the weights of both CLIP encoders. We use this loss because, being a batch-wise contrastive loss, it does not require the definition of a sampling strategy: it considers all negative samples in a mini-batch. Figure 2 shows an overview of the task-oriented fine-tuning stage.

Using the element-wise sum as the combination of query features goes in the direction of making CLIP's embedding spaces more additive. Consequently, similar concepts expressed in text and images no longer share similar features. Instead, the textual features serve as displacement vectors from the query to the target in the image space. From a high-level perspective, we notice that, in composed image retrieval, the image and the text do not play the same role. The task is not symmetric with respect to the input: we start from an image, and we would like to retrieve another image using textual guidance. For this reason, the break up of the unified embedding space is not an undesirable side-effect.

We will denote the fine-tuned image encoder and text encoder as $\overline{\psi_I}$ and $\overline{\psi_T}$, respectively.

3.2 Combiner training

During the training of the Combiner network, we follow the same general framework as in the previous stage. However, this time we train from scratch the Combiner network instead of updating the weights of the CLIP encoders. In contrast to the first stage, we use the Combiner network C_θ to combine the query features. Specifically, the combined features are obtained as $\overline{\phi_q} = C_\theta(\overline{\psi_I}(I_q), \overline{\psi_T}(T_q))$. We optimize the Combiner network by utilizing the $\mathcal{L}_{contr}$ loss described in Eq. (1) with $\overline{\phi_q}$ and $\overline{\phi_t} = \overline{\psi_T}(I_t)$ as inputs. Figure 3 depicts a visual overview of the Combiner network training stage. By employing the contrastive loss, we train the Combiner C_θ to produce features as close as possible to the target features and as far away as possible from all other image features.

The Combiner network, depicted in Fig. 4, is designed to take full advantage of the first stage of training and the increased additivity properties of the adapted embedding spaces. The idea is to learn the residual of a convex combination of the image-text query features. We begin by projecting the text and image features through a linear transformation followed by a ReLU function. The resulting projected features are then concatenated and passed to two separate branches. The first branch, labeled as (1) in Fig. 4, is responsible for computing the coefficients of a convex combination between the image and text features. To compute these coefficients, we feed the concatenated features into a linear layer, followed by the ReLU function, another linear layer, and the sigmoid function. The sigmoid output provides the coefficients needed for the query image-text convex combination. The second branch, labeled as (2), outputs the mixture contribution of the image and text features. The structure of this branch is the same as the first branch, except it does not include the final sigmoid function. Finally, we sum the convex combination of the query features and the learned image-text mixture. To reduce overfitting, we apply dropout to each layer.

By denoting the outputs of the first branch (1) as λ and $1-\lambda$, and the output of the second branch (2) as v, we can express the combined features as $\overline{\phi_q} = (1-\lambda) * \overline{\psi_I}(I_q) + \lambda * \overline{\psi_T}(T_q) + v$. Notably, the convex combination $(1-\lambda) * \overline{\psi_I}(I_q) + \lambda * \overline{\psi_T}(T_q)$ is a generalization of the element-wise sum of the query features. Consequently, as the embedding spaces exhibit stronger additivity properties, the Combiner's effectiveness in its task is enhanced. We intentionally design the Combiner to capitalize on the task adaptation achieved during the first stage of training.

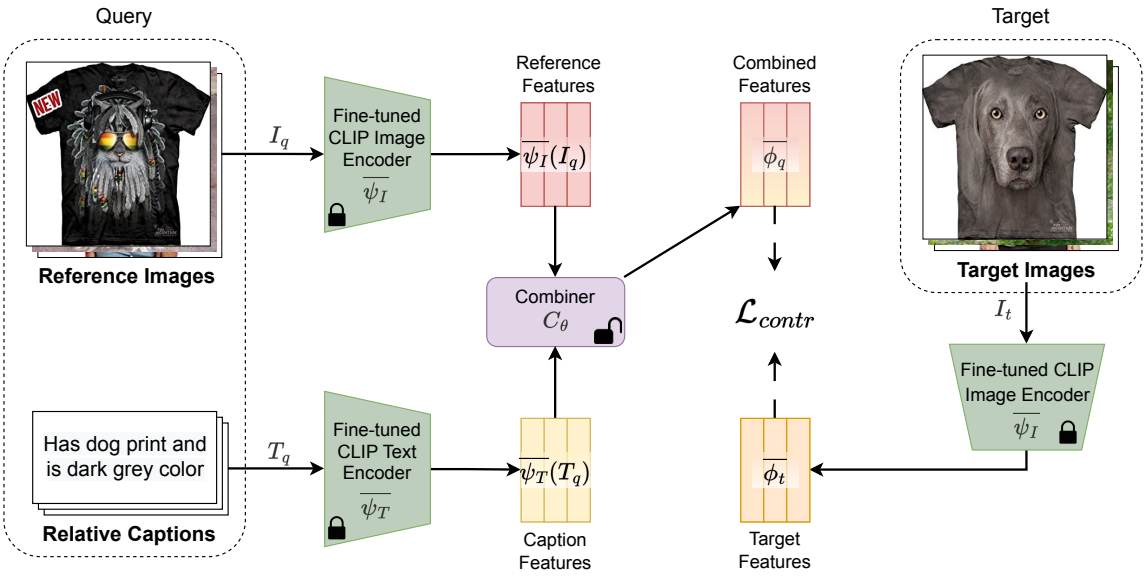


Fig. 3. Second stage of training. In this stage, we train from scratch a Combiner network that learns to fuse the multimodal features extracted with CLIP encoders. We start by extracting the image-text query features using the fine-tuned encoders, and we combine them using the Combiner network. We then employ a contrastive loss to minimize the distance between combined features and target image features in the same triplet and maximize the distance from the other images in the batch. We keep both CLIP encoders frozen while we only update the weights of the Combiner network. At inference time the fine-tuned encoders and the trained Combiner are used to produce an effective representation used to query the database.

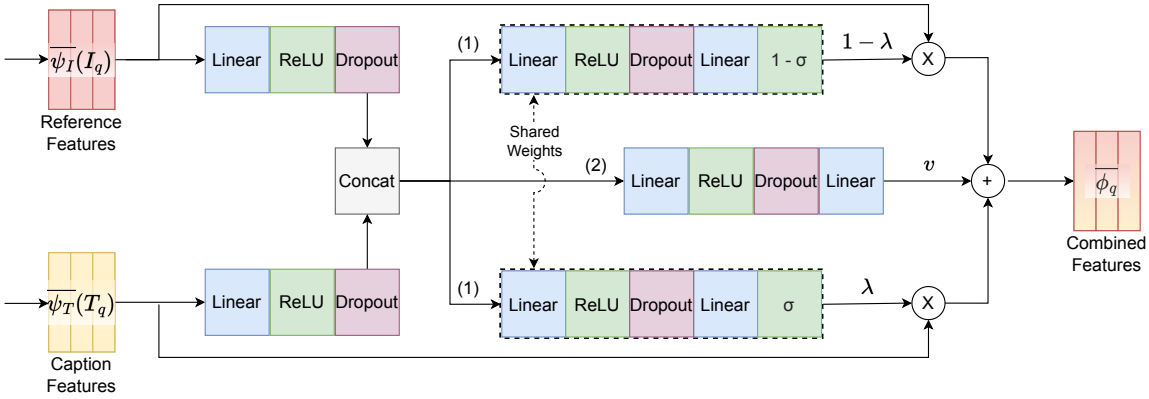
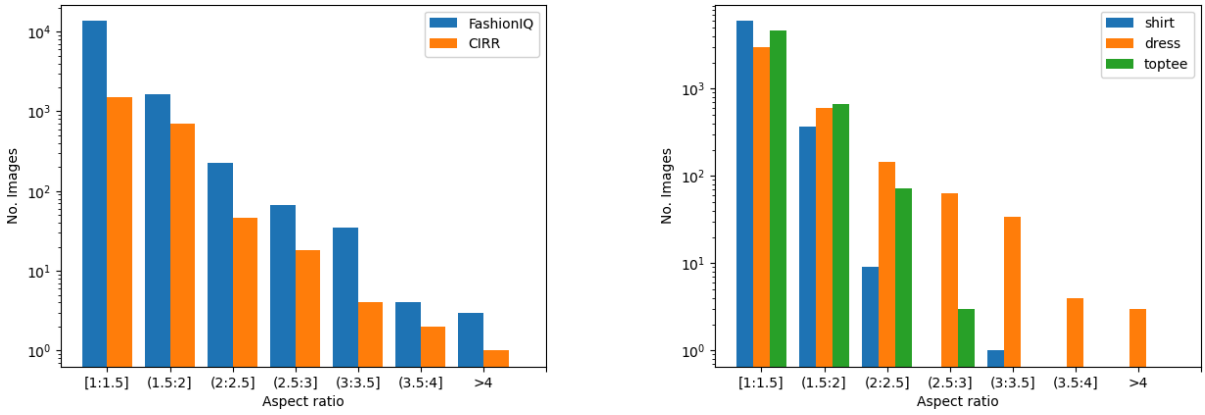


Fig. 4. Architecture of the Combiner network C_θ. It takes as input the multimodal query features and outputs a unified representation. σ represents the sigmoid function. We denote the outputs of the first branch (1) as λ and $1-\lambda$, while the output of the second branch (2) as v. The combined features are $\overline{\phi_q} = (1-\lambda) * \overline{\psi_I}(I_q) + \lambda * \overline{\psi_T}(T_q) + v$



(a) Dataset comparison

(b) FashionIQ categories

Fig. 5. Histogram of image aspect ratios in FashionIQ and CIRR datasets (a) and the three categories of FashionIQ (b). The x-axis represents the aspect ratio defined as $max(width, height)/min(width, height)$ while the y-axis represents the number of images (in logarithmic scale). The width of each bin is 0.5, and the first bin starts at 1. More than half of the dataset's images are skewed and have at least a 1.5 aspect ratio. In the FashionIQ dataset, the issue is evident in the Dress category.

3.3 Preprocess Pipeline

The standard preprocess pipeline of CLIP is mainly composed of two steps: a resize operation where the smaller side of the image matches the CLIP input dimension $input\_dim$ followed by a center crop operation which results in a square patch $input\_dim \times input\_dim$ output. Subsequently, as the ratio between the largest and the smaller side increases, the area of the image lost after the preprocess increases. From now on, we will say that an image has a high aspect ratio when it is far from having a square shape. In Fig. 5 is shown how, in the datasets we consider (detailed in Section 4), the number of images with a high aspect ratio is not negligible. As can be seen, this is especially true for the FashionIQ *dress* category and the CIRR dataset.

One way to address the loss of information due to the center crop operation is to apply zero-padding to match the smaller side to the larger side, effectively squaring the image. Although this approach eliminates the loss of content information, it also reduces the resolution of the useful portion of the image since the CLIP image encoder input dimension cannot change. Thus, we develop a preprocessing pipeline that seeks to balance the two approaches discussed above. Specifically, we apply padding to an image only if its aspect ratio exceeds a predefined target ratio. Additionally, instead of squaring the image, we adjust its aspect ratio to match the target ratio when padding is applied. The pseudocode for the proposed preprocess pipeline is shown in Algorithm 1.

Figure 6 presents the preprocess pipelines as mentioned earlier. It is evident that when the ratio between the larger and smaller sides deviates significantly from one, the standard CLIP preprocess removes a substantial portion of the image, which considerably hampers the retrieval process. Although the visual disparities between the square pad and the proposed pad (with a target ratio of 1.25) approaches are not substantial, we will demonstrate that the model benefits from having such an increased usable portion in the images during retrieval.

4 EXPERIMENTAL RESULTS

4.1 Implementation details

We perform the experiments using two CLIP models of different sizes. The smallest one relies on a modified ResNet-50 (RN-50) [22] architecture. It takes input images of 224×224, and the size of its embedding space is $d = 1024$. The biggest one, denoted as RN-50x4, follows the EfficientNet-style model scaling and uses approximately $4\times$ the computation of RN-50. It takes input images of 288×288, and the size of its embedding space is $d = 640$.

```
# in_image: input image to be preprocessed
# target_ratio: target aspect ratio
# dim: CLIP image encoder input dimension

def preprocess(in_image, target_ratio, dim):
    w, h = in_image.size
    aspect_ratio = max(w, h) / min(w, h)

    # pad the image only if the aspect ratio
    # is above a fixed target
    if aspect_ratio < target_ratio:
        out_image =  in_image
    else:
        # zero-pad the image to bring its aspect
        # ratio to target ratio
        scaled_max_wh = max(w, h) / target_ratio
        hp = max((scaled_max_wh - w) // 2, 0)
        vp = max((scaled_max_wh - h) // 2, 0)
        padding = (hp, vp, hp, vp)
        out_image = pad(in_image, padding, 0)

    # Resize and center crop the image
    out_image = resize(out_image, dim)
    out_image = center_crop(out_image, dim)
return out_image
```

Algorithm 1: Python-style pseudocode of the proposed preprocess pipeline.

Fig. 6. Comparison among different preprocesses pipelines. The proposed padding method results in images that contain more details than square padding and provides a better overview than the standard CLIP padding.

In the Combiner network (Figure 4), the first two linear layers before the concatenation have input-output dimensionality equal to $(d, 4d)$. After the concatenation in both branches, we have two linear layers. In the first branch (1), the first linear layer has input-output dimensionality of $(4d, 8d)$, and the second one $(8d, 1)$. In the other branch, the first linear layer has input-output dimensionality of $(4d, 8d)$ while the second one $(8d, d)$. Following the standard practice, we set the dropout rate to 0.5. During retrieval, we normalize both the combined and index set features to have a unit L_2-norm.

Following the original CLIP training strategy, in the fine-tuning stage, we employed AdamW optimizer [38] with a learning rate of $2e-6$ and a weight decay coefficient of $1e-2$. Due to GPU memory constraints, we set the batch size to 512 for fine-tuning the RN-50-based CLIP model and 192 for fine-tuning the RN-50x4-based model. We kept the batch normalization layer frozen. We fine-tuned the CLIP encoders for a maximum of 150 epochs. During the training of the Combiner network, we keep both fine-tuned CLIP encoders frozen and only train the Combiner function. We set the learning rate to $2e-5$ and train the model for a maximum of 300 epochs. We set the batch size to 4096 when using both backbones. We used the PyTorch library throughout the experiments. We set the target ratio in the preprocessing pipeline to 1.25. Following the approach described in [41], we set the parameter τ in Eq. (1) to 100. This value ensures that the logits have a sufficient dynamic range. To mitigate overfitting, we adopt an early stopping strategy. We use mixed-precision training [39] to save memory and speed up the training in both stages. We employ gradient checkpointing [7] to further reduce memory usage.

We conduct all experiments on a single NVIDIA Titan RTX (24GB) GPU. The first stage of training requires approximately 4 hours for the RN-50x4 model and 2 hours for the RN-50 model. The training of the Combiner network takes less than an hour for both models.

4.2 Datasets and metrics

4.2.1 FashionIQ. FashionIQ [51] is composed of 77,684 fashion images crawled from the web and split into the train, validation, and test sets, divided into three different categories: *Dress*, *Toptee* and *Shirt*. Among the 46,609 training images, there are 18,000 training triplets made of a reference image, a pair of relative captions, and a

target image. The captions describe properties to modify in the reference image to match the target image. The validation and test sets consist of 15,537 and 15,538 images, respectively, with 6,017 and 6,119 triplets.

We follow the standard experimental setting as in [29, 31]. We employ the average recall at rank K (Recall@K) as an evaluation metric, namely Recall@10 (R@10) and Recall@50 (R@50). Note that for each triplet, there is only a positive index image. Hence, each query has R@K zero or one. All results are on the validation set since, at the time of writing, test set ground-truth labels have not been released yet.

4.2.2 CIRR. The authors of [37] designed the CIRR dataset to address two common problems encountered in composed image retrieval datasets, such as FashionIQ. These problems are the lack of sufficient visual complexity caused by the restricted image domain and the numerous false negatives due to the unfeasibility of extensively labeling target images for each (reference, text) pair. As a result, some images in the dataset that correspond to valid matches for a query are not labeled as valid targets. CIRR (Compose Image Retrieval on Real-life images) dataset consists of 21,552 real-life images taken from the popular natural language reasoning $NLVR^2$ dataset [47]. It has the same structure as the FashionIQ dataset and contains 36,554 triplets randomly assigned in 80% for training, 10% for validation, and 10% for the test. The dataset images are grouped in multiple subsets of six semantically and visually similar images. To have negative images with high visual similarity the relative captions are collected describing the differences between two images in the same subset.

The standard evaluation protocol proposed by the authors of the dataset is to report the recall at rank K (Recall@K) at four different ranks (1, 5, 10, 50). Moreover, thanks to the unique design of the CIRR dataset, it is also possible to report the $\text{Recall}_{\text{Subset}}$ metric that considers only the images in the subset of the query. This *subset* metric has two main benefits: it is not affected by false-negative samples and, thanks to negative samples with high visual similarity, it captures fine-grained image-text modifications. The reference metrics are the R@5 which accounts for possible false negatives in the entire corpus, and the R_{Subset}@1, which better illustrates the fine-grained reasoning abilities.

4.3 Task-oriented fine-tuning effects

In this section, we present a set of experiments that illustrate how the task-oriented fine-tuning of CLIP encoders and their increased additivity properties contribute to easing the task of the Combiner network and help to improve retrieval performance. For each dataset, we compare the performance varying the combining function and the modality of the CLIP fine-tuning. Throughout all the experiments, we use the RN-50 CLIP model. For each fine-tuning modality, we train from scratch a different Combiner network. We report the results in Table 1 for the FashionIQ dataset and in Table 2 for the CIRR dataset.

Notably, the element-wise sum of out-of-the-box CLIP features achieves impressive results without domain or task-specific training on both datasets. This performance is intriguing as it demonstrates that the CLIP image-text common embedding space exhibits good additivity properties, even though its training objective does not explicitly optimize for this aspect. Fine-tuning only the CLIP image encoder brings an interesting performance boost compared to the out-of-the-box CLIP features. This improvement is expected when employing the element-wise sum as the combining function, given that the out-of-the-box CLIP features lack domain or task-specific training. However, the most promising improvement occurs when utilizing the trained Combiner network. The text encoder fine-tuning achieves slightly better performance than image encoder fine-tuning. We can notice that on the FashionIQ dataset, the improvement over the image encoder fine-tuning remains constant when using either the element-wise sum or the Combiner network as a combining function. However, on the CIRR dataset, the situation differs. When comparing with the performance of the image encoder fine-tuning, using the element-wise sum to combine the query features results in comparable global metrics, but significantly improved fine-grained *subset* metrics. In contrast, when utilizing the Combiner network, we observe a reduction in the gaps within the *subset* metrics, while achieving a greater improvement in the global metrics. We achieve the best results on both

			Shirt		Dress		Toptee		Average	
CF	**IFT**	**TFT**	*R*@10	*R*@50	*R*@10	*R*@50	*R*@10	*R*@50	*R*@10	*R*@50
Sum	✗	✗	19.53	35.57	17.70	36.29	21.88	42.93	19.70	38.26
	✓	✗	30.08	52.94	29.10	52.01	34.42	57.62	31.20	54.19
	✗	✓	32.29	53.73	27.76	52.31	35.14	60.12	31.73	55.39
	✓	✓	38.67	59.42	35.99	62.22	43.35	67.52	39.34	63.05
Combiner	✗	✗	31.85	52.50	27.22	50.62	33.81	57.57	30.96	53.56
	✓	✗	34.30	55.79	32.47	55.18	38.45	62.36	35.07	57.78
	✗	✓	35.87	57.21	31.43	54.98	38.20	63.22	35.16	58.47
	✓	✓	**39.87**	**60.84**	**37.67**	**63.16**	**44.88**	**68.59**	**40.80**	**64.20**

Table 1. Recall at K on the FashionIQ validation set while varying the combining function and the modality of CLIP fine-tuning. We denote **IFT** (image encoder fine-tuning) and **TFT** (text encoder fine-tuning) to represent whether the image encoder or the text encoder is fine-tuned in the first stage. **CF** (combining function) indicates the function used to combine the query features. We highlight the best scores in bold and underline the second-best scores.

			Recall@K				$\mathbf{R_{subset}}$@K		
CF	**IFT**	**TFT**	$K=1$	$K=5$	$K=10$	$K=50$	$K=1$	$K=2$	$K=3$
Sum	✗	✗	21.38	50.85	64.00	87.23	54.48	76.01	87.16
	✓	✗	31.67	66.08	79.36	95.38	58.12	78.42	89.78
	✗	✓	32.72	66.63	79.22	94.86	67.21	86.00	93.81
	✓	✓	40.97	74.70	85.51	96.94	68.81	86.96	93.90
Combiner	✗	✗	31.26	64.79	77.71	95.31	61.56	81.08	91.12
	✓	✗	34.01	69.07	81.77	95.72	62.78	81.80	91.41
	✗	✓	36.86	71.32	82.32	96.24	68.28	86.51	94.14
	✓	✓	**42.05**	**76.13**	**86.51**	**97.49**	**70.15**	**87.18**	**94.40**

Table 2. Recall at K on the CIRR validation set while varying the combining function and the modality of CLIP fine-tuning. We denote **IFT** (image encoder fine-tuning) and **TFT** (text encoder fine-tuning) to represent whether the image encoder or the text encoder is fine-tuned in the first stage. **CF** (combining function) indicates the function used to combine the query features. We highlight the best scores in bold and underline the second-best scores.

datasets when we fine-tune both encoders. The element-wise sum of the fine-tuned features outperforms the performance of the out-of-the-box features combined with the trained Combiner network by a significant margin. Moreover, when we combine the query features with the Combiner network, the performances further improve. It is worth highlighting that when utilizing the Combiner as a combining function, the improvement achieved by fine-tuning both encoders over the out-of-the-box CLIP features is the arithmetic sum of the improvements obtained by fine-tuning either the image or the text encoder.

Given this last observation and all the other results, we formulate the hypothesis that the fine-tuning of the image and the text encoder learn different and complementary information that improves performances differently. We conjecture that the fine-tuning of the image encoder adapts the image manifold to the domain of the data (e.g., the fashion domain for the FashionIQ dataset). On the contrary, the fine-tuning of the text-encoder adapts the text embedding space to the task of composed image retrieval by transforming textual features into displacement vectors within the image embedding space. In support of this conjecture, we highlight the difference in performances between the global metrics and *subset* metrics on the CIRR dataset when comparing the image

	Shirt		Dress		Toptee		Average	
Model	*R*@10	*R*@50	*R*@10	*R*@50	*R*@10	*R*@50	*R*@10	*R*@50
Element-wise sum	38.67	59.42	35.99	62.22	43.35	67.52	39.34	63.05
Convex combination	<u>39.45</u>	60.16	36.44	62.57	44.05	67.87	39.98	63.53
W/o convex combination	31.40	55.64	35.94	61.03	40.29	64.97	35.87	60.55
Static skip	39.00	<u>60.54</u>	<u>36.99</u>	<u>63.11</u>	<u>44.26</u>	<u>68.23</u>	<u>40.08</u>	<u>63.96</u>
Proposed Combiner	**39.87**	**60.84**	**37.67**	**63.16**	**44.88**	**68.59**	**40.80**	**64.20**

Table 3. Recall at K on the FashionIQ validation set, with variations on the Combiner architecture. We highlight the best scores in bold and underline the second-best scores.

	Recall@K				R_{subset}@K		
Model	$K = 1$	$K = 5$	$K = 10$	$K = 50$	$K = 1$	$K = 2$	$K = 3$
Element-wise sum	40.97	74.70	85.51	96.94	68.81	86.96	93.90
Convex combination	41.11	75.56	85.55	97.44	**70.46**	87.08	<u>94.33</u>
W/o convex combination	36.98	72.06	82.83	96.67	65.53	84.74	93.06
Static skip	<u>41.88</u>	<u>75.87</u>	<u>86.20</u>	<u>97.46</u>	69.89	**87.35**	94.21
Proposed Combiner	**42.05**	**76.13**	**86.51**	**97.49**	<u>70.15</u>	<u>87.18</u>	**94.40**

Table 4. Recall at K on the CIRR validation set, with variations on the Combiner architecture. We highlight the best scores in bold and underline the second-best scores.

and the text encoder fine-tuning using the element-wise sum as a combining function (second and third row in Table 2). We note that in the global metrics, where the domain of the images is diverse, the performance differences between the two experiments approach zero. Conversely, in the *subset* metrics, where the visual differences among the images are low, the image fine-tuning is not capable of capturing the fine-grained differences making the textual information more discriminative and thus making the fine-tuning of the text encoder perform better. The experiments described in Section 4.8 provide additional confirmation of our intuition.

4.4 Combiner ablation study

In this section, we present a set of experiments with ablations and variations of the proposed Combiner network. We perform all the experiments using the fine-tuned RN-50 CLIP model. We train all the Combiner networks using a batch size of 4096 and a learning rate of $2e-5$.

Given the proposed Combiner network illustrated in Fig. 4, we denote the outputs of the first branch (1) as λ and $1-\lambda$, while the output of the second branch (2) as v. The output features of the proposed Combiner are: $\overline{\phi_q} = (1-\lambda) * \overline{\psi_I}(I_q) + \lambda * \overline{\psi_T}(T_q) + v$.

To evaluate each component of the proposed design, we tested the following variations:

- **Element-wise sum**: fine-tuned image and text features are summed: $\overline{\phi_q} = \overline{\psi_I}(I_q) + \overline{\psi_T}(T_q)$
- **Convex combination**: only convex combination of image and text features, i.e. the model without the mixture contribution of text and image: $\overline{\phi_q} = (1-\lambda) * \overline{\psi_I}(I_q) + \lambda * \overline{\psi_T}(T_q)$
- **W/o convex combination**: only the mixture contribution of text and image, i.e the model without the convex combination of text and image features: $\overline{\phi_q} = v$
- **Static skip**: the convex coefficients are statically set to 0.5: $\overline{\phi_q} = 0.5 * \overline{\psi_I}(I_q) + 0.5 * \overline{\psi_T}(T_q) + v$
- **Proposed Combiner**: the Combiner architecture illustrated in Fig. 4.

			Shirt		Dress		Toptee		Average	
Approach	**IFT**	**TFT**	*R*@10	*R*@50	*R*@10	*R*@50	*R*@10	*R*@50	*R*@10	*R*@50
End-to-end	✓	✗	31.79	53.14	30.29	53.49	33.55	59.15	31.87	55.26
	✗	✓	33.02	54.41	30.19	53.64	35.90	61.60	33.03	56.55
	✓	✓	<u>37.29</u>	<u>59.02</u>	<u>34.65</u>	<u>60.83</u>	<u>41.20</u>	<u>65.99</u>	<u>37.71</u>	<u>61.95</u>
Two-stages	✓	✗	34.30	55.79	32.47	55.18	38.45	62.36	35.07	57.78
	✗	✓	35.87	57.21	31.43	54.98	38.20	63.22	35.16	58.47
	✓	✓	**39.87**	**60.84**	**37.67**	**63.16**	**44.88**	**68.59**	**40.80**	**64.20**

Table 5. Recall at K on the FashionIQ validation set employing either the two-stage or the end-to-end approach. We denote **IFT** (image encoder fine-tuning) and **TFT** (text encoder fine-tuning) to represent whether the image encoder or the text encoder is fine-tuned in the first stage. We highlight the best scores in bold and underline the second-best scores.

We report the results for each variation in Table 3 for the FashionIQ dataset and in Table 4 for the CIRR dataset. The element-wise sum of the fine-tuned features serves as a solid starting point. As shown in section 4.3, the task-oriented fine-tuning process is highly effective and results in significant improvements over the out-of-the-box features on both datasets. The convex combination baseline, which dynamically computes text and image convex coefficients for greater adaptability to the query, achieves a slight improvement over the element-wise sum of the features. Notably, when we remove the text and image convex combination, we observe a significant drop in performance compared to the proposed Combiner. This emphasizes the importance of the text and image convex combination in achieving good performance. This result demonstrates that allowing the Combiner network to learn the residual from the element-wise sum (or its generalization, the convex combination) leads to a considerable improvement in performance. This outcome is expected because without the contribution of the image-text convex combination, the effectiveness of the first-stage training, which aims to enhance the additivity properties of the embedding spaces, is compromised. It is worth noting that setting the convex coefficients statically to 0.5 leads to a slight decrease in performance, which is attributed to the greater adaptability of the dynamically computed coefficients.

Our experiments demonstrate the crucial role of the Combiner architecture in effectively exploiting the full potential of the additive embedding spaces constructed during the first stage of training. By enabling the network to learn the residual from the dynamically computed convex combination, we observe significant performance improvements.

4.5 Analysis of Two-Stage vs. End-to-End approach

In order to explain why a two-stage training method, where the CLIP encoder and Combiner are trained separately, in contrast to an end-to-end approach, we perform an experiment where we compare the two settings on both CIRR and FashionIQ datasets. First, we train end-to-end by fine-tuning CLIP encoders while training the Combiner network simultaneously. Then, we followed the proposed two-stage approach. In both settings, we also enable fine-tuning of the textual or image encoders separately and jointly. In all the experiments in this section, we use the RN-50 CLIP model.

We present the results in Table 5 for the FashionIQ dataset and in Table 6 for the CIRR dataset. Remarkably, the two-stage approach consistently outperforms the end-to-end one on both datasets. These superior results remain consistent even when varying the fine-tuning modality.

The results validate the effectiveness of constructing an embedding space with robust additivity properties before combining the features using a non-linear function. We hypothesize that when training the Combiner network simultaneously with the CLIP encoders, the entire system struggles to effectively learn the additive

			Recall@K				R_{subset}@K		
Approach	**IFT**	**TFT**	$K = 1$	$K = 5$	$K = 10$	$K = 50$	$K = 1$	$K = 2$	$K = 3$
End-to-end	✓	✗	33.19	67.01	79.83	95.52	58.90	79.33	90.28
	✗	✓	33.58	67.59	79.57	95.28	67.37	85.58	93.44
	✓	✓	<u>40.03</u>	<u>74.09</u>	<u>85.14</u>	<u>97.12</u>	68.14	86.06	93.64
Two-stages	✓	✗	34.01	69.07	81.77	95.72	62.78	81.80	91.41
	✗	✓	36.86	71.32	82.32	96.24	<u>68.28</u>	<u>86.51</u>	<u>94.14</u>
	✓	✓	**42.05**	**76.13**	**86.51**	**97.49**	**70.15**	**87.18**	**94.40**

Table 6. Recall at K on the CIRR validation set employing either the two-stage or the end-to-end approach. We denote **IFT** (image encoder fine-tuning) and **TFT** (text encoder fine-tuning) to represent whether the image encoder or the text encoder is fine-tuned in the first stage. We highlight the best scores in bold and underline the second-best scores.

embedding spaces and the non-linear combining function in a cohesive manner. As a result, this limitation negatively impacts the overall performance, leading to suboptimal outcomes.

4.6 Preprocess upshot

In this section, we show how the proposed preprocess pipeline, described in Section 3.3, contributes to further improving performance. We compare the proposed preprocess with two other methods: the standard CLIP preprocess pipeline, primarily consisting of resize and center crop operations, and the Square preprocess, which involves applying a square zero-pad to the image before resizing and center cropping. The comparison among the different preprocess techniques is presented in Table 7 for the FashionIQ dataset and Table 8 for the CIRR dataset.

On the FashionIQ dataset, the improvement obtained using the proposed preprocess pipeline over the standard one is substantial in the *Dress* category and noticeable in the *Toptee* category. Conversely, the square pad preprocess technique achieves comparable performance to the proposed one in the *Dress* and *Toptee* categories while suffering a performance deficit in the *Shirt* category. Overall, we observe a correlation between the difference in performance among the methods and the number of images with a high aspect ratio, as depicted in Figure 5. In other words, when dealing with images with a high aspect ratio, it is preferable to pad them to avoid losing crucial portions of the image during the center crop operation. On the other hand, when images have a low aspect ratio, it is more effective not to reduce the usable portion of the image with padding. The proposed preprocess pipeline achieves the best performance by effectively adapting to the aspect ratio of each image. On the CIRR dataset, we observe that the proposed preprocess significantly improves performance compared to the standard CLIP and the square preprocess. The performance gain is particularly significant in low-rank recall measures, where the importance of every lost detail is crucial.

4.7 Comparison with SotA

We compare the proposed method with state-of-the-art approaches on two standard and challenging datasets. To ensure a fair comparison, we follow the standard experimental settings of the two datasets [37, 51]. Unless specifically mentioned, we report the metrics for each method as documented in the official papers, and we refer to those papers for more comprehensive details about the individual approaches.

Table 9 reports the comparison between the proposed method and other state-of-the-art approaches. We divide the table into two sections: the upper section includes methods that are not directly comparable to our approach. These approaches either do not utilize a pre-trained textual encoder [8, 29, 31, 45, 50, 52] or, in the case of TRACE [27], they use BERT [15] as a pre-trained textual encoder but do not update its weights. It is important to note that even when a competitor [8, 29, 45] utilizes the GloVe word embedding [40], we do not consider

		Shirt		Dress		Toptee		Average	
CF	**Preprocess**	*R*@10	*R*@50	*R*@10	*R*@50	*R*@10	*R*@50	*R*@10	*R*@50
Sum	**Standard**	37.64	59.76	33.42	59.84	40.90	66.80	37.32	62.13
	Square	37.09	58.52	35.94	62.03	42.53	66.29	38.52	62.28
	Proposed	38.67	59.42	35.99	62.22	43.35	67.52	39.34	63.05
Combiner	**Standard**	<u>39.40</u>	**61.33**	35.25	60.44	43.95	67.72	39.53	63.16
	Square	38.71	60.21	**37.97**	<u>62.86</u>	<u>44.12</u>	<u>68.03</u>	<u>40.26</u>	<u>63.70</u>
	Proposed	**39.87**	<u>60.84</u>	<u>37.67</u>	**63.16**	**44.88**	**68.59**	**40.80**	**64.20**

Table 7. Recall at K on FashionIQ validation set varying the combining function and the preprocessing pipeline used. **CF** (combining function) indicates the function used to combine the query features. We highlight the best scores in bold and underline the second-best scores.

		Recall@K				R_{subset}@K		
CF	**Preprocess**	$K = 1$	$K = 5$	$K = 10$	$K = 50$	$K = 1$	$K = 2$	$K = 3$
Sum	**Standard**	39.51	74.00	84.72	<u>97.20</u>	68.36	86.15	94.26
	Square	41.26	74.34	85.00	96.84	69.15	85.89	93.90
	Proposed	40.97	74.70	<u>85.51</u>	96.94	68.81	<u>86.96</u>	93.90
Combiner	**Standard**	40.08	74.15	84.67	<u>97.20</u>	69.53	86.27	**94.45**
	Square	<u>41.95</u>	<u>74.96</u>	85.24	96.58	**70.65**	86.67	94.24
	Proposed	**42.05**	**76.13**	**86.51**	**97.49**	<u>70.15</u>	**87.18**	<u>94.40</u>

Table 8. Recall at K on CIRR validation set varying the combining function and the preprocessing pipeline used. **CF** (combining function) indicates the function used to combine the query features. We highlight the best scores in bold and underline the second-best scores.

their textual encoder as pre-trained. All the methods in this section rely on a ResNet model pre-trained on the ImageNet dataset [43] and fine-tuned during training. We include the results of these methods to provide a more comprehensive discussion. The lower section of Table 9 reports methods that are directly comparable to ours: they rely on both pre-trained visual and language models updating all the weights of both backbones during training. CIRRPLANT [37] relies on the OSCAR pretrained model as a textual backbone, while [16, 21, 26] rely on the pre-trained BERT model. It is worth mentioning that FashionViL is a fashion-oriented approach that carries out a large-scale pre-training for learning V+L representation in the fashion domain. For this reason, it is not surprising that it exhibits strong performances in a fashion dataset such as FashionIQ. When considering the RN50-based method, the proposed approach outperforms the competitors by improving up to 9% in average R@10 and 7% in average R@50 compared to the best-performing competitor, FashionViL, when using the same visual backbone architecture. Our method demonstrates the highest recall across all categories, with a particularly significant margin observed in the Shirt category. When considering the larger RN-50x4-based model, we observe an improvement ranging from 2% to 4% in all categories compared to the smaller backbone. This result demonstrates that our approach scales well when using larger and heavier VL models.

In Table 10, we report a comparison between the proposed method and other state-of-the-art approaches. As for FashionIQ, the upper section of the table reports methods that are not directly comparable with the proposed one: they do not utilize a pre-trained textual encoder. As the visual backbone, they employ a ResNet-based model, which is pre-trained on ImageNet and fine-tuned during training. The lower section of the table includes directly comparable methods, such as MAAF, which utilizes BERT as a text encoder, and CIRPLANT, which relies on the

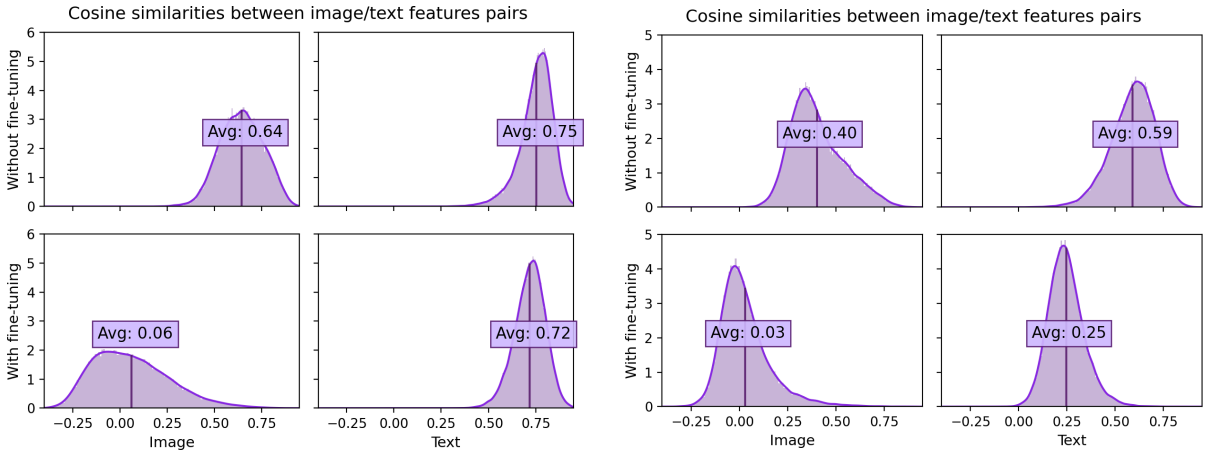
	Encoder		Shirt		Dress		Toptee		Average	
Method	Visual	Textual	*R*@10	*R*@50	*R*@10	*R*@50	*R*@10	*R*@50	*R*@10	*R*@50
TRACE [27]	RN-50	BERT [15]	20.80	40.80	22.70	44.91	24.22	49.80	22.57	46.19
VAL [8]	RN-50	LSTM(GloVe) [23]	22.38	44.15	22.53	44.00	27.53	51.68	24.15	46.61
CurlingNet [52]	RN-152	biGRU [10]	21.45	44.56	26.15	53.24	30.12	55.23	25.90	51.01
RTIC-GCN [45]	RN-50	LSTM(GloVe)	23.79	47.25	29.15	54.04	31.61	57.98	28.18	53.09
CoSMo [31]	RN-50	LSTM	24.90	49.18	25.64	50.30	29.21	57.46	26.58	52.31
DCNet [29]	RN-50	Conv1D(GloVe)	23.95	47.30	28.95	56.07	30.44	58.29	27.78	53.89
CLVC-Net [50]	RN-50	LSTM	28.75	54.76	29.85	56.47	33.50	64.00	30.70	58.41
CIRPLANT [37]	RN-152	OSCAR [34]	17.53	38.81	17.45	40.41	21.64	45.38	18.87	41.53
MAAF [16]	RN-50	BERT	18.55	37.63	18.59	39.66	23.05	45.95	20.06	41.08
SAC [26]	RN-50	BERT	28.02	51.86	26.52	51.01	32.70	61.23	29.08	54.70
FashionViL [21]	RN-50	BERT	25.17	50.39	33.47	59.94	34.98	60.79	31.20	57.04
Ours	RN-50	Transformer	<u>39.87</u>	<u>60.84</u>	<u>37.67</u>	<u>63.16</u>	<u>44.88</u>	<u>68.59</u>	<u>40.80</u>	<u>64.20</u>
Ours	RN-50x4	Transformer	**44.41**	**65.26**	**39.46**	**64.55**	**47.48**	**70.98**	**43.78**	**66.93**

Table 9. Comparison between our method and current state-of-the-art models on the Fashion-IQ validation set. We highlight the best scores in bold and underline the second-best scores. The upper section of the table presents methods that are not directly comparable to our proposed approach, as they either do not utilize a pre-trained textual encoder or do not update its weights. "RN" stands for ResNet.

	Encoder		**Recall@K**				**R_{subset}@K**		
Method	Visual	Textual	$K=1$	$K=5$	$K=10$	$K=50$	$K=1$	$K=2$	$K=3$
TIRG† [48]	RN-18	LSTM	14.61	48.37	64.08	90.03	22.67	44.97	65.14
TIRG+LastConv† [48]	RN-18	LSTM	11.04	35.68	51.27	83.29	23.82	45.65	64.55
MAAF† [16]	RN-50	LSTM	10.31	33.03	48.30	80.06	21.05	41.81	61.60
MAAF–IT† [16]	RN-50	LSTM	9.90	32.86	48.83	80.27	21.17	42.04	60.91
MAAF–RP† [16]	RN-50	LSTM	10.22	33.32	48.68	81.84	21.41	42.17	61.60
ARTEMIS [14]	RN-152	biGRU	16.96	46.10	61.31	87.73	39.99	62.20	75.67
MAAF† [16]	RN-50	BERT	10.12	33.10	48.01	80.57	22.04	42.41	62.14
CIRPLANT† [37]	RN-152	OSCAR	19.55	52.55	68.39	92.38	39.20	63.03	79.49
Ours	RN-50	Transformer	<u>40.91</u>	<u>74.53</u>	<u>84.77</u>	<u>97.35</u>	<u>70.22</u>	<u>87.80</u>	<u>94.46</u>
Ours	RN-50x4	Transformer	**44.82**	**77.04**	**86.65**	**97.90**	**73.16**	**88.84**	**95.59**

Table 10. Comparison between our method and current state-of-the-art models on the CIRR test set. We highlight the best scores in bold and underline the second-best scores.. † denotes results cited from [37]. The upper section of the table presents methods that are not directly comparable to our proposed approach, as they either do not utilize a pre-trained textual encoder or do not update its weights. In the lower section, we report methods that are directly comparable to our approach. "RN" stands for ResNet.

pre-trained Vision-Language model OSCAR. The results presented in Table 10 are obtained through the official evaluation server. Our approach consistently outperforms the competitors by a significant margin, particularly in low-rank recall measures, where we notice an improvement of approximately 20% in R@1 when using the RN50 visual backbone. When considering the larger RN-50x4 model, we observe improvements ranging from 3% in low-rank recall metrics to 1% as the recall rank increases.



(a) FashionIQ dataset (b) CIRR dataset

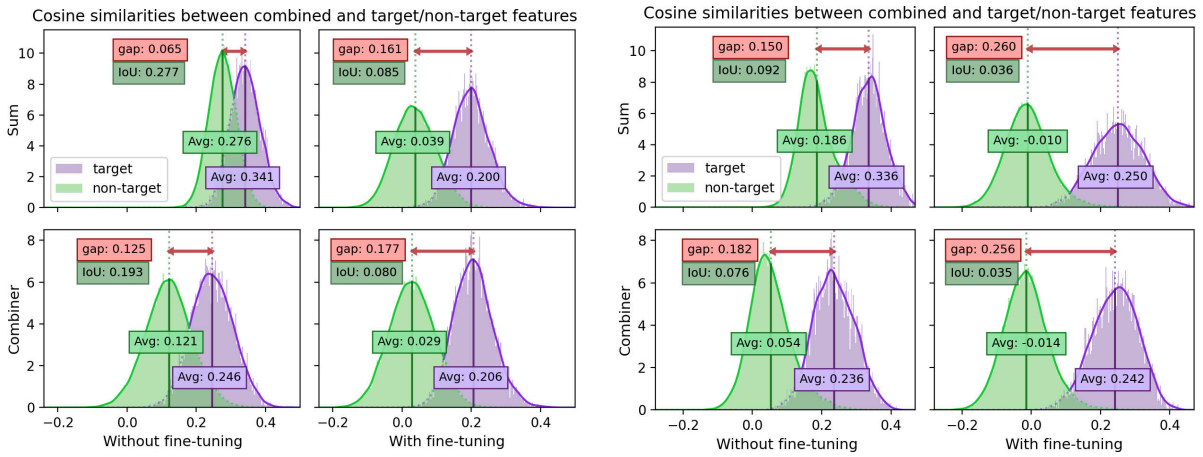
Fig. 7. Histograms of cosine similarities between image/text feature pairs. The x-axis represents the cosine similarities. The y-axis represents the (normalized) number of pairs. In the top-line plots, we have used the out-of-the-box CLIP model. In the bottom line, we have used the model fine-tuned during the first stage of training. In the left-side plots, we compare the image features. In the right ones, we compare the text features. The histograms are normalized such that the area under each curve integrates to 1.

4.8 Feature distribution study

The experiments in this section aim to provide intuition on how the feature distribution in the embedding spaces affects the retrieval performances. All the experiments were carried out on the validation sets using the RN-50 model. We are going to present two different sets of experiments that have slightly different purposes. The first set aims to investigate how the image and text features are distributed in the embedding spaces, while the second one explores how the distribution of the features affects retrieval performance.

To investigate how the features distribute in the embedding spaces, we followed [35] and calculate pairwise similarities among them. If the features occupy the embedding space uniformly, their similarities will be lower. Throughout all experiments, due to the quadratic growth of possible pairs, we compute the similarities between 50K randomly sampled pairs. Figure 7 shows the histograms of the features pairwise similarities on both FashionIQ and CIRR datasets. First of all, we can notice that due to the broader domain of CIRR, on such a dataset, both the image and text features similarities are higher when compared to FashionIQ. On both datasets, fine-tuning the image encoder leads to a drastic reduction in the average similarity of the visual features and, thus, to much more efficient use of the embedding space during retrieval. This fact confirms our hypothesis (Section 4.3) that fine-tuning the image encoder adapts the image manifold to the data domain. The fine-tuning of the text encoder leads to a lower reduction in the average pairwise similarity of textual features (almost negligible in FashionIQ) than that observed in visual ones. We suppose that efficient use of the image embedding space is far more critical than efficient use of the textual space since the retrieval is carried out in the image space. In all the experiments, we observe that the fine-tuning of CLIP encoders contributes to reducing the *cone effect*: “the effective embedding space is restricted to a narrow cone for trained models and models with random weights" [35].

The previous experiments demonstrate how the two-stage approach proposed in this study affects the textual and visual CLIP embedding spaces. However, these experiments do not clarify why this increased utilization of embedding space can improve the retrieval process. We conducted additional experiments to investigate the impact of this embedding space reshaping on the image retrieval task. We compute and compare the cosine similarities (the distance function used in the retrieval) between the combined and the index image features.



(a) FashionIQ dataset (b) CIRR dataset

Fig. 8. Histograms of the cosine similarities between combined and target/non-target feature pairs. The x-axis represents the cosine similarities. The y-axis represents the (normalized) number of pairs. In green●: cosine similarities between combined and non-target index features. In violet●: cosine similarities between combined and target features. In red●: similarity gap between combined-target and combined-non target features. In dark green●: intersection over union area between the two histograms. In the top-line plots, we have used the simple sum as a combining function. In the bottom line ones, we have used the Combiner network. In the left-side plots, we used the out-of-the-box CLIP model. In the right ones, we used the model fine-tuned during the first stage of the training. The histograms are normalized such that the area under each curve integrates to 1.

Specifically, we perform two distinct computations: in the first one, we compute the similarity between the combined features and the target image features belonging to the same query triplet. In the second one, we compute the similarity between the combined features and random image features that differ from the target ones. Given a query, we will refer to the images that differ from the target as *non-target images*. We compare each combined feature with ten non-target image features to reduce the variance.

Figure 8 emphasizes the similarity gaps between the combined and target/non-target features. On both FashionIQ and CIRR datasets, we notice that the element-wise sum of out-of-the-box CLIP features achieves the highest average combined-target features similarity. During both the fine-tuning and the Combiner network training stages, the contrastive training increases the cosine distances between the combined and non-target features instead of increasing their similarity to the target features. By observing both Fig. 8a and Fig. 8b and the corresponding retrieval results in Table 1 and Table 2, we argue that, in these two datasets, the retrieval performances are highly correlated with the similarity gap between combined-target and combined-non target features (displayed as the red arrows in Fig. 8) and with the size of intersection area between the histograms (the smaller the intersection area, the smaller the retrieval errors will be). On the contrary, the absolute value of the combined-target similarity does not seem to be of great importance.

The two sets of experiments highlight different but strongly related aspects. The first set shows that fine-tuning both CLIP encoders leads to more efficient use of the embedding spaces. In the second set, we prove that the increased occupation of the image space helps to "move away" the combined features from the non-target features.

4.9 Qualitative results

To obtain a clearer understanding of which parts of the images the system considers most important during retrieval, we conducted qualitative experiments using the GradCAM technique [44]. Instead of computing

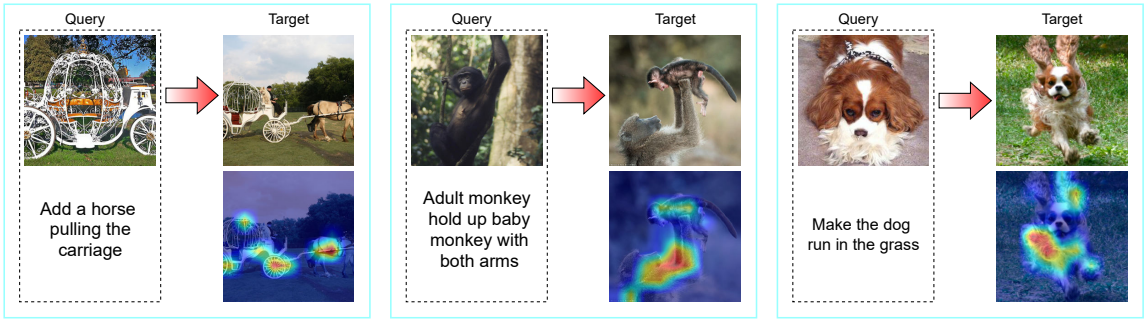


Fig. 9. Examples of GradCAM visualization on CIRR dataset computing the gradients with respect to the Combiner output.

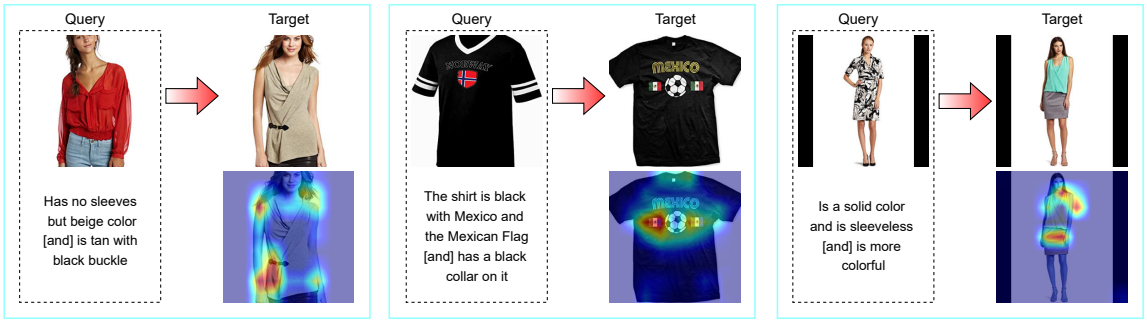


Fig. 10. Examples of GradCAM visualization on FashionIQ computing the gradients with respect to the Combiner output.

gradients versus an output class, we compute gradients with respect to the combined features, which summarize both the visual and textual content of the image and caption, using the GradCAM technique. This approach makes each heat map generated by GradCAM dependent on the reference image and its relative caption, simulating the retrieval process. We use the last convolutional layer of CLIP's image encoder as the saliency layer.

In Fig. 9 and in Fig. 10 are displayed some examples of the above-described visualization technique. The system is capable of attending to a wide range of concepts, such as style and color changes for the fashion dataset and behavior modification for the CIRR dataset, as we can notice from the experiments with the GradCAM technique. For instance, in Fig. 9, the system attends to the carriage and horse in the first example, the pose of the holding monkey and the baby monkey in the second example, and the pose of the dog in the third example. In Fig. 10, the system attends to the arms and shoulders of the person when the conditioning text referred to the sleeves of the dress and to the logo of the shirt when it was requested to change the Norwegian flag into a Mexican one.

Finally, we complete the qualitative analysis of our approach by presenting examples of multimodal queries and their corresponding results on both datasets in Fig. 12 and Fig. 11. In FashionIQ, the correct result is returned most of the time in the first three results, while in CIRR, it is returned in the top-5 global and top-3 subset results. Interestingly, the excellent performance of the proposed system let us notice an issue with the FashionIQ dataset: from these examples, we can see that in the FashionIQ dataset, the existence of many false negatives is a real issue that can harm both the results and the training process; examining the first four and the last queries, we can observe that several results returned in the first positions are corresponding to the conditioning text, although only one of them is marked as such. E.g. in the first query, where several dresses have light floral patterns and bright colors, similarly, the first three results for the shirts should be considered correct. We can also see that in the CIRR dataset, the domain of the images is wider compared to FashionIQ, and the problem of the false negatives is a minor issue.

Fig. 11. Qualitative results for the FashionIQ dataset. We highlight with a green border when the retrieved image is labeled as ground truth for the given query.

5 CONCLUSIONS

In this work, we propose a novel task-oriented fine-tuning scheme to adapt vision-language models for the composed image retrieval task. The primary goal of this fine-tuning is to address the mismatch between the large-scale pre-training of CLIP and the downstream task, thereby enhancing the additivity properties of the embedding spaces. We then propose a two-stage approach that combines fine-tuning with the training of a carefully crafted Combiner network, enabling the meaningful fusion of the fine-tuned multimodal features. To further enhance performance, we introduce a novel pre-processing padding method, which, as demonstrated in the ablation studies, improves performance on datasets with images of varying aspect ratios. We perform experiments on the challenging fashion dataset FashionIQ and the recently presented CIRR dataset. Experiments on both datasets show that our approach outperforms state-of-the-art methods by a significant margin. We also perform qualitative experiments to explain how our approach works. These experiments investigate the impact of the proposed approach on the feature distribution in the embedding spaces and how the reshaping of such embedding spaces influences retrieval performance. Additionally, we conduct visualization experiments using the gradCAM technique.

ACKNOWLEDGMENTS

This work was partially supported by the European Commission under European Horizon 2020 Programme, grant number 101004545 - ReInHerit.

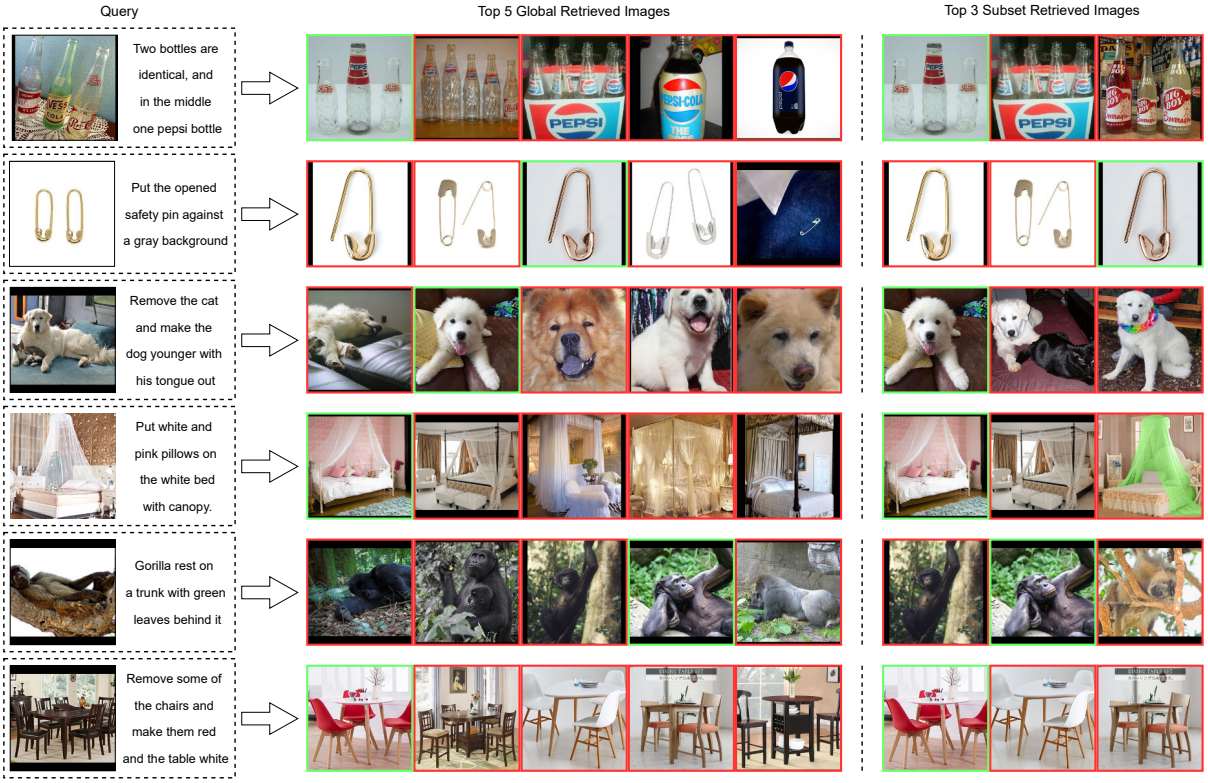


Fig. 12. Qualitative results for the CIRR dataset. We highlight with a green border when the retrieved image is labeled as ground truth for the given query.

REFERENCES

[1] Sandhini Agarwal, Gretchen Krueger, Jack Clark, Alec Radford, Jong Wook Kim, and Miles Brundage. 2021. Evaluating clip: towards characterization of broader capabilities and downstream implications. *arXiv preprint arXiv:2108.02818* (2021).

[2] Jamil Ahmad, Khan Muhammad, Sambit Bakshi, and Sung Wook Baik. 2018. Object-oriented convolutional features for fine-grained image retrieval in large surveillance datasets. *Future Generation Computer Systems* 81 (2018), 314–330.

[3] Muhammad Umer Anwaar, Egor Labintcev, and Martin Kleinsteuber. 2021. Compositional Learning of Image-Text Query for Image Retrieval. In *Proc. of IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 1140–1149.

[4] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. 2022. Exploiting CLIP-Based Multi-modal Approach for Artwork Classification and Retrieval. In *The Future of Heritage Science and Technologies: ICT and Digital Heritage: Third International Conference, Florence Heri-Tech 2022, Florence, Italy, May 16–18, 2022, Proceedings*. Springer, 140–149.

[5] Imon Banerjee, Camille Kurtz, Alon Edward Devorah, Bao Do, Daniel L Rubin, and Christopher F Beaulieu. 2018. Relevance feedback for enhancing content based image retrieval and automatic prediction of semantic image features: Application to bone tumor radiographs. *Journal of biomedical informatics* 84 (2018), 123–135.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[7] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174* (2016).

[8] Yanbei Chen, Shaogang Gong, and Loris Bazzani. 2020. Image Search With Text Feedback by Visiolinguistic Attention Learning. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*.

[9] Ruizhe Cheng, Bichen Wu, Peizhao Zhang, Peter Vajda, and Joseph E. Gonzalez. 2021. Data-Efficient Language-Supervised Zero-Shot Learning With Self-Distillation. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 3119–3124.

[10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[11] Mario GCA Cimino, Federico A Galatolo, and Gigliola Vaglini. 2021. Generating Images from Caption and Vice Versa via CLIP-Guided Generative Latent Space Search. In *Proceedings of the International Conference on Image Processing and Vision Engineering*. 166–174.

[12] Claudia Companioni-Brito, Zygred Mariano-Calibjio, Mohamed Elawady, and Sule Yildirim. 2018. Mobile-Based Painting Photo Retrieval Using Combined Features. In *Proc. of International Conference on Image Analysis and Recognition (ICIAR)*, Vol. 10882. Springer, 278.

[13] Marcos V Conde and Kerem Turgutlu. 2021. CLIP-Art: Contrastive Pre-Training for Fine-Grained Art Classification. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 3956–3960.

[14] Ginger Delmas, Rafael S Rezende, Gabriela Csurka, and Diane Larlus. 2021. ARTEMIS: Attention-based Retrieval with Text-Explicit Matching and Implicit Similarity. In *International Conference on Learning Representations*.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

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

[17] Xiao Dong, Xunlin Zhan, Yangxin Wu, Yunchao Wei, Xiaoyong Wei, Minlong Lu, and Xiaodan Liang. 2021. M5product: A multi-modal pretraining benchmark for e-commercial product downstream tasks. *arXiv preprint arXiv:2109.04275* (2021).

[18] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. 2021. CLIP2Video: Mastering Video-Text Retrieval via Image CLIP. *arXiv preprint arXiv:2106.11097* (2021).

[19] Xiaoxiao Guo, Hui Wu, Yu Cheng, Steven Rennie, Gerald Tesauro, and Rogerio Feris. 2018. Dialog-based interactive image retrieval. *Advances in neural information processing systems* 31 (2018).

[20] Xintong Han, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis. 2017. Automatic spatially-aware fashion concept discovery. In *Proceedings of the IEEE international conference on computer vision*. 1463–1471.

[21] Xiao Han, Licheng Yu, Xiatian Zhu, Li Zhang, Yi-Zhe Song, and Tao Xiang. 2022. Fashionvil: Fashion-focused vision-and-language representation learning. In *European Conference on Computer Vision*. Springer, 634–651.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[23] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[24] Bogdan Ionescu, Henning Müller, Renaud Péteri, Yashin Dicente Cid, Vitali Liauchuk, Vassili Kovalev, Dzmitri Klimuk, Aleh Tarasau, Asma Ben Abacha, Sadid A Hasan, et al. 2019. ImageCLEF 2019: Multimedia retrieval in medicine, lifelogging, security and nature. In *Proc. of International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF)*. Springer, 358–386.

[25] Bogdan Ionescu, Henning Müller, Renaud Péteri, Duc-Tien Dang-Nguyen, Liting Zhou, Luca Piras, Michael Riegler, Pål Halvorsen, Minh-Triet Tran, Mathias Lux, et al. 2020. ImageCLEF 2020: Multimedia retrieval in lifelogging, medical, nature, and internet applications. *Advances in Information Retrieval* 12036 (2020), 533.

[26] Surgan Jandial, Pinkesh Badjatiya, Pranit Chawla, Ayush Chopra, Mausoom Sarkar, and Balaji Krishnamurthy. 2022. SAC: Semantic Attention Composition for Text-Conditioned Image Retrieval. In *Proc. of IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 4021–4030.

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

[28] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proc. of International Conference on Machine Learning (ICML)*.

[29] Jongseok Kim, Youngjae Yu, Hoeseong Kim, and Gunhee Kim. 2021. Dual Compositional Learning in Interactive Image Retrieval. In *Proc. of AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 35. 1771–1779. https://ojs.aaai.org/index.php/AAAI/article/view/16271

[30] Adriana Kovashka, Devi Parikh, and Kristen Grauman. 2015. WhittleSearch: Interactive Image Search with Relative Attribute Feedback. *International Journal of Computer Vision (IJCV)* 115, 2 (Apr 2015), 185–210. https://doi.org/10.1007/s11263-015-0814-0

[31] Seungmin Lee, Dongwan Kim, and Bohyung Han. 2021. CoSMo: Content-Style Modulation for Image Retrieval With Text Feedback. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 802–812.

[32] Manling Li, Ruochen Xu, Shuohang Wang, Luowei Zhou, Xudong Lin, Chenguang Zhu, Michael Zeng, Heng Ji, and Shih-Fu Chang. 2022. Clip-event: Connecting text and images with event structures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16420–16429.

[33] Xiaoqing Li, Jiansheng Yang, and Jinwen Ma. 2021. Recent developments of content-based image retrieval (CBIR). *Neurocomputing* 452 (2021), 675–689. https://doi.org/10.1016/j.neucom.2020.07.139

[34] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*. Springer, 121–137.

[35] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. 2022. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems* 35 (2022), 17612–17625.

[36] Yating Liu and Yan Lu. 2021. Multi-grained Fusion for Conditional Image Retrieval. In *Proc. of International Conference on Multimedia Modeling (MMM)*. Springer International Publishing, Cham, 315–327.

[37] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. 2021. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2125–2134.

[38] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Bkg6RiCqY7

[39] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2018. Mixed Precision Training. In *International Conference on Learning Representations*.

[40] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

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

[42] Yong Rui, T.S. Huang, M. Ortega, and S. Mehrotra. 1998. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)* 8, 5 (1998), 644–655. https://doi.org/10.1109/76.718510

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115 (2015), 211–252.

[44] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2019. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision (IJCV)* 128, 2 (Oct 2019), 336–359. https://doi.org/10.1007/s11263-019-01228-7

[45] Minchul Shin, Yoonjae Cho, Byungsoo Ko, and Geonmo Gu. 2021. RTIC: Residual Learning for Text and Image Composition using Graph Convolutional Network. *arXiv preprint arXiv:2104.03015* (2021).

[46] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. 2000. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 22, 12 (2000), 1349–1380.

[47] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A Corpus for Reasoning about Natural Language Grounded in Photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 6418–6428.

[48] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. 2019. Composing text and image for image retrieval-an empirical odyssey. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6439–6448.

[49] Zhecan Wang, Noel Codella, Yen-Chun Chen, Luowei Zhou, Jianwei Yang, Xiyang Dai, Bin Xiao, Haoxuan You, Shih-Fu Chang, and Lu Yuan. 2022. CLIP-TD: CLIP Targeted Distillation for Vision-Language Tasks. *arXiv preprint arXiv:2201.05729* (2022). arXiv:2201.05729 [cs.CV]

[50] Haokun Wen, Xuemeng Song, Xin Yang, Yibing Zhan, and Liqiang Nie. 2021. Comprehensive Linguistic-Visual Composition Network for Image Retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1369–1378. https://doi.org/10.1145/3404835.3462967

[51] Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. 2021. Fashion iq: A new dataset towards retrieving images by natural language feedback. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 11307–11317.

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

[53] Yifei Yuan and Wai Lam. 2021. Conversational Fashion Image Retrieval via Multiturn Natural Language Feedback. In *Proc. of International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM. https://doi.org/10.1145/3404835.3462881

[54] Xunlin Zhan, Yangxin Wu, Xiao Dong, Yunchao Wei, Minlong Lu, Yichi Zhang, Hang Xu, and Xiaodan Liang. 2021. Product1M: Towards Weakly Supervised Instance-Level Product Retrieval via Cross-Modal Pretraining. In *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV)*. 11782–11791.

[55] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. 2017. Memory-Augmented Attribute Manipulation Networks for Interactive Fashion Search. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*. 6156–6164. https://doi.org/10.1109/CVPR.2017.652

[56] Liang Zheng, Yi Yang, and Qi Tian. 2017. SIFT meets CNN: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40, 5 (2017), 1224–1244.