G-MIXER: Geodesic Mixup-based Implicit Semantic Expansion and Explicit Semantic Re-ranking for Zero-Shot Composed Image Retrieval

Supplementary Material

1. Ablation Study and Analysis

To verify the contribution of each component in the proposed G-MIXER, we conducted a series of ablation studies. All experiments were performed using the ViT-L/14 backbone, and the captions were generated with GPT-4o for a fair comparison. The results are summarized in Table 1

Components				CIRCO		CIRR		Fashion-IQ	
Mixup	S_λ	Δ	S_m	k=5	k=10	k=1	k=5	k=10	k=50
✓	✓	✓	✓	**28.29**	**29.04**	**37.42**	**67.69**	41.92	**62.47**
✓	✓		✓	22.49	23.77	34.87	67.06	40.61	61.05
✓		✓	✓	16.43	17.47	27.40	53.23	32.36	57.38
✓			✓	11.80	12.97	28.34	55.08	30.76	56.00
✓	✓			11.32	13.24	25.59	55.71	36.82	56.71
✓		✓		12.28	13.34	12.55	29.78	24.94	53.02
✓	✓	✓		24.30	25.28	20.50	47.34	40.96	61.60
✓	✓	only ***in***	✓	17.54	18.59	26.84	61.27	33.20	57.37
✓	✓	only ***ex***	✓	21.17	22.58	32.38	63.80	**42.24**	60.91
	✓	✓	✓	24.77	25.74	33.69	63.74	34.36	55.99

Table 1. Ablation study on CIRCO, CIRR, and Fashion-IQ datasets using ViT-L/14.

Ablation on first stage top-*k*. By varying k in first stage, we observe that performance is not highly sensitive to the choice of k.

Top *k*	CIRCO		CIRR		Fashion-IQ	
	k=5	k=10	k=1	k=5	k=10	k=50
5	24.80	22.78	35.66	59.61	40.23	62.06
10	27.34	27.57	36.94	64.77	41.62	63.44
50	28.21	28.97	**37.90**	67.06	**42.41**	**64.78**
100 (Ours)	**28.29**	**29.04**	37.42	**67.69**	41.92	62.47

Table 2. Ablation study in ViT-L/14.

2. Algorithm of G-MIX Process

Algorithm 1 outlines the geodesic mixup-based implicit semantic expansion process for training-free zero-shot composed image retrieval (ZS-CIR). First, our method employs both the frozen pre-trained MLLM and CLIP encoders. The MLLM takes the reference image I_r, the manipulation text T_m, and the prompt p_s as inputs, and generates the target description T_t along with two explicit attribute texts, the include-attribute text T_{in} and the exclude-attribute text T_{ex}. The CLIP text encoder Ψ_T converts these texts into their corresponding embeddings, f_t, f_{in}, and f_{ex}, and all embeddings are used in their normalized form.

G-MIX generates query representations by performing geodesic interpolation between f_t and f_{in} using a predefined set of mixup ratios Λ. In our implementation, the mixup ratio λ starts at 0.7 and is incremented in steps of 0.05, producing multiple query variants. Each query vector is normalized and matched against the image embeddings in the database $\mathcal{D}$ based on cosine similarity. For each λ, the Top-k retrieved results $\mathcal{R}_\lambda$ are added to the unified candidate set $\mathcal{R}_{\text{union}}$, enabling the method to capture a broader range of implicit semantics that may not be reflected by a single query.

Through this process, G-MIX aggregates the Top-k retrieval results obtained from each mixup ratio into the unified candidate set $\mathcal{R}_{\text{union}}$. This candidate set is used as the input to the subsequent Explicit semantic Re-ranking stage.

3. Algorithm of ER Process

Algorithm 2 describes the Explicit semantic Re-ranking process, which refines the candidate set $\mathcal{R}_{\text{union}}$ obtained from the G-MIX process by explicitly incorporating the attributes to be included and those to be excluded. In this stage, the mixup query embedding $\hat{m}_\lambda$ generated by G-MIX is used alongside with the include-attribute embedding f_{in} and the exclude-attribute embedding f_{ex} to assess how well each candidate image satisfies these attribute conditions.

For each candidate image embedding f^j_{set}, the similarity to the mixup query is first computed, followed by the independent computation of its similarities to the include-attribute embedding and the exclude-attribute embedding. These three similarity measures capture different aspects of the information, and the ER stage combines them to perform a more precise attribute-based ranking.

To reflect the relative relationship between the include and exclude attributes, an additional adjustment term is calculated. This term increases the score of a candidate when its similarity to the mixup query is higher than its similarity to the exclude-attribute embedding, and decreases the score when the similarity falls below that of the include-attribute embedding. Through this adjustment, candidates that do not satisfy the intended attribute conditions are effectively prevented from being ranked near the top.

Finally, the mixup similarity, the modification text–based similarity, and the adjustment term are combined to compute the final score. Among all candidates, the candidate with the highest final score is selected as I_t, which serves as the final retrieval result produced by the overall G-MIXER method.

Algorithm 1 G-MIX Algorithm

Require: Reference image I_r, manipulation text T_m, prompt p_s, image-search database $\mathcal{D}$, Frozen MLLM Ψ_M, frozen CLIP vision encoder Ψ_I, language encoder Ψ_T

Ensure: Candidate Set R_{union}

Initialize pre-trained and frozen models Ψ_M, Ψ_I, Ψ_T.

Generate target image description (Eq. 4):

$$T_t, T_{in}, T_{ex} = \Psi_M(p_s \circ I_r \circ T_m)$$

Compute normalized text embedding f_t:

$$f_t = \frac{\Psi_T(T_t)}{\|\Psi_T(T_t)\|}.$$

Compute normalized text embedding f_{in}, f_{ex}, f_m,

Initialize first-stage candidate set: $\mathcal{R}_{\text{union}} \leftarrow \emptyset$.

for each $\lambda \in \Lambda$ **do**

Geodesic mixup feature (Eq. 1):

$$\mathbf{m}_\lambda(\mathbf{f}_t, \mathbf{f}_i) = \mathbf{f}_t \frac{\sin(\lambda\theta)}{\sin(\theta)} + \mathbf{f}_i \frac{\sin((1-\lambda)\theta)}{\sin(\theta)}.$$

Normalize query feature: $\mathbf{m}_\lambda$,

Similarity computation: for each image $I_j \in \mathcal{D}$, compute score $s_\lambda(j) = \mathbf{m}_\lambda^\top f_i$.

Top-k retrieval (Eq. 2):

$$\mathcal{R}_\lambda = \text{TopK}(s_\lambda, k).$$

Update first-stage candidate set (Eq. 3):

$$\mathcal{R}_{\text{union}} \leftarrow \mathcal{R}_{\text{union}} \cup \mathcal{R}_\lambda.$$

end for

return R_{union}

Algorithm 2 ER Algorithm

Require: Candidate Set R_{union}, normalized text embedding f_{in}, f_{ex}, f_m, frozen CLIP vision encoder Ψ_I

Ensure: Retrieved target image I_t

for each image I_{set}^j in $\mathcal{R}_{\text{union}}$ **do**

Compute normalized image embedding:

$$f_{set}^j = \Psi_I(I_{set}^j)$$

Compute similarity w.r.t. modification text:

$$S_m(j) = (f_{set}^j)^\top f_m$$

Compute similarity w.r.t. mixup query:

$$S_\lambda(j) = (f_{set}^j)^\top \hat{\mathbf{m}}_\lambda$$

Compute similarity w.r.t. include caption:

$$S_{in}(j) = (f_{set}^j)^\top f_{in}$$

Compute similarity w.r.t. exclude caption:

$$S_{ex}(j) = (f_{set}^j)^\top f_{ex}$$

Compute **margin-based adjustment term (Eq. 5)**:

$$\Delta(j) = \text{ReLU}(S_\lambda(j) - S_{ex}(j)) - \text{ReLU}(S_\lambda(j) - S_{in}(j))$$

Compute **final score (Eq. 6)**:

$$\text{FinalScore}(j) = S_m(j) + S_\lambda(j) + \Delta(j)$$

end for

Retrieve target image:

$$I_t = \arg\max_{I_{\text{set}}^j \in \mathcal{R}_{\text{union}}} \text{FinalScore}(j)$$

return I_t

4. Complete Prompt Template

Most existing LLM/MLLM-based ZS-CIR methods incorporate thought, reasoning chains, or reflection prompts to encourage explicit inference of implicit information. While such prompts can help generate detailed descriptions, they often cause the model to over-explicitly rewrite contextual cues from the reference image. This explicit conversion undesirably narrows the retrieval space, reducing the diversity of generated candidates and hindering the discovery of images that rely on subtle or ambiguous contextual relations.

In contrast, our method adopts a step-by-step instruction rather than explicit reasoning prompts (Figure 1). This design intentionally avoids forcing the model to verbalize all implicit information. Instead, the step-by-step structure guides the MLLM to decompose the task (e.g., identifying key attributes, separating include/exclude factors, generating minimal target cues) while preserving ambiguous or implicit elements in their original form.

By preventing unnecessary explicit expansions, the proposed prompt maintains a balanced description that captures essential modification cues without over-specifying implicit context. This contributes to a broader and more flexible retrieval space, enabling G-MIXER to effectively capture implicit compositional relationships during geodesic mixup.

5. Geodesic Mixup vs Linear Mixup

In unimodal settings, the linear mixup technique (Eq. 1) generates augmented features by linearly interpolating two embeddings f_t and f_i with coefficients λ and $1-\lambda$. This linear interpolation naturally fits Euclidean feature spaces but

```
- You are an image description expert for composed image retrieval.
- You are given an Original Image and a Manipulation Text describing how the Target Image differs from the original.
- Your goal is to generate a concise textual description of the Target Image.

You are given:
- An Original Image
- A Manipulation Text describing how the Target Image differs from the Original Image.
{
 "Original Image": <image_url>,
 "Manipulation Text": <manipulation_text>
}

Global Rules
 - Be concise and concrete.
 - Do not use pronouns (“it,” “they,” “this”), comparatives/superlatives (“bigger,” “more,” “less,” “largest”), or vague terms (“something,” “stuff,” “nice”).
 - Use plain, general expressions that can stand alone for retrieval.
 - Only state changes that are explicitly mentioned or strongly implied by the Manipulation Text. Avoid guessing hidden attributes (e.g., lighting/mood) unless stated.
 - When listing tokens, use lowercase, noun/adjective phrases (no punctuation except hyphens within a word).

### Step-by-Step Instructions
## Step 1. Write the Original Image Description
Provide a single, neutral sentence that factually describes the Original Image's visible content (people/objects, key attributes like color, pattern, clothing type, pose, viewpoint, salient scene context).

## Step 2. Plan the Target Changes
Based on the Manipulation Text and the Original Image Description, reason about how the description must change to reach the target. Broadly consider these possible change types and apply only those present
 - Broadly consider possible change types such as cardinality, addition, negation, direct addressing, comparison and change, conjunction, viewpoint, and spatial relations.
Write 2-4 short bullets that state concrete edits to the description (e.g., “add 'wearing sunglasses'” “replace 'red shirt' with 'blue shirt'” “remove 'hat'” “change viewpoint to side view”). Avoid comparatives and pronouns.

## Step 3. Describe the Target Image
Using the plan, write a single, stand-alone sentence that describes the Target Image after the changes. It must be retrieval-ready on its own:
 - No pronouns/comparatives/superlatives.
 - Use generic, common vocabulary.
 - Keep it specific enough to retrieve the target but not over-fitted.

## Step 4. Describe Changed Elements (Rationale)
Write 1-2 sentences explaining why the chosen include/exclude tokens are necessary, explicitly tying them to the Manipulation Text and the Target Description (no pronouns; refer to attributes by name).

## Step 5. Describe Changed Elements (Include / Exclude)
From the Original + Manipulation + Target Description, infer two must-include and two must-exclude items that will help retrieval filters.
Rules:
 - If the Manipulation Text adds something (e.g., “wearing sunglasses”), put that exact attribute in include, and infer the opposite in exclude (e.g., “bare face” or “without sunglasses”).
 - If the Manipulation Text removes something (e.g., “remove hat”), put the opposite in exclude (“wearing a hat”) and ensure the new state is in include (“without hat” / “bare head”).
 - Use short, atomic tokens in lowercase (e.g., “sunglasses,” “blue shirt,” “side view,” “no hat”). Avoid verbs when possible; prefer noun/adjective phrases.
 - Provide exactly two items for each of include and exclude. If fewer are justified, choose the two most salient.
```

Figure 1. **Prompt template** used in our method. The template provides step-by-step instructions (original description → planned changes → target description → include/exclude tokens) while restricting explicit inference. By avoiding thought/reflection-style reasoning, it prevents unnecessary expansion of implicit information and helps maintain a broad retrieval space.

becomes problematic when applied to vision–language pre-trained (VLP) models such as CLIP. Since CLIP is trained with cosine similarity across image and text modalities, its embedding space is normalized to lie on a unit hypersphere, and semantic relationships are represented by angular distances. When standard mixup is applied in this hyperspherical space, the resulting features deviate from the manifold and fail to preserve the semantic trajectory between the two embeddings, producing distorted or semantically invalid intermediate representations.

Methods	CIRCO		CIRR		Fashion-IQ	
	k=5	k=10	k=1	k=5	k=10	k=50
G-MIXER m_λ	**28.29**	**29.04**	**37.42**	**67.69**	**41.92**	**62.47**
Mixup $\tilde{m}_\lambda$	27.14	27.98	31.59	62.22	40.53	61.76

Table 3. Comparison between geodesic mixup and linear mixup

$$\tilde{m}_\lambda(f_t, f_i) = f_t \cdot \lambda + f_i \cdot (1 - \lambda) \tag{1}$$

To address this issue, geodesic mixup (Eq. 2) was introduced. Instead of linear interpolation, it performs interpolation along the geodesic curve that connects two embeddings on the hypersphere. This formulation preserves the angular geometry of the embedding space and yields intermediate representations that follow the true semantic path between f_t and f_i. As our setting involves bi-modality queries that combine image and text information, we adopt geodesic mixup rather than the standard linear variant to ensure semantically consistent composed features.

$$m_\lambda(f_t, f_i) = f_t \frac{\sin(\lambda\theta)}{\sin(\theta)} + f_i \frac{\sin((1-\lambda)\theta)}{\sin(\theta)} \tag{2}$$

The effectiveness of this choice is clearly demonstrated in Table 3. Across all benchmarks, geodesic mixup consistently outperforms standard mixup. Particularly in CIRR, the improvements at k=1 and k=5 retrieval accuracy are substantial, confirming that following the correct semantic path on the hypersphere leads to more accurate composed features and improved retrieval performance.