DECOUPLING ENDPOINT AND SEMANTIC TRANSITION LEARNING FOR ZERO-SHOT COMPOSED IMAGE RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot composed image retrieval (ZS-CIR) retrieves a target image from a reference image and a text modification without human-annotated CIR triplets. Projection-based ZS-CIR methods are attractive because they do not rely on LLMs at inference and remain lightweight, but they often underperform LLM-based approaches on complex semantic modifications. This gap reflects a semantic transition bottleneck in projection-based ZS-CIR: endpoint-level matching can let the edit text act as a target-side attribute cue rather than grounding it as a source-conditioned semantic transition. We further show that adding semantic transition supervision to the same text adapter creates an endpoint–transition conflict between endpoint alignment and semantic transition alignment. To address this conflict, DeCIR decouples endpoint and transition learning. It constructs paired forward/reverse edit tuples from image-caption pairs, trains separate low-rank text adapter branches for endpoint alignment and semantic transition alignment, and merges them with Low-Rank Directional Merge (LRDM) into one deployable adapter. Extensive experiments on CIRR, CIRCO, FashionIQ, and GeneCIS demonstrate that DeCIR consistently improves projection-based ZS-CIR without increasing inference complexity.

1 INTRODUCTION

Composed image retrieval (CIR) searches for a target image that matches both a reference image and a text modification. The target should apply the requested edit while preserving relevant visual evidence from the reference, i.e., it must be reachable from the reference under the instruction rather than merely satisfy an endpoint cue. Conventional CIR models learn this behavior from human-annotated reference–instruction–target triplets, which are expensive to collect and difficult to scale. Zero-shot composed image retrieval (ZS-CIR) instead trains on large image-caption datasets such as CC3M (Sharma et al., 2018), but without triplets the model must infer source-to-target semantic transitions from weaker supervision. This makes the central supervision problem sharper: the model must infer how the source should change, even though the desired transition is not directly observed.

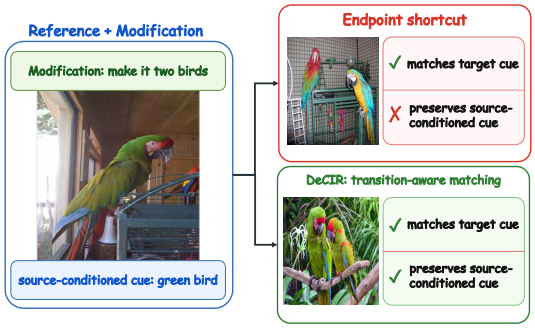

Figure 1: **Illustration of our motivation.** The semantic transition bottleneck in projection-based ZS-CIR can lead to an endpoint shortcut, where the edit text is treated as a target-side cue while source-conditioned evidence from the reference is dropped. This motivates DeCIR to explicitly learn transition-aware matching.

Existing ZS-CIR methods mainly follow two routes. LLM-based methods leverage the knowledge and reasoning abilities acquired during pretraining to infer a semantic transition from the source to the target (Karthik et al., 2024; Sun et al., 2025; Huynh et al., 2025; Tang et al., 2025b; Wang et al., 2026). By rewriting the composed query into a target caption, LLMs serve as a bridge from the reference–modification input to a hypothesized target endpoint, making them effective at handling complex edits, but often at the cost of heavier inference pipelines. Projection-based methods take a much lighter route: they map the reference image into the text embedding space of a pretrained

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

vision-language model such as CLIP, combine it with the modification text, and encode the resulting prompt into a composed query embedding for retrieval (Saito et al., 2023; Baldrati et al., 2023a; Tang et al., 2024; Suo et al., 2024; Zhong et al., 2025). This route is attractive for deployment because it avoids heavy LLM calls and keeps the retrieval pipeline lightweight. However, by avoiding an explicit reasoning bridge, projection-based ZS-CIR leaves the source-to-target transition to be inferred implicitly by the CLIP text encoder, which encodes the pseudo-word and modification into a retrieval embedding rather than explicitly constructing target semantics as an LLM can. This raises a key question: can such a lightweight projection-based pipeline reliably capture source-to-target semantic transitions for complex modifications?

We argue that existing projection-based methods remain limited by a *semantic transition bottleneck*: The CLIP text encoder is expected to infer the source-to-target semantic transition, although such transitions are neither explicitly supervised in contrastive pretraining nor directly optimized during pseudo-word mapping network training. Without source-conditioned semantic transition supervision, the modification text can collapse into a target-side attribute cue under complex edits: the model may capture the target attribute in the instruction without binding it to the source condition provided by the reference image. For example, as shown in Figure 1, the reference image supplies the source-conditioned cue, a green bird, while the instruction is only "make it two birds". A projection-based model may treat the text as the endpoint cue "two birds" and retrieve any image containing two birds, instead of learning the semantic transition from one green bird in the reference image to two green birds. This shortcut can work in easy cases but becomes brittle when hard distractors share the same target attribute without satisfying the intended source-conditioned semantic transition.

Explicit semantic transition supervision is a natural way to address this bottleneck. However, we further show that adding it to the same adapter introduces a new optimization challenge. Endpoint alignment pulls the composed query toward target endpoints, whereas semantic transition alignment asks instruction embeddings to represent source-to-target displacement. These objectives are complementary for CIR, but forcing them through the same low-rank text adapter can cause gradient interference, which is especially pronounced in deeper text transformer blocks and thus degrade retrieval performance. This evidence motivates conflict-aware endpoint–transition decoupling: the model should allow endpoint alignment and transition alignment to specialize during training while keeping a lightweight projection-based retrieval pipeline at inference.

We propose DeCIR, a conflict-aware endpoint–transition decoupling framework for lightweight ZS-CIR. We leverage an LLM to convert image-caption pairs into paired forward/reverse edit tuples, providing diverse semantic transition supervision without human-annotated CIR triplets. DeCIR then trains two low-rank text-adapter branches over the CLIP text encoder: an endpoint alignment branch that aligns the composed query embedding with the target-caption endpoint, and a transition branch for instruction-level semantic displacement. Finally, Low-Rank Directional Merge (LRDM) folds the semantic transition alignment branch into the endpoint alignment branch, producing a single deployable text adapter. The resulting model preserves the standard projection-based inference pipeline while obtaining stronger source-conditioned transition awareness.

Overall, our contributions are summarized as follows:

- We identify a semantic transition bottleneck in projection-based ZS-CIR, where edit text is matched as a target-side cue rather than grounded as a source-conditioned transition.
- We show that endpoint and transition objectives interfere in the same low-rank adapter, motivating decoupled endpoint and transition learning.
- We introduce LRDM, a shared-basis coefficient merge that folds transition sensitivity into the projection-based retrieval pipeline while retaining lightweight inference.
- Experiments on CIRR, CIRCO, FashionIQ, and GeneCIS show that DeCIR improves projection-based ZS-CIR without LLMs or reasoning loops at inference.

2 Related Work

2.1 Zero-Shot Composed Image Retrieval

Composed image retrieval was first studied with human-annotated reference–instruction–target triplets (Vo et al., 2019; Chen et al., 2020b; Wu et al., 2021; Liu et al., 2021; Baldrati et al., 2022).

ZS-CIR removes these triplets and mainly follows two paths. Projection-based methods map the reference image into the text space and retrieve with a single composed query (Saito et al., 2023; Baldrati et al., 2023a; Tang et al., 2024; Suo et al., 2024; Zhong et al., 2025); related CIR work further improves composition modules, diffusion proxies, hard negatives, and robustness settings (Baldrati et al., 2023b; Gu et al., 2024a; Kwak et al., 2025; Psomas et al., 2024; Sun et al., 2026). These methods strengthen different parts of the endpoint-matching pipeline, such as source pseudo-word inversion, context-dependent visual mapping, or instruction-aware distillation. However, the modification itself is not explicitly isolated and supervised as a source-conditioned semantic transition. LLM-based methods leverage LLMs or unified multimodal models (UMMs) to reason over the composed query, rewrite it into a modified caption, or even synthesize imagined target images to improve retrieval (Karthik et al., 2024; Sun et al., 2025; Huynh et al., 2025). These methods (Tang et al., 2025b; Li et al., 2025a; Wang et al., 2026) have demonstrated strong reasoning ability and impressive qualitative results, but such gains often come at the cost of heavy inference pipelines.

DeCIR instead stays in the projection-based setting and is closest to DistillCIR (Zhong et al., 2025) in that both use LLM-generated supervision during training rather than LLMs at inference. The difference is the role assigned to that supervision. DistillCIR uses LLM-generated signals to distill the instruction awareness of LLMs into projection-based models, while DeCIR uses generated forward/reverse edits to explicitly supervise source-conditioned semantic displacement and studies how this transition objective interacts with existing endpoint matching.

2.2 Vision-Language Adaptation and Model Merging

Modern CIR methods rely heavily on pretrained vision-language models. Beyond CLIP, large-scale image-text pretraining has been improved through noisy web supervision, caption bootstrapping, frozen-encoder bootstrapping, and sigmoid contrastive losses (Jia et al., 2021; Li et al., 2022; 2023; Zhai et al., 2023). Lightweight tuning then specializes these models without full retraining, using adapters, prefix tuning, visual prompt tuning, or multimodal prompt learning (Houlsby et al., 2019; Li & Liang, 2021; Jia et al., 2022; Khattak et al., 2023). These techniques provide strong reusable representations and low-cost adaptation, but they do not by themselves specify how a modification instruction should encode a source-conditioned semantic transition. DeCIR uses this lightweight adaptation setting as the base, then separates endpoint and transition learning.

Model merging combines multiple fine-tuned models into a single checkpoint while preserving useful capabilities. Prior work studies weight averaging, importance-weighted merging, task-vector arithmetic, interference-aware merging, robust ability transfer, and low-cost merging settings (Wortsman et al., 2022; Matena & Raffel, 2022; Ilharco et al., 2023; Yadav et al., 2023; Yu et al., 2024; Zeng et al., 2025). DeCIR uses merging for a different purpose: the endpoint alignment branch defines the inference backbone, while the transition branch contributes semantic transition sensitivity learned from paired forward/reverse supervision. In this paper, we propose LRDM to merge low-rank coefficients while keeping the retrieval-oriented output basis fixed.

3 Method

3.1 Problem Formulation and Model Overview

Composed image retrieval aims to retrieve a target image from a reference image I_{ref} and a text modification t. DeCIR uses a projection-based CIR model based on the pretrained Pic2Word (Saito et al., 2023). Pic2Word uses a CLIP (Radford et al., 2021) visual encoder E_V, a CLIP text encoder E_T, and a lightweight mapping network f_ϕ consisting of a three-layer MLP. Given I_{ref}, the visual encoder first extracts an image feature, and the mapping network projects it into a pseudo-word token in the CLIP text embedding space, which is denoted as $s_{\text{ref}} = f_\phi(E_V(I_{\text{ref}}))$. Subsequently, this token is inserted into a text prompt with the modification, producing a composed query embedding. Retrieval is then performed by matching this query with cached gallery image embeddings. Unless otherwise noted, embeddings are ℓ_2 normalized, and $S(a, b) = a^\top b$ denotes cosine similarity.

DeCIR keeps this lightweight projection-based inference form while adding endpoint alignment and semantic transition supervision during training. Instead of treating generated supervision only as a source-to-target transition, DeCIR constructs paired forward and reverse edits from each image-caption pair. The forward edit maps the source caption toward a synthetic target caption, while the

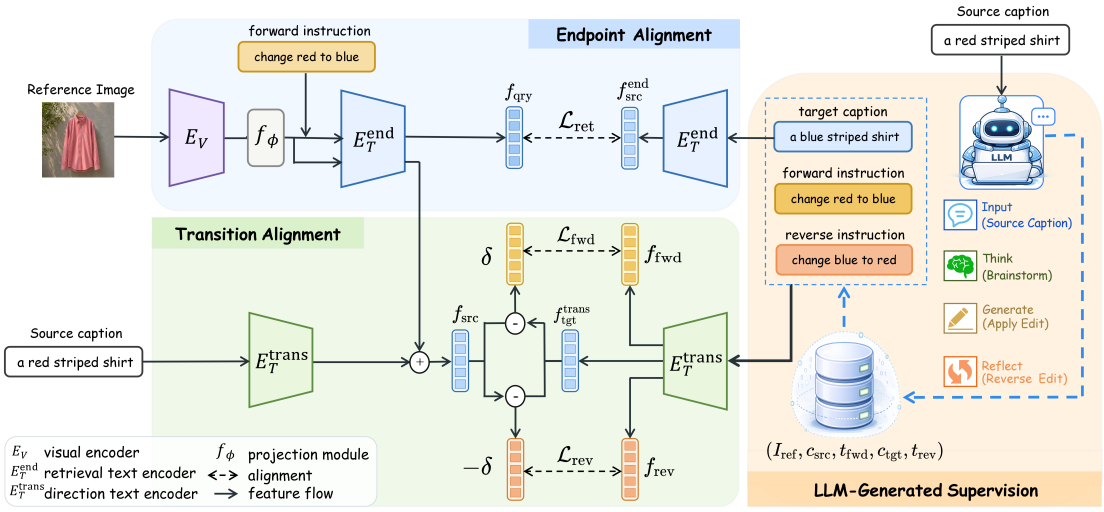

Figure 2: **Overview of DeCIR.** DeCIR decouples endpoint alignment and semantic transition alignment into two specialized branches during training. The text adapters in the two branches, i.e., E_T^{end} and E_T^{trans}, are then merged into a single deployable adapter via Low-Rank Directional Merge, enabling efficient inference.

reverse edit maps the target caption back toward the source semantics. The target caption supports endpoint alignment, and the forward/reverse instructions support semantic transition alignment. Since optimizing these two objectives in one shared text adapter can introduce interfering updates, DeCIR decouples endpoint alignment and semantic transition learning, then uses LRDM to fold them back into one deployable adapter, enabling efficient inference without any reliance on LLMs.

3.2 Endpoint and Semantic Transition Alignment

A standard image-caption pair does not specify which semantic change should be applied to the source, so it cannot directly supervise source-conditioned semantic transitions. For each CC3M pair $(I_{\text{ref}}, c_{\text{src}})$, we prompt an LLM (Team GLM et al., 2024) to identify source details, brainstorm one plausible visual change, write a forward instruction, apply it to obtain a modified target caption, and write the reverse instruction that undoes only that change. This produces a bidirectional CIR-style tuple:

$$(I_{\text{ref}},\ c_{\text{src}},\ t_{\text{fwd}},\ c_{\text{tgt}},\ t_{\text{rev}})\,. \tag{1}$$

The target caption supplies the endpoint for endpoint alignment, while t_{fwd} and t_{rev} supply opposite semantic transitions. The tuple is bidirectional only at the instruction and text level: no reverse image retrieval pair is created, and the reverse instruction is used only for semantic transition supervision. Appendix B gives the detailed prompt and examples of llm-generated supervision.

Endpoint alignment. The endpoint alignment branch learns to align the composed query with the generated target caption embedding, providing basic multimodal retrieval capability. Let E_T^{end} denote the endpoint alignment text encoder and D_T the text embedding dimension. We use $\mathcal{T}_{\text{cir}}(t)$ for a composed prompt such as "a photo of * and t" and $\mathcal{T}_{\text{src}}(*)$ for the source-only prompt "a photo of *," where $*$ is replaced by the pseudo token. For one training tuple, we define:

$$\begin{aligned} f_{\text{qry}} &= E_T^{\text{end}}(\mathcal{T}_{\text{cir}}(t_{\text{fwd}}), f_\phi(E_V(I_{\text{ref}})))\,, \\ f_{\text{tgt}}^{\text{end}} &= E_T^{\text{end}}(c_{\text{tgt}}), \qquad f_{\text{qry}}, f_{\text{tgt}}^{\text{end}} \in \mathbb{R}^{D_T}. \end{aligned} \tag{2}$$

Let $\mathcal{F}_{\text{qry}}$ and $\mathcal{F}_{\text{tgt}}^{\text{end}}$ be the query and target-caption embeddings in the mini-batch. We optimize the symmetric contrastive loss

$$\mathcal{L}_{\text{end}} = -\frac{1}{2}\left[\log \frac{\exp(\tau S(f_{\text{qry}}, f_{\text{tgt}}^{\text{end}}))}{\sum_{\bar{f}\in\mathcal{F}_{\text{tgt}}^{\text{end}}} \exp(\tau S(f_{\text{qry}}, \bar{f}))} + \log \frac{\exp(\tau S(f_{\text{tgt}}^{\text{end}}, f_{\text{qry}}))}{\sum_{\bar{f}\in\mathcal{F}_{\text{qry}}} \exp(\tau S(f_{\text{tgt}}^{\text{end}}, \bar{f}))}\right], \tag{3}$$

where τ is the learned temperature. This follows the common contrastive-learning form (Chen et al., 2020a; Radford et al., 2021), and the objective is averaged over the mini-batch.

Transition alignment. The semantic transition alignment branch E_T^{trans} aims to teach edit instructions to encode the direction of change from the source semantics to the target semantics. We first compute the source and target anchors and use their difference as a transition proxy. The transition branch then learns to align forward and reverse instruction embeddings with this semantic displacement. The source anchor is built from the source caption and the reference-image-conditioned source prompt:

$$\begin{aligned} f_{\text{src}}^{\text{cap}} &= E_T^{\text{trans}}(c_{\text{src}}), \quad f_{\text{src}}^{\text{cap}} \in \mathbb{R}^{D_T}, \\ f_{\text{src}}^{\text{img}} &= E_T^{\text{trans}}(\mathcal{T}_{\text{src}}(*), f_\phi(E_V(I_{\text{ref}}))), \quad f_{\text{src}}^{\text{img}} \in \mathbb{R}^{D_T}, \\ f_{\text{src}} &= (1-\omega) f_{\text{src}}^{\text{cap}} + \omega f_{\text{src}}^{\text{img}}, \quad f_{\text{src}} \in \mathbb{R}^{D_T}. \end{aligned} \tag{4}$$

Here ω controls how much image-conditioned evidence enters the anchor. Given this source anchor, the target caption provides the endpoint used to define the desired semantic transition. Notably, this endpoint is encoded by the semantic transition text encoder, and the source-to-target difference becomes the transition proxy:

$$f_{\text{tgt}}^{\text{trans}} = E_T^{\text{trans}}(c_{\text{tgt}}), \quad \delta = f_{\text{tgt}}^{\text{trans}} - f_{\text{src}}, \quad f_{\text{tgt}}^{\text{trans}}, \delta \in \mathbb{R}^{D_T}. \tag{5}$$

Thus, δ is a source-to-target displacement for semantic transition. Since no target image is observed during CC3M training, this displacement is a text-embedding proxy for the desired visual edit. Before defining the transition loss, the semantic transition branch maps both generated edit instructions into the same text space:

$$f_{\text{fwd}} = E_T^{\text{trans}}(t_{\text{fwd}}), \quad f_{\text{rev}} = E_T^{\text{trans}}(t_{\text{rev}}), \quad f_{\text{fwd}}, f_{\text{rev}} \in \mathbb{R}^{D_T}. \tag{6}$$

We adapt the pretrained text encoder to learn directional instruction representations, aligning the forward and reverse embeddings with δ and $-\delta$, respectively. The two transition losses are

$$\mathcal{L}_{\text{fwd}} = 1 - S(f_{\text{fwd}}, \delta), \qquad \mathcal{L}_{\text{rev}} = 1 - S(f_{\text{rev}}, -\delta). \tag{7}$$

Here, the forward term makes the generated instruction follow the source-to-target displacement, while the reverse term uses the paired reverse instruction as an opposite-direction constraint. We combine the two terms to train the semantic transition branch for semantic transition alignment:

$$\mathcal{L}_{\text{trans}} = \mathcal{L}_{\text{fwd}} + \mathcal{L}_{\text{rev}}. \tag{8}$$

Together, the two terms encourage the modification text to encode a semantic displacement instead of serving only as a target-side attribute cue.

3.3 Shared Adapter Interference

A shared adapter can optimize $\mathcal{L}_{\text{end}} + \mathcal{L}_{\text{trans}}$ directly, but the two losses impose different local updates on the same low-rank factors. This can lead to gradient conflict: endpoint alignment and transition alignment may push the shared adapter in divergent directions, degrading retrieval performance. In this section, we diagnose this effect with an offline layer-wise gradient probe (Hao et al., 2026) rather than relying on raw gradient similarity alone.

For an adapted text block ℓ, write the LoRA update as $\Delta W^\ell = B^\ell A^\ell$. Starting from the same shared-adapter checkpoint, we aggregate and vectorize the gradients induced by $\mathcal{L}_{\text{end}}$ and $\mathcal{L}_{\text{trans}}$ over multiple mini-batches, denoted as $\bar{g}_{\text{end}}^\ell$ and $\bar{g}_{\text{trans}}^\ell$. Their cosine similarity measures the raw agreement between the endpoint and transition objectives. However, raw agreement can be biased by layer-dependent gradient stability, since different transformer blocks may naturally exhibit different levels of stochastic gradient consistency. To control for this effect, we also estimate a same-objective baseline by splitting gradients from the same objective into two disjoint groups, $\bar{g}_A^\ell$ and $\bar{g}_B^\ell$, where the subscripts denote the two gradient groups rather than the LoRA factors. The cross-objective and same-objective agreements are

$$s_{\text{cross}}^\ell = S\big(\bar{g}_{\text{end}}^\ell, \bar{g}_{\text{trans}}^\ell\big), \qquad s_{\text{base}}^\ell = S\big(\bar{g}_A^\ell, \bar{g}_B^\ell\big). \tag{9}$$

A larger interference score $\mathrm{GI}_\ell = s^\ell_{\mathrm{base}} - s^\ell_{\mathrm{cross}}$ means that endpoint and transition gradients agree less than expected under ordinary same-objective stochastic variation. In other words, a high GI_ℓ indicates that the disagreement is not merely due to noisy gradients in that layer, but reflects additional conflict between the endpoint and transition objectives. Figure 3 shows a clear conflict peak in middle and deep layers; PCGrad reduces the mean level but retains the same peak pattern. Together with the joint-training ablation in Table 3, this motivates separating the objectives during training and merging them only after specialization.

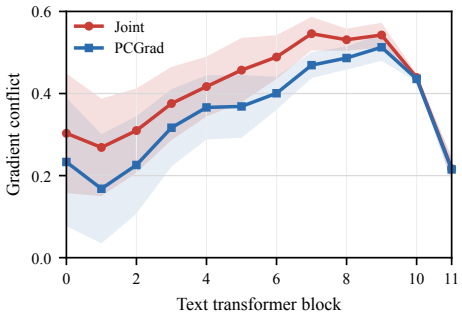

Figure 3: **Layer-wise gradient interference**. Result is shown with mean ± std over five seeds.

3.4 Low-Rank Directional Merge

DeCIR trains two text adapter branches initialized from the same CLIP text encoder. The endpoint alignment branch E_T^{end} optimizes $\mathcal{L}_{\mathrm{end}}$ and defines the deployable composed retrieval pathway, while the transition branch E_T^{trans} optimizes $\mathcal{L}_{\mathrm{trans}}$ and learns semantic instruction displacement. The visual encoder LoRA parameters and the inversion network f_ϕ belong to the retrieval pathway and are updated only by $\mathcal{L}_{\mathrm{end}}$.

LRDM makes the later merge well defined by sharing the low-rank output basis. For each adapted text transformer layer ℓ, the two branches share B^ℓ but keep branch-specific coefficients:

$$\Delta W^\ell_{\mathrm{end}} = B^\ell A^\ell_{\mathrm{end}}, \qquad \Delta W^\ell_{\mathrm{trans}} = B^\ell A^\ell_{\mathrm{trans}}. \tag{10}$$

Training uses two sequential passes over the same mini-batch. At each adapted layer ℓ, the endpoint pass updates A^ℓ_{end} and B^ℓ, along with the visual LoRA parameters, f_ϕ, and τ; the transition pass updates only A^ℓ_{trans}. Thus, transition supervision changes semantic transition coefficients without rewriting the retrieval basis or visual pathway.

After training, LRDM restores a single low-rank adapter by merging coefficients rather than full weights:

$$A^\ell_{\mathrm{merge}} = (1-\alpha)A^\ell_{\mathrm{end}} + \alpha A^\ell_{\mathrm{trans}}. \tag{11}$$

The deployable layer keeps the shared basis and absorbs the merged update:

$$B^\ell_{\mathrm{merge}} = B^\ell, \qquad W'^\ell = W^\ell + B^\ell_{\mathrm{merge}} A^\ell_{\mathrm{merge}}. \tag{12}$$

Here α controls how much semantic transition information is injected into the low-rank adapter. LRDM is asymmetric: the endpoint alignment branch keeps the backbone and shared basis, while the transition branch contributes semantic transition coordinates inside that basis. The transition branch is then discarded; gallery embeddings are encoded once with the final visual encoder and cached, and query retrieval uses only the merged pathway. Algorithm 1 in Appendix B summarizes the full training and merge procedure.

4 Experiments

4.1 Experimental Setup

For data processing, we prompt the lightweight LLM `glm-4-flash` to generate a forward edit, a modified target caption, and a reverse edit as described in Section 3.2, obtaining 667,229 format-valid tuples after format filtering. To ensure supervision quality, we manually review random samples for edit fidelity and reverse-instruction consistency. Appendix B provides the prompt, examples, and other details. We evaluate DeCIR with CLIP ViT-B/32 and ViT-L/14 backbones. Following common projection-based baselines, we train on CC3M and initialize from the pretrained Pic2Word (Saito et al., 2023) checkpoint and employ AdamW (Loshchilov & Hutter, 2019) with a learning rate of 2×10^{-5}, weight decay of 0.1, and a linear warmup of 200 steps. The batch size is 768. The training is conducted on 4 NVIDIA A100 (80G) GPUs for one epoch with cosine learning rate scheduling. For parameter-efficient training, we train the full Pic2Word mapping network while applying LoRA (Hu et al., 2022) to both visual and text encoder projection layers in CLIP. We report the performance averaged over three trials.

Table 1: **Quantitative comparison on the test set of CIRCO and the validation set of Fashion-IQ.** We report the mAP@K metrics for CIRCO test set and the R@K metrics for Fashion-IQ validation set across ViT-B/32 and ViT-L/14 backbones. Best results are in bold, second-best results are underlined, and our method is highlighted in grey.

	Method	CIRCO				FashionIQ							
		mAP@k				Dress		Shirt		TopTee		Avg.	
		k=5	k=10	k=25	k=50	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50
ViT-B/32	SEARLE (ICCV'23)	9.4	9.9	11.1	11.8	18.5	39.5	24.4	41.6	25.7	46.5	22.9	42.5
	Slerp (ECCV'24)	9.3	10.3	11.7	12.3	19.2	42.1	23.0	42.0	26.6	47.8	23.0	44.0
	CIReVL (ICLR'24)	14.9	15.4	17.0	17.8	**25.3**	**46.4**	28.4	47.8	31.2	53.9	28.3	49.4
	DistillCIR (ICCV'25)	16.3	16.9	18.5	19.4	25.1	45.8	29.3	49.1	33.9	53.1	29.5	49.3
	DeCIR (ours)	**17.1**	**17.6**	**19.1**	**20.0**	25.2	46.0	**29.9**	**49.4**	**33.9**	**54.1**	**29.7**	**49.8**
ViT-L/14	Pic2Word (CVPR'23)	8.7	9.5	10.6	11.3	20.0	40.2	26.2	43.6	27.9	47.4	24.7	43.7
	SEARLE (ICCV'23)	11.7	12.7	14.3	15.1	20.5	43.2	27.4	45.7	29.3	50.2	25.7	46.4
	LinCIR (CVPR'24)	12.6	13.6	15.0	15.9	20.9	42.4	29.1	46.8	28.8	50.2	26.3	46.5
	Context-I2W (AAAI'24)	13.0	14.6	16.1	17.2	23.1	45.3	29.7	48.6	30.6	52.9	27.8	48.9
	Slerp (ECCV'24)	18.5	19.4	21.4	22.4	23.4	45.1	29.6	46.5	32.0	51.2	28.3	47.6
	CIReVL (ICLR'24)	18.6	19.0	20.9	21.8	24.8	44.8	29.5	47.4	31.4	53.7	28.5	48.6
	MOA (SIGIR'25)	15.3	17.1	18.5	19.3	25.2	48.5	31.9	50.7	33.2	54.8	30.1	51.3
	HIT (ICCV'25)	15.5	16.7	18.9	19.9	25.6	47.1	**32.4**	**51.2**	32.8	54.7	30.3	51.0
	DistillCIR (ICCV'25)	18.7	20.3	22.8	23.9	25.5	46.6	30.7	50.1	33.9	53.7	30.0	50.2
	DeCIR (ours)	**21.3**	**22.5**	**24.7**	**25.7**	**25.8**	**49.8**	31.2	50.2	**34.1**	**54.9**	**30.4**	**51.6**

4.2 Benchmarks and Baselines

We evaluate DeCIR on four CIR benchmarks that cover different aspects of composed retrieval: CIRR for natural image disambiguation, CIRCO for multi-target ranking, FashionIQ for fashion edits, and GeneCIS for conditional retrieval. Following established practice, we report Recall@k (R@k) for CIRR, FashionIQ, and GeneCIS, with the CIRR subset metric (R_s@k) measuring retrieval within a curated candidate set. Because CIRCO can include multiple correct targets for a single query, we report mean Average Precision (mAP@k). We compare DeCIR with representative projection-based methods, including Pic2Word (Saito et al., 2023), SEARLE (Baldrati et al., 2023a), LinCIR (Gu et al., 2024b), Context-I2W (Tang et al., 2024), Slerp (Jang et al., 2024), HIT (Li et al., 2025b), MOA (Li et al., 2025c), PrediCIR (Tang et al., 2025a), and DistillCIR (Zhong et al., 2025), as well as the LLM-based CIReVL (Karthik et al., 2024). Due to space limitations, additional results are provided in Appendix A.

CIRCO. CIRCO includes multiple valid targets per query, making ranking quality particularly important for composed retrieval. DeCIR consistently achieves the best mAP across all evaluation cutoffs with both ViT-B/32 and ViT-L/14 backbones. With ViT-B/32, DeCIR improves over the strongest prior method by 0.8%, 0.7%, 0.6%, and 0.6% on mAP@5, mAP@10, mAP@25, and mAP@50, respectively. The gains become more pronounced with ViT-L/14, where DeCIR achieves 21.3% mAP@5 and improves over DistillCIR by 2.6%. Across all cutoffs, DeCIR outperforms the best competing method by 2.6%, 2.2%, 1.9%, and 1.8% on mAP@5, mAP@10, mAP@25, and mAP@50, respectively. These consistent improvements indicate that the proposed transition modeling is particularly effective for open-domain composed retrieval, where the target is determined by fine-grained semantic changes rather than category-level similarity alone.

FashionIQ. FashionIQ focuses on fine-grained attribute modifications within the fashion domain, covering categories such as dresses, shirts, and toptees. DeCIR obtains competitive or superior performance across the three fashion categories. With ViT-B/32, DeCIR achieves the best average results, reaching 29.7% R@10 and 49.8% R@50, while also producing clear gains on Shirt and TopTee. With ViT-L/14, DeCIR further improves the average performance to 30.4% R@10 and 51.6% R@50, achieving the best average performance among all compared methods. These results suggest that the learned semantic transition transfers beyond the natural-image CC3M training distribution and provides a more robust training signal for zero-shot composed retrieval in fine-grained scenarios.

CIRR. CIRR contains real-world natural scenes where the instruction must disambiguate the target from visually related candidates. As shown in Table 2, DeCIR reaches 33.4% R@1 with ViT-L/14. On the subset protocol, where the target is retrieved from six curated candidates, DeCIR achieves 67.1% R_s@1. These results suggest that explicit semantic transition alignment is helpful when retrieval depends on fine-grained edits rather than coarse visual similarity alone.

Table 2: **Quantitative comparison on the CIRR test set.** We compare our DeCIR with state-of-the-art methods across ViT-B/32 and ViT-L/14 backbones. Best results are in bold, second-best results are underlined, and our method is highlighted in grey.

	Method	Recall@k			RecallSubset@k		
		R@1	R@5	R@10	R_s@1	R_s@2	R_s@3
ViT B/32	SEARLE	24.0	53.4	66.8	54.9	76.6	88.2
	Slerp	28.2	55.9	68.8	61.1	80.6	90.7
	CIReVL	23.9	52.5	66.0	60.2	80.1	90.2
	DistillCIR	<u>29.9</u>	<u>58.8</u>	<u>70.3</u>	<u>62.3</u>	<u>82.5</u>	<u>90.8</u>
	DeCIR (ours)	**30.7**	**59.8**	**72.4**	**63.1**	**83.3**	**91.3**
ViT L/14	Pic2Word	23.9	51.7	65.3	53.8	74.5	87.1
	SEARLE	24.2	52.5	66.3	53.8	75.0	88.2
	LinCIR	25.0	53.3	66.7	57.1	77.4	88.9
	Context-I2W	25.6	55.1	68.5	-	-	-
	Slerp	30.9	59.4	70.9	64.7	82.9	**92.3**
	CIReVL	24.6	52.3	64.9	59.5	79.9	89.7
	MOA	27.1	56.5	69.2	-	-	-
	HIT	27.9	57.6	70.5	-	-	-
	DistillCIR	<u>32.3</u>	<u>63.6</u>	<u>74.3</u>	<u>66.8</u>	<u>84.0</u>	91.8
	DeCIR (ours)	**33.4**	**64.4**	**74.8**	**67.1**	**84.7**	<u>92.2</u>

4.3 Ablation Study

We evaluate the contribution of the core components in DeCIR. We first test whether endpoint alignment and transition alignment should be trained jointly or decoupled. We then compare merge and adaptation variants. Unless otherwise specified, we use ViT-L/14 as the backbone and report validation set results. Additional ablations and analyzes are provided in Appendix C.

Effect of Decoupled Training To understand the contribution of each training branch, we compare five settings: transition-only training with $\mathcal{L}_{\text{trans}}$, retrieval-only training with $\mathcal{L}_{\text{end}}$, joint training in one shared adapter, joint training with PCGrad (Yu et al., 2020), and the full decoupled approach followed by LRDM. All variants in Table 3 use the same backbone and paired supervision source; the PCGrad row applies gradient projection to the shared-adapter joint objective.

Table 3: **Ablation study on decoupling endpoint and semantic transition learning.** All variants are trained with the same backbone and data, differing only in the optimization and merging strategy for endpoint alignment and semantic transition alignment.

Method	$\mathcal{L}_{\text{end}}$	$\mathcal{L}_{\text{trans}}$	Inference	CIRR	CIRCO	GeneCIS
				R_s@1	mAP@50	Avg. R@1
Transition only	×	✓	transition branch	54.92	13.27	13.20
Endpoint only	✓	×	endpoint branch	64.46	24.92	16.50
Joint training	✓	✓	shared adapter	61.09	21.93	16.27
Joint training (PCGrad)	✓	✓	shared adapter	64.20	24.30	16.43
Decoupled + LRDM	✓	✓	merged adapter	66.08	27.26	16.86

Table 3 reveals three distinct patterns that validate our architectural design. First, transition-only training performs substantially worse than retrieval-only training. This is theoretically expected, as $\mathcal{L}_{\text{trans}}$ is designed to force instruction embeddings to capture relative semantic displacements rather than optimizing full composed queries to locate absolute gallery endpoints. Thus, semantic transition learning acts as a complementary signal to enrich endpoint alignment, rather than a standalone retrieval objective. Second, naively combining these objectives highlights a critical optimization bottleneck: shared-adapter joint training actively underperforms the retrieval-only baseline. This degradation indicates severe gradient interference when semantic transition supervision and endpoint alignment compete for the same low-rank parameter space. PCGrad (Yu et al., 2020) improves joint training, but its CIRCO result remains below the retrieval-only baseline. Third, our proposed strategy—decoupled training followed by LRDM—achieves the best results on CIRR and CIRCO while maintaining the strongest overall balance. This conclusively demonstrates that semantic transition supervision is most effective when isolated to allow branch specialization during training and systematically integrated only after convergence.

Table 4: **Merge strategy comparison.** All merge weights are set to 0.5.

Merge strategy	CIRR R_s@1	CIRCO mAP@50
Task Arithmetic (Ilharco et al., 2023)	65.22	27.07
TIES (Yadav et al., 2023)	63.29	26.70
DARE (Yu et al., 2024)	64.96	27.05
DARE-TIES (Yu et al., 2024; Yadav et al., 2023)	63.38	26.75
Model Breadcrumbs (Davari & Belilovsky, 2024)	64.41	27.14
RobustMerge (Zeng et al., 2025)	65.20	27.06
LRDM	66.08	27.26

Effect of LRDM LRDM integrates the transition branch into a single deployable low-rank adapter. Table 4 compares LRDM with representative generic merge rules. Task Arithmetic merges branches trained with separate low-rank bases, whereas LRDM uses a shared basis. LRDM performs best on both validation metrics, supporting the asymmetric merge design: the endpoint alignment branch preserves the endpoint-aligned backbone, while the transition branch injects semantic transition coefficients in the shared low-rank basis. Appendix C reports the full merge table with additional rules and metrics, plus merge-weight sensitivity and trainable component scope.

Figure 4 shows that DeCIR better preserves reference cues while applying the edit, whereas Pic2Word can follow target-side cues but lose source evidence.

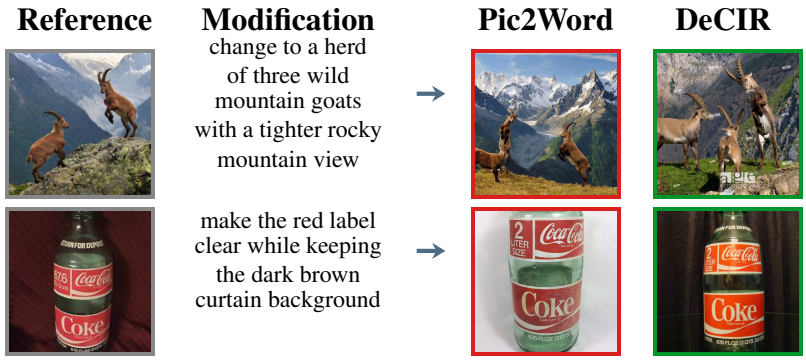

Figure 4: **Qualitative comparison on CIRR.** Pic2Word can follow parts of the modification while dropping source-conditioned evidence; DeCIR better preserves reference cues while applying the requested change.

5 Limitations

DeCIR relies on an offline LLM-generated supervision pipeline. Although the LLM is not used at inference, training still depends on the quality and diversity of generated edit tuples, and investigating richer or more visually grounded instruction sources is an important future direction. This paper focuses on projection-based ZS-CIR for its lightweight deployment advantages; extending conflict-aware endpoint–transition decoupling to LLM-based CIR pipelines, stronger vision-language backbones, or fully supervised CIR settings remains to be systematically studied.

6 Conclusion

Projection-based ZS-CIR is fundamentally limited by weak supervision for source-conditioned semantic transitions, which often causes the model to collapse into an endpoint shortcut. DeCIR addresses this semantic transition bottleneck through conflict-aware endpoint-transition decoupling: paired supervision explicitly provides semantic transition signals, decoupled learning separates endpoint and transition alignment, and LRDM restores a single lightweight inference path. Consequently, DeCIR significantly strengthens projection-based CIR without relying on inference-time LLMs.

AI USE STATEMENT

We used `glm-4-flash` to generate synthetic forward edits, modified target captions, and reverse edits for offline training data construction, as described in Section 3.2 and Appendix B. Generated records were format-filtered, and random samples were manually reviewed for edit fidelity and reverse-instruction consistency. We used GPT-5.6-Luna to screen tuple quality, with external annotators reviewing uncertain cases. We also used generative AI to polish the manuscript. The authors are responsible for the paper's claims and final content.

REFERENCES

Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Effective conditioned and composed image retrieval combining clip-based features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21466–21474, 2022.

Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Alberto Del Bimbo. Zero-shot composed image retrieval with textual inversion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15338–15347, 2023a.

Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Composed image retrieval using contrastive learning and task-oriented clip-based features. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(3):1–24, 2023b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020a.

Yanbei Chen, Shaogang Gong, and Loris Bazzani. Image search with text feedback by visiolinguistic attention learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3001–3011, 2020b.

MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision*, pp. 270–287. Springer, 2024.

Geonmo Gu, Sanghyuk Chun, Wonjae Kim, HeeJae Jun, Yoohoon Kang, and Sangdoo Yun. Compodiff: Versatile composed image retrieval with latent diffusion. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. URL https://openreview.net/forum?id=mKtlzW0bWc. Expert Certification.

Geonmo Gu, Sanghyuk Chun, Wonjae Kim, Yoohoon Kang, and Sangdoo Yun. Language-only training of zero-shot composed image retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13225–13234, 2024b.

Jitai Hao, Hao Liu, Xinyan Xiao, Qiang Huang, and Jun Yu. Uni-x: Mitigating modality conflict with a two-end-separated architecture for unified multimodal models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IJLIYpCkwz.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chuong Huynh, Jinyu Yang, Ashish Tawari, Mubarak Shah, Son Tran, Raffay Hamid, Trishul Chilimbi, and Abhinav Shrivastava. Collm: A large language model for composed image retrieval. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3994–4004, 2025.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Young Kyun Jang, Dat Huynh, Ashish Shah, Wen-Kai Chen, and Ser-Nam Lim. Spherical linear interpolation and text-anchoring for zero-shot composed image retrieval. In *European conference on computer vision*, pp. 239–254. Springer, 2024.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

Shyamgopal Karthik, Karsten Roth, Massimiliano Mancini, and Zeynep Akata. Vision-by-language for training-free compositional image retrieval. In *The Twelfth International Conference on Learning Representations*. International Conference on Learning Representations, ICLR, 2024.

Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19113–19122, 2023.

Jaehyun Kwak, Ramahdani Muhammad Izaaz Inhar, Se-Young Yun, and Sung-Ju Lee. Qure: Query-relevant retrieval through hard negative sampling in composed image retrieval. In *Forty-second International Conference on Machine Learning*, 2025.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Wei Li, Hehe Fan, Yongkang Wong, Yi Yang, and Mohan Kankanhalli. Improving context understanding in multimodal large language models via multimodal composition learning. In *International conference on machine learning*, 2024.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

You Li, Fan Ma, and Yi Yang. Imagine and seek: Improving composed image retrieval with an imagined proxy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3984–3993, 2025a.

Zhe Li, Lei Zhang, Zheren Fu, Kun Zhang, and Zhendong Mao. Hierarchy-aware pseudo word learning with text adaptation for zero-shot composed image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24319–24329, 2025b.

Zhe Li, Lei Zhang, Kun Zhang, Weidong Chen, Yongdong Zhang, and Zhendong Mao. Rethinking pseudo word learning in zero-shot composed image retrieval: From an object-aware perspective. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 833–843, 2025c.

Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2125–2134, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

Bill Psomas, Ioannis Kakogeorgiou, Nikos Efthymiadis, Giorgos Tolias, Ondřej Chum, Yannis Avrithis, and Konstantinos Karantzalos. Composed image retrieval for remote sensing. In *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium*, pp. 8526–8534. IEEE, 2024.

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

Kuniaki Saito, Kihyuk Sohn, Xiang Zhang, Chun-Liang Li, Chen-Yu Lee, Kate Saenko, and Tomas Pfister. Pic2word: Mapping pictures to words for zero-shot composed image retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19305–19314, 2023.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, 2018.

Shitong Sun, Qilei Li, Shaogang Gong, Weitong Cai, Philip Torr, and Jindong Gu. Benchcir: Benchmarking robustness in composed image retrieval across modalities. *Pattern Recognition*, pp. 113724, 2026.

Zelong Sun, Dong Jing, and Zhiwu Lu. Cotmr: chain-of-thought multi-scale reasoning for training-free zero-shot composed image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22675–22684, 2025.

Yucheng Suo, Fan Ma, Linchao Zhu, and Yi Yang. Knowledge-enhanced dual-stream zero-shot composed image retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26951–26962, 2024.

Yuanmin Tang, Jing Yu, Keke Gai, Jiamin Zhuang, Gang Xiong, Yue Hu, and Qi Wu. Context-i2w: Mapping images to context-dependent words for accurate zero-shot composed image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5180–5188, 2024.

Yuanmin Tang, Jing Yu, Keke Gai, Jiamin Zhuang, Gang Xiong, Gaopeng Gou, and Qi Wu. Missing target-relevant information prediction with world model for accurate zero-shot composed image retrieval. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24785–24795, 2025a.

Yuanmin Tang, Jue Zhang, Xiaoting Qin, Jing Yu, Gaopeng Gou, Gang Xiong, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Wu. Reason-before-retrieve: One-stage reflective chain-of-thoughts for training-free zero-shot composed image retrieval. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14400–14410, 2025b.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Jingyu Sun, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024.

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

Tianyue Wang, Leigang Qu, Tianyu Yang, Xiangzhao Hao, Yifan Xu, Haiyun Guo, and Jinqiao Wang. Wiser: Wider search, deeper thinking, and adaptive fusion for training-free zero-shot composed image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16865–16875, 2026.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. 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*, pp. 11307–11317, 2021.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.

Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. Robustmerge: Parameter-efficient model merging for mllms with direction robustness. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Wenliang Zhong, Rob Barton, Weizhi An, Feng Jiang, Hehuan Ma, Yuzhi Guo, Abhishek Dan, Shioulin Sam, Karim Bouyarmane, and Junzhou Huang. Zero-shot composed image retrieval via dual-stream instruction-aware distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22221–22231, 2025.

A Additional Benchmark Results

Table 5 provides the detailed GeneCIS comparison across four conditional retrieval settings. This benchmark checks whether DeCIR only improves natural-image CIR benchmarks or also transfers to conditional retrieval where the model must either focus on or change attributes and objects. DeCIR obtains the strongest average R@1 under both ViT-B/32 and ViT-L/14. While it is not uniformly best in every subcondition, the balanced average supports our claim that semantic transition learning improves source-conditioned retrieval rather than overfitting to one edit type.

Table 5: **Quantitative comparison on the test set of GeneCIS.** We report the R@K metrics across ViT-B/32 and ViT-L/14 backbones. Best results are in bold, second-best results are underlined, and our method is highlighted in grey.

	Method	Focus Attribute			Change Attribute			Focus Object			Change Object			Average
		R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	R@1
ViT-B/32	SEARLE (ICCV'23)	18.9	30.6	41.2	13.0	23.8	33.7	12.2	23.0	33.3	13.6	23.8	33.3	14.4
	CIReVL (ICLR'24)	17.9	29.4	40.4	14.8	25.8	35.8	14.6	24.3	33.3	16.1	27.8	37.6	15.9
	DistillCIR (ICCV'25)	**20.0**	31.0	42.5	**15.2**	**26.9**	35.9	13.2	23.8	33.0	16.7	28.2	37.1	16.3
	DeCIR (ours)	**20.0**	**32.4**	**43.5**	14.0	25.7	**36.8**	**14.9**	**26.7**	**36.1**	**17.2**	**28.5**	**38.5**	**16.5**
ViT-L/14	Pic2Word (CVPR'23)	15.7	28.2	38.7	13.9	24.7	33.1	8.4	18.0	25.8	6.7	15.1	24.0	11.2
	SEARLE (ICCV'23)	17.1	29.6	40.7	16.3	25.2	34.2	12.0	22.2	30.9	12.0	24.1	33.9	14.4
	LinCIR (CVPR'24)	16.9	30.0	41.5	16.2	28.0	36.8	8.3	17.4	26.2	7.4	15.7	25.0	12.2
	Context-I2W (AAAI'24)	17.2	30.5	41.7	16.4	28.3	37.1	8.7	17.9	26.9	7.7	16.0	25.4	12.5
	CIReVL (ICLR'24)	19.5	31.8	42.0	14.4	26.0	35.2	12.3	21.8	30.5	17.2	**28.9**	37.6	15.9
	PrediCIR (CVPR'25)	18.2	31.9	42.6	**18.7**	**30.4**	35.4	12.7	19.0	31.2	16.9	25.5	34.1	16.6
	DistillCIR (ICCV'25)	20.7	32.6	**44.2**	16.1	27.6	37.7	13.3	**24.4**	33.6	**17.3**	28.8	**37.7**	16.8
	DeCIR (ours)	**21.1**	**32.7**	43.1	16.4	28.3	**39.0**	**14.4**	23.4	**33.8**	16.3	27.2	36.9	**17.1**

B Supervision Construction and Training Details

This section documents how DeCIR obtains semantic transition supervision without annotated CIR triplets. Each CC3M image-caption pair is converted into a structured tuple containing a forward instruction, a modified target caption, and a reverse instruction. The target caption provides the endpoint used by endpoint alignment, while the forward and reverse instructions provide opposite semantic transition signals. The LLM is used only for offline supervision construction and is not used during inference.

We construct the supervision offline with `glm-4-flash` through the ZhipuAI batch `/v4/chat/completions` API, with the temperature set to 0.7. For each source caption, the prompt asks the LLM to choose exactly one plausible visual change, apply that change while preserving unrelated source details, and write a reverse instruction that undoes only the forward edit. We parse the batch outputs as JSON and discard malformed records or outputs missing any required fields. This format filtering leaves 667,229 format-valid tuples. The `brainstorming` field is used only to make generation more controlled and is not used by any training loss. Code and generated supervision will be released after acceptance.

We additionally screened the generated tuples using GPT-5.6-Luna for source–target consistency, instruction validity, and semantic ambiguity. External annotators reassessed cases that the model

could not judge reliably. The combined audit identified 2.8% of the screened tuples as erroneous, mainly due to ambiguous instructions.

Table 6 shows generated forward edits, target captions, and reverse instructions. Each tuple changes one visual attribute while preserving unrelated source details, and the reverse instruction undoes that edit.

Figure 5 gives the prompt used to generate each tuple. It requires valid JSON, one significant visual change, preservation of unrelated details, and a reverse instruction that undoes only that change.

Table 6: **Randomly sampled tuples.** Each example applies one visual change, preserves unrelated source evidence in the target caption, and provides a reverse instruction for the opposite transition.

Source evidence	Forward tuple	Reverse instruction
open red umbrella casting a shadow	**Forward:** Change the color of the umbrella from red to blue. **Target:** An open blue umbrella with a shadow.	Change the color of the umbrella from blue to red.
worker clearing snow off the sidewalk with a broom	**Forward:** Replace the broom with a shovel for clearing snow. **Target:** A worker clears snow off the sidewalk with a shovel.	Replace the shovel with a broom for clearing snow.
dog standing on a balcony and looking down	**Forward:** Change the dog's posture from standing to sitting. **Target:** A dog sits on a balcony looking down.	Change the dog's posture from sitting to standing.
young man struggling to row through a river	**Forward:** Change the river to a lake. **Target:** A young man struggles to row through a lake.	Change the lake to a river.
statue of a builder at a shrine, originally stone	**Forward:** Change the material of the statue from stone to gold. **Target:** A gold statue of a builder at the shrine.	Change the material of the statue to stone.

You are helping to create a multimodal dataset for Composed Image Retrieval (CIR). Given one source image caption, create one plausible visual edit from source to target, a modified target caption, and the reverse instruction that would transform the modified caption back to the source.

Task
1. Input: A source image caption will be provided.
2. Brainstorming: Identify the key source details in the source caption (objects, actions, setting) and propose one significant, plausible visual change.
3. `instruction`: Write the forward edit from source to target.
4. `modified_caption`: Apply the edit while preserving unrelated source details.
5. `reverse_instruction`: Write the reverse edit from `modified_caption` back to the source.

Output Requirements
Return valid JSON only with exactly these keys:
1. `"brainstorming"` – Briefly explain the source details and the proposed change.
2. `"instruction"` – A short statement of the exact source-to-target change.
3. `"modified_caption"` – The new target caption after applying the edit.
4. `"reverse_instruction"` – A short instruction that undoes only the forward edit.

Important
1. Make exactly one significant visual change, such as changing an object color, location, action, material, count, or category.
2. Keep unrelated details unchanged in `modified_caption`.
3. The `reverse_instruction` must undo only the forward edit.
4. The instruction and modified caption should be coherent and plausible.

Input: {source_caption}

Figure 5: **Prompt template used for offline supervision construction.** We prompt the LLM to produce the forward edit, edited caption, and reverse instruction in one structured output.

Table 7: **Main DeCIR training hyperparameters.** Common optimization settings follow the DistillCIR-style setup where applicable, while DeCIR-specific values such as the LRDM merge weight and source-anchor image weight are shown explicitly.

Configuration	Value	Configuration	Value
Visual tower	CLIP ViT	Optimizer	AdamW
Text tower	CLIP text encoder	Learning rate	2×10^{-5}
Format-valid generated tuples	667,229	Weight decay	0.1
LoRA rank	64	Warmup	200 steps
LoRA alpha	16	LR scheduler	Cosine
Precision	FP16	LRDM merge weight α	0.50
Training epochs	1	Source-anchor weight ω	0.25

As shown in Table 7, we list the main training configuration used for DeCIR. We follow the DistillCIR-style training setup where applicable to keep the comparison focused on the proposed endpoint–transition decoupling. DeCIR-specific choices, including the number of format-valid generated tuples, LRDM merge weight α, and source-anchor image weight ω, are reported explicitly.

Algorithm 1 summarizes the training procedure of DeCIR and the LRDM merge. The key implementation detail is update ownership: the endpoint alignment branch updates the endpoint-aligned retrieval pathway and the shared low-rank bases, while the transition alignment branch computes retrieval-space anchors, stops gradients through these anchors, and updates only the transition coefficients A_{trans}. This enables training-time decoupling while keeping the two branches in a shared coefficient space for LRDM.

Algorithm 1 DeCIR training and LRDM merge.

Require: Batch $\mathcal{B} = \{(I_{\text{ref}}, c_{\text{src}}, t_{\text{fwd}}, c_{\text{tgt}}, t_{\text{rev}})\}$; shared bases $\{B^\ell\}$; branch coefficients $\{A^\ell_{\text{end}}, A^\ell_{\text{trans}}\}$; merge weight α

Training phase

for each mini-batch $\mathcal{B}$ **do**
  Build composed queries with E_T^{end}, E_V, and f_ϕ using $(I_{\text{ref}}, t_{\text{fwd}})$.
  Compute $\mathcal{L}_{\text{end}}$ against target caption embedding $f_{\text{tgt}}^{\text{end}}$.
  Update $\{A^\ell_{\text{end}}\}$, shared $\{B^\ell\}$, visual LoRA parameters, f_ϕ, and τ.
  Compute f_{src}, $f_{\text{tgt}}^{\text{trans}}$, and δ and detach.
  Encode t_{fwd} and t_{rev} with E_T^{trans} and compute $\mathcal{L}_{\text{trans}}$.
  Update only $\{A^\ell_{\text{trans}}\}$.
end for

Merge phase

for each adapted text layer ℓ **do**
  $A^\ell_{\text{merge}} \leftarrow (1-\alpha)A^\ell_{\text{end}} + \alpha A^\ell_{\text{trans}}, \quad B^\ell_{\text{merge}} \leftarrow B^\ell$
  $W'^\ell \leftarrow W^\ell + B^\ell_{\text{merge}} A^\ell_{\text{merge}}$
end for
return final retrieval model $\{E_V, f_\phi, E_T(W')\}$

C Additional Ablations

To test whether DeCIR transfers beyond Pic2Word, we integrate it with two other projection-based methods, SEARLE and LinCIR. Table 8 shows that DeCIR improves CIRR R@1 by 8.22 and 5.10 points, respectively, and also improves average GeneCIS R@1 for both methods. This suggests that the training scheme is not limited to the Pic2Word initialization.

We ablate the two directions of transition supervision while keeping the other settings fixed. Table 9 shows that combining forward and reverse instructions achieves the best result on all four reported metrics, supporting the use of paired directions in transition alignment.

Table 8: **Transfer of DeCIR to SEARLE and LinCIR on CIRR and GeneCIS.** The without-DeCIR scores are the corresponding baselines in the integration experiments.

Method	Setting	CIRR R@1	GeneCIS Avg. R@1
SEARLE	without DeCIR	22.91	13.04
	with DeCIR	**31.13**	**17.15**
LinCIR	without DeCIR	24.68	14.70
	with DeCIR	**29.78**	**17.17**

Table 9: **Forward and reverse transition-supervision ablation on validation sets.** All other training settings are held fixed.

Transition supervision	CIRR R@1	CIRR R@10	CIRR R_s@1	GeneCIS Avg. R@1
Forward only	30.97	74.43	64.60	16.48
Reverse only	30.81	74.33	65.34	16.54
Forward + reverse	**31.97**	**74.79**	**66.08**	**16.86**

Table 10 compares LRDM with off-the-shelf model merging rules, including Task Arithmetic (Ilharco et al., 2023), TIES (Yadav et al., 2023), DARE (Yu et al., 2024), Model Breadcrumbs (Davari & Belilovsky, 2024), and RobustMerge (Zeng et al., 2025). LRDM performs best across all three validation metrics, showing that the gain is not simply due to arbitrary parameter averaging. Instead, it supports the asymmetric design of LRDM: the endpoint alignment branch preserves the deployable endpoint-aligned backbone, while the transition branch contributes semantic transition coefficients in the shared low-rank basis.

Figure 6 tests whether the failure of shared-adapter joint training is merely a loss-weighting issue. We optimize the same shared adapter with $\mathcal{L}_{\text{end}} + \lambda_{\text{trans}}\mathcal{L}_{\text{trans}}$ and sweep λ_{trans} on the CIRR validation set. If the problem were only a loss-scale imbalance, some weights should approach the decoupled result in Table 3. Instead, increasing the transition weight generally degrades retrieval, and the shared-adapter objective does not close the gap to the decoupled learning. This supports our claim that semantic transition supervision should be integrated through conflict-aware decoupling rather than treated as a standard auxiliary loss.

Table 10: **Merge strategy comparison on validation sets.** All merge weights are set to 0.5. LRDM outperforms generic merge rules, supporting shared-basis coefficient merging rather than generic weight-space merging.

Merge strategy	CIRR R_s@1	CIRCO mAP@50	GeneCIS Avg. R@1
Task Arithmetic (Ilharco et al., 2023)	65.22	27.07	16.57
TIES (Yadav et al., 2023)	63.29	26.70	16.49
TIES (frequency) (Yadav et al., 2023)	63.45	26.86	16.31
DARE (linear) (Yu et al., 2024)	64.96	27.05	16.55
DARE+TIES (Yu et al., 2024; Yadav et al., 2023)	63.38	26.75	16.45
Model Breadcrumbs (Davari & Belilovsky, 2024)	64.41	27.14	16.55
RobustMerge (Zeng et al., 2025)	65.20	27.06	16.60
LRDM	66.08	27.26	16.86

Table 11 isolates the trainable component scope while keeping LRDM fixed. Text-adapter-only training underperforms the setting that also adapts the visual encoder and trains the full inversion network. This suggests that source-conditioned composed retrieval benefits from adapting the pathway that exposes reference-image evidence to the text encoder, not only the text adapter that receives the modification. We therefore use LoRA on both CLIP towers and train the full inversion network.

Figure 7 studies two DeCIR-specific weights. The LRDM weight α controls how much transition-branch coefficient information is injected into the endpoint alignment branch. The endpoints of the sweep are informative: $\alpha = 0$ reduces to retrieval-only inference, while $\alpha = 1$ uses pure transition coefficients. The best performance occurs at an intermediate value, supporting our view that semantic

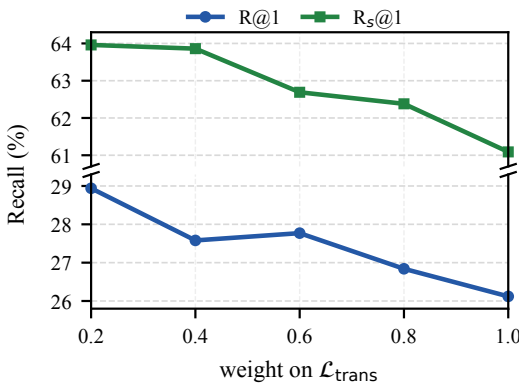

Figure 6: **Shared-adapter transition-loss weight sweep.** We sweep λ_{trans} in $\mathcal{L}_{\text{end}} + \lambda_{\text{trans}}\mathcal{L}_{\text{trans}}$ on CIRR validation. The sweep supports that joint underperformance is not simply due to an untuned transition-loss scale.

transition learning is complementary residual knowledge rather than a standalone low-rank adapter. The source-anchor weight ω controls how much image-conditioned evidence enters the source anchor. The sensitivity curve shows that incorporating visual source evidence is useful, while an overly image-dominated anchor is not necessary.

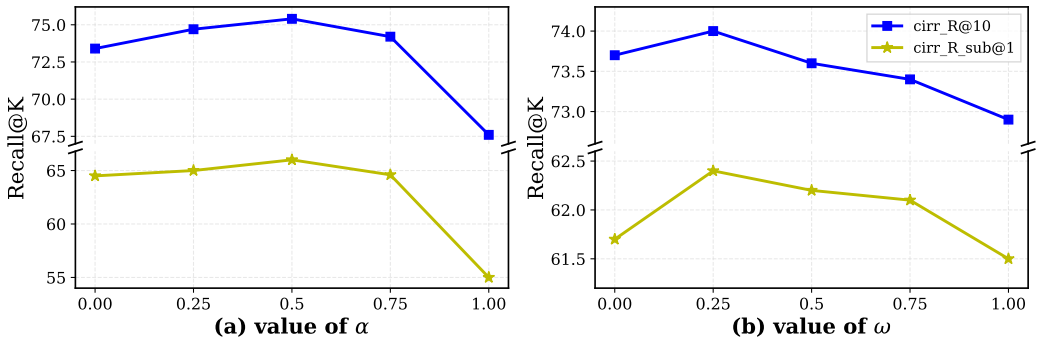

Figure 7: **Parameter sensitivity analysis on CIRR validation set.** The α sweep shows that transition coefficients work best when injected into an endpoint alignment branch rather than used alone. The ω sweep uses ViT-B/32 and studies how much image-conditioned source evidence should enter the transition anchor.

Table 11: **Trainable component ablation on validation sets.** The merge rule is fixed to LRDM; only the trainable component scope changes. This implementation ablation justifies the final adaptation scope used by DeCIR.

Trainable components	CIRR R$_s$@1	CIRCO mAP@50	GeneCIS Avg. R@1
LoRA on E_T	60.11	23.10	14.88
LoRA on E_T, full on f_ϕ	60.32	23.97	14.94
LoRA on (E_V, E_T), full on f_ϕ	66.08	27.26	16.86

D Deployment Profile

Table 12: **Deployment profile in the ViT-L/14 setting.** DeCIR uses LLM supervision only offline during training. After LRDM, it deploys a single merged retrieval model with no LLM or second text branch at inference.

Method	LLM at inference	Trainable params	Inference params
Context-I2W (AAAI'24)	×	65.3M	493M
CIReVL (ICLR'24)	✓	–	12.5B
MCL (ICML'24)	✓	25.8M	7.4B
DistillCIR (ICCV'25)	×	42M	467M
WISER (CVPR'26)	✓	–	≥33.4B
DeCIR (ours)	×	50M	467M

Table 12 compares the deployment footprint of DeCIR with representative ZS-CIR systems, including Context-I2W (Tang et al., 2024), CIReVL (Karthik et al., 2024), MCL (Li et al., 2024), DistillCIR (Zhong et al., 2025), and WISER (Wang et al., 2026). DeCIR uses LLM-generated supervision only offline during training; after LRDM, inference uses one visual encoder, one inversion network, and one merged text adapter. Thus, its active inference parameter count remains close to projection-based methods such as Context-I2W and DistillCIR, while avoiding the online LLM or reasoning modules used by heavier systems. Combined with the benchmark results in the main text, this supports the claim that DeCIR improves the projection-based accuracy–efficiency trade-off rather than trading accuracy for an inference-time reasoning loop.

E Qualitative Analysis

Figures 8 and 9 compare CIRR and FashionIQ cases. DeCIR more consistently preserves source evidence while applying the requested edits.

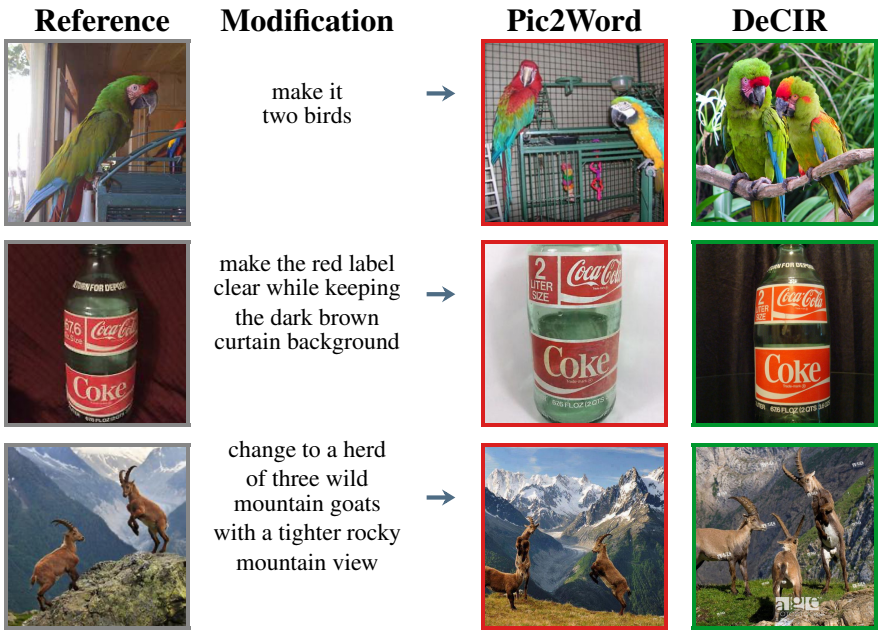

Figure 8: **Qualitative comparison on CIRR.** The Pic2Word can follow parts of the modification but drop source-conditioned evidence. DeCIR retrieves images that satisfy the modification while preserving the reference constraint.



Figure 9: **Qualitative comparison on FashionIQ.** The examples cover fashion modifications involving coverage, sleeve length, color, logos, and style. DeCIR improves edit sensitivity while preserving source garment evidence. Green boxes indicate correct top-1 retrievals, and red boxes indicate incorrect top-1 retrievals.