SkillCIR: Intent-Guided Skill Composition for Training-Free Composed Image Retrieval

Yuanmin Tang[1] **Lin Li**[2] **Yang Du**[2] **Yuan Gao**[2] **Massimiliano Mancini**[3]
Jun Song[2] **Gaopeng Gou**[1] **Gang Xiong**[1*] **Meikang Qiu**[4] **YuCheng**[2] **Bo Zheng**[2]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]Alibaba Group [3]University of Trento [4]Augusta University

Abstract

Composed Image Retrieval (CIR) aims to retrieve a target image given a reference image and a textual modification. Existing training-free methods prompt a frozen multimodal large language model (MLLM) to infer the edit intent and rewrite the composed query into a target caption or an edited image. However, collapsing heterogeneous edit constraints into a global retrieval representation leaves limited means to execute each constraint during ranking. We argue that the bottleneck of training-free CIR lies not only in intent understanding, but also in intent execution. To address this *intent execution gap*, we propose `SkillCIR`, a training-free framework that routes each inferred constraint to a specialized retrieval skill. Specifically, the frozen MLLM predicts a structured constraint plan that activates a query-specific subset of typed skills; each skill scores gallery candidates along one evidence channel, and the resulting scores are combined with explicit signs and checked by local verification. Across three benchmarks and three CLIP backbones, `SkillCIR` improves the primary metrics by 2.8 to 7.9 points over the prior training-free state of the art, without any parameter updates. Code is available at `https://github.com/Pter61/skillcir`.

1 Introduction

Given a reference image and a textual edit from a user, composed image retrieval (CIR) returns the corresponding target image from a gallery. CIR is attractive in any scenario requiring interactive visual search: the reference image anchors what should be preserved, while the text specifies what should change. CIR queries are also diverse, ranging from local attribute edits in fashion to entity removal and scene rewrites in open-domain photography [5, 27, 47]. Training a dedicated CIR model for every domain is costly and often brittle, due to the need for large-scale triplet annotations (*i.e.,* reference image, edit text, target image).

Recent training-free CIR methods have made substantial progress by using MLLMs to reason about the modification. However, most pipelines use this reasoning mainly to produce a target caption [21, 41] or edited image, and even approaches that incorporate object cues [38], include/exclude reranking [26], or dual-branch verification [44] do not route each inferred constraint type to a dedicated scoring channel. We argue that the remaining bottleneck lies less in intent *understanding* than in intent *execution*: we call this the **intent execution gap**, the discrepancy between the fine-grained constraints a frozen MLLM can infer and what the retrieval representation explicitly acts on at ranking time. Figure 1(a) illustrates this gap on a representative query: the user asks to "remove the seashells and make the water green" on a stingray scene, and the MLLM correctly infers a *remove* + *color* + *preserve* intent together with the target description "a stingray underwater in green water without seashells", yet caption-cosine retrieval returns a stingray whose water is still blue. The intent is

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

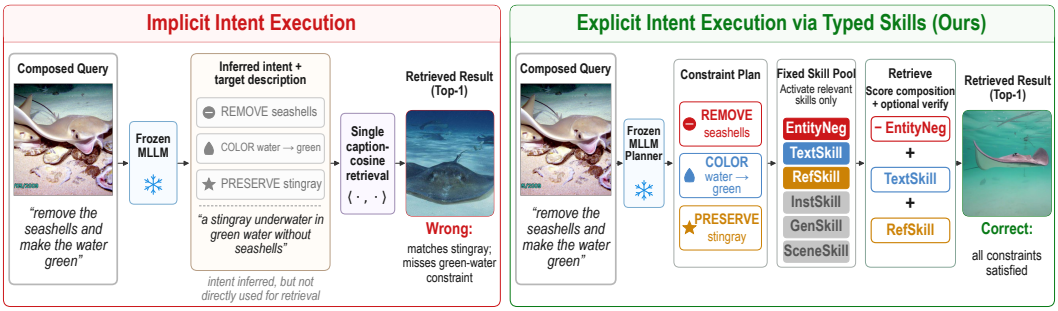


Figure 1: Illustration of our motivation. **(a)** Implicit intent execution in caption-based training-free CIR: the MLLM infers *remove+color+preserve*, but the constraints collapse into a single caption similarity and a blue-water stingray is returned. **(b)** Our `SkillCIR` with explicit intent execution via typed skills: each constraint activates a typed skill (`EntityNeg`, `TextSkill`, `RefSkill`) and their signed composition returns the green-water target.

inferred but not directly used for retrieval: once the constraints are collapsed into one similarity, they no longer bias ranking individually. Similar challenges arise when an edit carries multiple heterogeneous constraints, *e.g., "change the dog breed and add a pirate hat"* (CIRCO) or *"maroon short-sleeved shirt with no graphics"* (FashionIQ).

To address this limitation, we propose `SkillCIR`, a training-free framework that tackles the intent execution gap by routing each inferred constraint to a specialized retrieval skill, as shown in Figure 1(b). A frozen MLLM analyzes the composed query and predicts a structured constraint plan; the plan activates a query-specific subset of a fixed skill pool induced once offline from pseudo-triplets. For the stingray example, the planner maps the three constraints to three skills: `EntityNeg` repels seashell evidence, `TextSkill` scores the green-water target description, and `RefSkill` preserves the stingray; their per-skill candidate scores are combined with explicit signs into a single ranking, returning the green-water target that caption-cosine missed. By keeping each constraint as a separately scoreable channel rather than collapsing them into one embedding, `SkillCIR` brings intent reasoning closer to intent execution. This design draws on late-interaction retrieval [22]: keeping per-channel evidence separate rather than averaging it into a single vector improves both multi-constraint matching and per-skill interpretability.

Contributions. To summarize: (1) We identify the *intent execution gap* in training-free CIR: standard rewrite-and-retrieve pipelines collapse heterogeneous edit constraints into global similarity matching rather than executing each constraint on its own evidence channel. (2) We propose `SkillCIR`, a training-free framework that tackles this gap by routing each inferred constraint to a typed retrieval skill, linearly combining per-skill candidate scores with explicit attract/repel signs, and re-checking high-ranked candidates with local verification operators. (3) Across three benchmarks and three CLIP backbones, `SkillCIR` improves the primary metrics by 2.8 to 7.9 points over the prior training-free state of the art without any parameter updates, reaching 52.51% average R@10 on FashionIQ, 41.80% mAP@5 on CIRCO, and 88.64% R@10 on CIRR.

2 Related Work

Composed Image Retrieval. Composed image retrieval (CIR) retrieves a target image from a query composed of a reference image and a text modification [42]. Supervised CIR methods [4, 9, 27] learn task-specific composition models from annotated triplets, while CompoDiff [15] trains a diffusion prior on synthetic triplets. Zero-shot CIR avoids target-domain annotations by mapping the reference image into pseudo-word tokens for CLIP retrieval [5, 16, 32, 39, 40], training an MLLM query embedder with dual-level reranking (UniCVR [46]), or using MLLM CoT to guide trained multi-level vision selection (MCoT-MVS [14]). Recent training-free methods use MLLMs to infer the modification: CIReVL [21] establishes the caption-modify-retrieve pattern, OSrCIR [41] uses one-stage reflective CoT, LDRE [49] ensembles divergent captions, CoTMR [38] combines target-caption similarity with positive and negative object scores, AutoCIR [8] uses an automatic multi-agent loop, Paracosm [45] generates a synthetic "mental image," and concurrent G-MIXER [26]

expands candidates with geodesic mixup and include/exclude re-ranking. WISER [44] is the closest training-free predecessor, generating both an edited caption and an edited image and adaptively fusing the T2I/I2I retrieval branches with candidate verification and refinement. While these works enrich target representations and incorporate fixed-pattern score fusion or verification, `SkillCIR` organizes retrieval around a query-specific plan over a fixed pool of typed skills, selectively activating and composing attractive, repulsive, local, and reference-relative evidence in candidate-score space.

Decomposed Retrieval and Pseudo-Triplets. Beyond single-vector matching, several retrieval paradigms decompose evidence into finer units. ColBERT-style late interaction [22, 33] represents queries and documents with multiple token vectors and scores them via max-sim operations, while mixture-of-experts models [11] route inputs to specialized learned experts. Both designs improve matching through decomposed evidence, but require trained retrievers or learned routers. VISA [10] uses visual abstraction and question answering to adapt gallery-side descriptions to the granularity of a text query, whereas our setting decomposes a reference-image edit into query-side candidate-scoring requirements. On the data side, MegaPairs [53] synthesizes millions of pseudo-triplets to train BGE-VL retrievers, and PTG-FSCIR [17] uses pseudo-triplets for CIR pretraining. `SkillCIR` uses pseudo-triplets differently: rather than training a retriever or router, it uses an external MegaPairs subset once offline to induce and audit a fixed pool of retrieval skills, and a frozen MLLM selects which skills fire per query at test time. Prior CIR work has also examined preservation and modification, entity–action relations, and edit categories such as addition, replacement, and negation [7, 25, 39, 48], which inform our six-intent taxonomy. Because the resulting skills carry explicit signs, score composition can directly repel candidates that violate removal or forbidden-entity constraints.

LLM Agents, Tool Use, and Skills. Tool-augmented LLMs generate calls or plans to invoke actions, APIs, or expert models [18, 35, 36, 50], and recent works further coordinate large-scale pretrained models or external knowledge in training-free and multimodal reasoning pipelines [12, 28]. Recent multimodal agents extend this to interactive visual settings: SeeAct [52] grounds GPT-4V plans into web actions, Vision Search Assistant [51] combines VLMs with web agents for open-world visual search, and MMAT-1M [13] studies multimodal agent tuning with CoT, reflection, and dynamic tool use. Skill-library agents such as Voyager [43] also show that reusable skills can make LLM behavior more compositional. These systems inspire our use of planning and skills, but their tools execute external actions or solve intermediate tasks. `SkillCIR` is more constrained: a single frozen MLLM call selects retrieval skills, and each skill outputs candidate scores over the same gallery rather than executing web actions or generating an external answer trace, with the skill pool fixed offline rather than expanded per query; we discuss a tool-backed `SearchSkill` extension in Appendix F.

3 Methodology

In this section we first formalize the training-free composed image retrieval problem and define a candidate score (§3.1). We then present `SkillCIR` as three operations: a fixed **Skill Pool** of typed retrieval evidence induced offline from external pseudo-triplets (§3.2); a **Planner** that uses a frozen MLLM to identify which skills the edit requires and emit a target description (§3.3); and **Score Composition and Verification** that combines the per-skill evidence into a final ranking (§3.4–§3.5).

3.1 Preliminary

Standard caption-based training-free CIR pipelines compress the composed query into a single target description and rank gallery candidates by one similarity score, leaving an *intent execution gap* between what the MLLM reasons about and what the retriever can act on. `SkillCIR` tackles this gap by keeping each inferred edit constraint as a separately scoreable channel, executing the inferred intent at the moment of ranking instead of consuming it at the captioning step.

Given a reference image I_r and a textual edit T_m, training-free CIR retrieves a target image I_t from a gallery $\mathcal{D}$ using a frozen CLIP [31] retriever (image encoder Ψ_I, text encoder Ψ_T mapping into a shared d-dimensional space) and a frozen multimodal language model Ψ_M that maps multimodal inputs to text, all without parameter updates [21, 41]. A caption-based baseline turns the composed query into a single target description $T_t = \Psi_M(I_r, T_m)$ and ranks candidates by cosine similarity:

$$I_t \;=\; \arg\max_{I \in \mathcal{D}} \frac{\Psi_I(I)^\top \Psi_T(T_t)}{\|\Psi_I(I)\| \cdot \|\Psi_T(T_t)\|}. \tag{1}$$

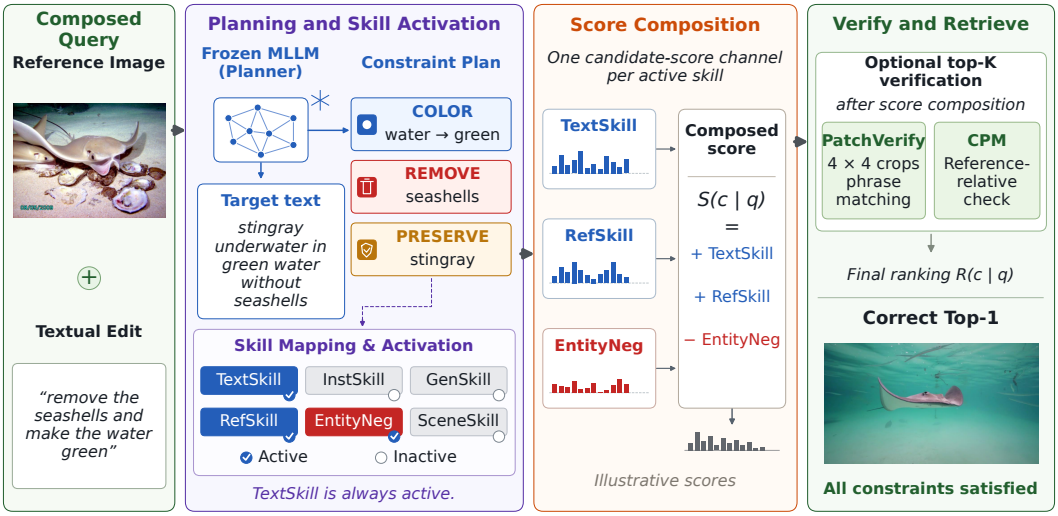


Figure 2: `SkillCIR` **overview. (1)** A frozen MLLM produces a target description and constraint plan. **(2)** The plan activates a subset of six skills induced offline from pseudo-triplets; for the stingray example, `TextSkill`, `RefSkill`, and `EntityNeg` are active. **(3)** Active CLIP cosine channels are combined with fixed signs (Eq. 3); `SceneSkill` also uses component signs. **(4)** Top-K candidates are re-checked by `PatchVerify` and `CPM`.

`SkillCIR` replaces single-cosine composition with score-space composition. For skill k and query q, $s_k(c \mid q) \in \mathbb{R}$ scores every candidate $c \in \mathcal{D}$ along one evidence channel; stacking over c gives one score vector per skill, and composition combines these score vectors. Figure 2 gives the overview, with operations detailed in §3.2–§3.5.

3.2 Skill Construction from Pseudo-Triplets

To construct intent-relevant skills while keeping the zero-shot setting, we induce the skill pool offline from external pseudo-triplets and freeze it before any benchmark is touched. We randomly sample a set $\mathcal{P}_s$ of 50K reference–target–edit pseudo-triplets from MegaPairs [53], an open-domain corpus external to FashionIQ, CIRR, and CIRCO and not constructed from their retrieval labels. GPT-4o [29] identifies recurring edit types on $\mathcal{P}_s$, which the authors consolidate into six overlapping intent categories $\mathcal{T} = \{\text{attribute, add, remove, preserve, relation, scene}\}$ based on recurrence in the sampled edits and the need for distinct retrieval evidence; scene/global edits concern the background, viewpoint, layout, or spatial context of the overall image. With this schema fixed, GPT-4o assigns one or more intent labels to each pseudo-triplet, defining the intent-specific subsets $\mathcal{P}_s^{(t)} \subseteq \mathcal{P}_s$ for $t \in \mathcal{T}$.

We build a candidate skill pool $\mathcal{K}_0$ aligned with these edit needs and audit each candidate skill $k \in \mathcal{K}_0$ on $\mathcal{P}_s^{(t)}$ by removing it from the pool and computing the per-intent drop in R@10:

$$\Delta R_k^{(t)} \;=\; \mathrm{R@10}\big(\mathcal{K}_0;\, \mathcal{P}_s^{(t)}\big) \;-\; \mathrm{R@10}\big(\mathcal{K}_0 \setminus \{k\};\, \mathcal{P}_s^{(t)}\big).$$

A candidate k is retained only if removing it lowers R@10 on at least one intent slice $t \in \mathcal{T}$, with the largest per-intent drops reported in Appendix B.1. The check is run intent by intent so contributions to specific edit types remain visible alongside the overall score. This MLLM-assisted, author-curated procedure yields a fixed pool $\mathcal{K}$ of **6 skills** with no retriever or planner parameter updates: `TextSkill`, `InstSkill`, `GenSkill`, `RefSkill`, `EntityNeg`, and `SceneSkill`. `TextSkill` scores candidates by CLIP text–image similarity against the target description and is activated for every query, while the other 5 skills add intent-specific evidence conditionally. The same pool $\mathcal{K}$ serves all three benchmarks without modification; per-benchmark variation comes from the planner's per-query subset selection, with overlapping roles supporting mixed queries.

Hypothesis skills ($\sigma_k = +1$)**.** Hypothesis skills attract candidates toward the inferred target via a CLIP cosine similarity to a per-skill query embedding $\mathbf{q}_k(q)$, $s_k(c \mid q) = \cos(\mathbf{q}_k(q), \Psi_I(c))$ for $k \in \{\texttt{TextSkill}, \texttt{InstSkill}, \texttt{GenSkill}, \texttt{RefSkill}\}$. Specifically, $\mathbf{q}_k(q)$ is the encoded planner-produced target description $\Psi_T(T_t)$ (`TextSkill`; always activated), the encoded raw modification

text $\Psi_T(T_m)$ (`InstSkill`), the encoded GPT-image-1 visual hypothesis $\Psi_I(\hat{I})$ (`GenSkill`), or the encoded reference image $\Psi_I(I_r)$ combined with a preservation gloss (`RefSkill`). Together they execute the *target-description*, *instruction-anchor*, *visual-hypothesis*, and *preservation* constraints. Per-skill implementation is in Appendix A.6.

Removal and scene evidence. `EntityNeg` computes the cosine $s_{\texttt{EntityNeg}}(c \mid q) = \cos(\Psi_T(T_{\text{remove}}), \Psi_I(c))$ between the removal phrase T_{remove} and the candidate, which enters composition (§3.4) with a negative outer sign ($\sigma_{\texttt{EntityNeg}} = -1$) to repel candidates containing the removed entity. `SceneSkill` combines scene-level component phrases $\{\phi_j\}$ as $s_{\text{scene}}(c \mid q) = \sum_j \epsilon_j s_j(c \mid q)$, where $s_j(c \mid q) = \cos(\Psi_T(\phi_j), \Psi_I(c))$ is the CLIP text–image cosine for phrase ϕ_j. Must-avoid and negative-relation components have $\epsilon_j = -1$, while required-relation and viewpoint components have $\epsilon_j = +1$. Together they execute the *removal* and *scene/relation* constraints, with signs fixed by component type across queries.

3.3 Planning and Skill Activation

The planner translates the user's edit into a structured plan that the skills can act on. Given a query $q = (I_r, T_m)$ and the skill pool $\mathcal{K}$ from §3.2, a frozen MLLM Ψ_M is prompted with chain-of-thought reasoning over the pool description and produces a plan with the following core fields in a single cached call:

$$\text{plan}(q) \;=\; \big(\mathcal{I}(q),\, \mathcal{A}(q),\, T_t\big) \;=\; \Psi_M(I_r, T_m \mid \mathcal{K}), \tag{2}$$

where $\mathcal{I}(q) \subseteq \mathcal{T}$ is the set of active edit intents, $\mathcal{A}(q) \subseteq \mathcal{K}$ is the activated skill subset, and T_t is the target description. The planner first identifies the active intents, then selects the skills whose evidence channels address each intent. `TextSkill` is in $\mathcal{A}(q)$ for every query because T_t is always produced; the other 5 skills are added conditionally. The skill menu associates attribute and add with `InstSkill` / `GenSkill`, remove with `EntityNeg`, preserve with `RefSkill`, and relation or scene with `SceneSkill`; the MLLM can select multiple skills for mixed queries. For "remove the seashells and make the water green" on a stingray scene, the planner returns $\mathcal{I}(q) = \{\textsf{remove}, \textsf{attribute}, \textsf{preserve}\}$, $\mathcal{A}(q) = \{\texttt{TextSkill}, \texttt{EntityNeg}, \texttt{RefSkill}\}$, and $T_t =$ *"stingray underwater in green water without seashells"*. Each activated skill $k \in \mathcal{A}(q)$ then scores every gallery candidate (§3.4); the intent set $\mathcal{I}(q)$ gates which verification operators run on the top-ranked candidates. The full prompt template, output schema, and an example completion are in Appendix D.

3.4 Score Composition

Per-skill scores must combine without losing per-constraint identity. Before gallery scoring, FashionIQ applies database-side augmentation to gallery embeddings, while CIRR and CIRCO use the original embeddings. Each channel carries a fixed outer sign, with $\sigma_{\texttt{EntityNeg}} = -1$ and $\sigma_k = +1$ for the remaining skills since `SceneSkill` already incorporates its component signs ϵ_j, and the composed score is the signed sum:

$$S(c \mid q) \;=\; \sum_{k \in \mathcal{A}(q)} \sigma_k \cdot s_k(c \mid q), \tag{3}$$

with $\mathcal{A}(q)$ the planner-activated subset. No model parameters are updated; only fixed signs are used. Operating over candidate scores keeps each constraint as an inspectable channel and pairs directly with nonlinear crop verification, while remaining algebraically equivalent to combining the corresponding signed query embeddings for normalized linear cosine channels.

3.5 Verification and Retrieval

A high $S(c \mid q)$ can still rank a candidate near the top while missing a small local change or violating a reference-relative constraint. Two verification operators re-check the top-K candidates after composition, with full equations given in Appendix A.7.

PatchVerify. `PatchVerify` divides each candidate into a 4×4 crop grid $\{c^{(p)}\}_{p=1}^{16}$ and matches planner-extracted must-contain phrases $\Phi^+(q)$ and must-avoid phrases $\Phi^-(q)$ to the most similar crop, rewarding required local details and penalizing forbidden content to execute the *local-detail* constraint.

Table 1: **FashionIQ validation results.** In each backbone block, training-based methods precede the inner rule and training-free methods (grey) follow it. **Bold**/underline marks the best/second-best training-free result per backbone. ViT-G/14 denotes OpenCLIP ViT-bigG/14.

		Shirt		Dress		TopTee		Avg	
Backbone	Method	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50
ViT-B/32	SEARLE [5]	24.44	41.61	18.54	39.51	25.70	46.46	22.89	42.53
	CIReVL [21]	28.36	47.84	25.29	46.36	31.21	53.85	28.29	49.35
	OSrCIR [41]	31.16	51.13	29.35	50.37	36.51	58.71	32.34	53.40
	G-MIXER [26]	37.24	55.99	36.39	58.21	45.23	64.10	39.62	59.43
	WISER [44]	41.51	56.92	37.48	56.97	46.97	62.32	41.99	58.74
	SkillCIR	**47.62**	**63.18**	**42.71**	**62.53**	**50.14**	**67.41**	**46.82**	**64.37**
ViT-L/14	LinCIR [16]	29.10	46.81	20.92	42.44	28.81	50.18	26.28	46.49
	PrediCIR [40]	31.80	52.00	25.40	49.50	33.10	55.40	30.10	52.30
	CIReVL [21]	29.49	47.40	24.79	44.76	31.36	53.65	28.55	48.57
	OSrCIR [41]	33.17	52.03	29.70	51.81	36.92	59.27	33.26	54.37
	G-MIXER [26]	40.87	60.35	37.98	60.93	46.91	66.14	41.92	62.47
	WISER [44]	43.13	57.85	38.42	55.92	45.39	61.50	42.17	58.51
	SkillCIR	**49.83**	**65.91**	**44.27**	**64.83**	**52.16**	**69.27**	**48.75**	**66.67**
ViT-G/14	LinCIR [16]	46.76	65.11	38.08	60.88	50.48	71.09	45.11	65.69
	PrediCIR [40]	48.20	67.40	39.70	62.40	53.70	73.60	47.20	67.80
	CIReVL [21]	33.71	51.42	27.07	49.53	35.80	56.14	32.19	52.36
	OSrCIR [41]	38.65	54.71	33.02	54.78	41.04	61.83	37.57	57.11
	G-MIXER [26]	39.65	59.61	34.71	58.85	44.77	67.47	39.71	61.98
	WISER [44]	45.98	61.58	38.57	58.35	49.21	66.96	44.59	62.30
	SkillCIR	**53.39**	**70.43**	**47.94**	**69.58**	**56.20**	**73.12**	**52.51**	**71.04**

Counterfactual Presence Matching (CPM). CPM extends PatchVerify with reference-relative checks over *add* phrases $\Phi^{\text{add}}(q)$ and *remove* phrases $\Phi^{\text{remove}}(q)$ using patch activation $a_\phi(I) = \max_p \cos(\Psi_T(\phi), \Psi_I(I^{(p)}))$. Because the reference terms $a_\phi(I_r)$ are constant across candidates, they do not change the ordering among the verified top-K candidates, while the candidate-dependent terms reward added content and penalize removed content to verify edit direction.

Final ranking. The final score adds verification to composition on the top-K candidates:

$$\mathcal{R}(c \mid q) \;=\; S(c \mid q) \;+\; \Pi(c \mid q) \;+\; \Gamma(c \mid q), \tag{4}$$

where Π and Γ are the PatchVerify and CPM terms, weighted by fixed dataset-specific coefficients and contributing 0 when the planner does not activate them. Candidates are ranked by $\mathcal{R}(c \mid q)$, followed on CIRR by a fixed InstSkill tiebreaker for near ties; DBA, coefficient values, and tiebreaker details are given in Appendix A.

4 Experiments

Datasets and metrics. We evaluate on three standard CIR benchmarks: FashionIQ [47] for fine-grained fashion attribute edits, CIRR [27] for general-domain composed queries built on NLVR^2, and CIRCO [5] for open-domain composed queries with multiple ground-truths to mitigate false negatives. Following each benchmark's official protocol on FashionIQ validation and CIRR and CIRCO test, we report category-averaged Recall@k (R@k) for FashionIQ, full-gallery R@k and $\text{Recall}_{\text{Subset}}$@$k$ for CIRR, and mean average precision (mAP@k) for CIRCO. Dataset statistics, splits, and evaluation servers are in Appendix A.1.

Baselines. We compare against training-based methods (SEARLE [5], LinCIR [16], PrediCIR [40]) and training-free methods: **CIReVL** [21], captioner + LLM rewriting; **OSrCIR** [41], one-stage reflective CoT; **LDRE** [49], divergent caption ensembles, and **LDRE+IP-CIR** [24], which augments them with generated image proxies; **G-MIXER** [26], geodesic mixup with explicit re-ranking; and

Table 2: **CIRCO and CIRR test results.** In each backbone block, training-based methods precede the inner rule and training-free methods (grey) follow it. **Bold**/underline marks the best/second-best training-free result per backbone.

		CIRCO (test)				CIRR Recall@k			CIRR Recall$_{\text{Subset}}$@k		
Backbone	Method	mAP@5	mAP@10	mAP@25	mAP@50	k=1	k=5	k=10	k=1	k=2	k=3
ViT-B/32	SEARLE [5]	9.35	9.94	11.13	11.84	24.00	53.42	66.82	54.89	76.60	88.19
	CIReVL [21]	14.94	15.42	17.00	17.82	23.94	52.51	66.00	60.17	80.05	90.19
	OSrCIR [41]	18.04	19.17	20.94	21.85	25.42	54.54	68.19	62.31	80.86	91.13
	G-MIXER [26]	21.33	22.00	24.00	24.98	35.18	66.19	76.82	69.98	85.78	92.92
	WISER [44]	32.23	33.18	34.82	35.35	49.45	76.55	85.21	77.30	88.63	92.27
	SkillCIR	**36.45**	**37.20**	**40.15**	**41.30**	**49.78**	**79.50**	**88.05**	**77.58**	**89.05**	**94.50**
ViT-L/14	LinCIR [16]	12.59	13.58	15.00	15.85	25.04	53.25	66.68	57.11	77.37	88.89
	PrediCIR [40]	15.70	17.10	18.60	19.30	27.20	57.00	70.20	—	—	—
	CIReVL [21]	18.57	19.01	20.89	21.80	24.55	52.31	64.92	59.54	79.88	89.69
	OSrCIR [41]	23.87	25.33	27.84	28.97	29.45	57.68	69.86	62.12	81.92	91.10
	G-MIXER [26]	28.29	29.04	31.44	32.39	37.42	67.69	78.58	71.88	87.04	92.82
	WISER [44]	35.10	36.30	38.46	39.15	49.23	76.72	85.11	77.81	88.89	92.77
	SkillCIR	**40.10**	**40.95**	**44.75**	**46.20**	**49.61**	**80.05**	**88.55**	**78.05**	**89.20**	**94.85**
ViT-G/14	LinCIR [16]	19.71	21.01	23.13	24.18	35.25	64.72	76.05	63.35	82.22	91.98
	PrediCIR [40]	23.70	24.60	25.40	26.00	37.00	66.10	77.90	—	—	—
	CIReVL [21]	26.77	27.59	29.96	31.03	34.65	64.29	75.06	67.95	84.87	93.21
	OSrCIR [41]	30.47	31.14	35.03	36.59	37.26	67.25	77.33	69.22	85.28	93.55
	G-MIXER [26]	31.79	32.54	35.49	36.87	39.18	69.83	79.35	72.36	87.25	93.85
	WISER [44]	36.53	38.14	40.46	41.26	**49.54**	77.40	85.76	**78.10**	89.06	92.68
	SkillCIR	**41.80**	**41.83**	**46.72**	**48.06**	49.06	**80.17**	**88.64**	77.42	**89.36**	**95.18**

WISER [44], multi-channel composed retrieval with MLLM verification and refinement. Main-table baseline values are taken from their publications; the refined-query and reranking comparisons use the matched settings described below.

Implementation details. We use GPT-4o [29] as the default MLLM planner with temperature 0, and report ablations with Claude Opus 4.6 [1], GPT-4.1 [30], and the open-source Qwen3-VL [3]. All retrieval runs on a single NVIDIA H100 GPU with CLIP weights from OpenCLIP [20] (ViT-B/32, ViT-L/14, and the default ViT-bigG/14, LAION-2B). For `GenSkill`, we use GPT-image-1 to synthesize visual hypotheses, with FLUX.1 Kontext (`[dev]`) [6] and SDXL-Turbo [34] reported as open-weight generator ablations. A randomly sampled 50K MegaPairs [53] subset is used once offline for skill-pool induction (§3.2); no gradient training is involved. Per-skill signs are fixed across benchmarks; CIRR verification uses top-$K = 10$, and skill scores and verification operators are cached across runs. Full prompts and post-processing details are in Appendix A and Appendix D.

4.1 Main Results

Tables 1 and 2 examine whether routing each inferred constraint to its own scoring channel addresses the intent execution gap in practice across three benchmarks that span the constraint-mix spectrum `SkillCIR` is designed to handle: attribute-heavy FashionIQ, mixed-intent CIRR, and scene-heavy CIRCO. With the default ViT-G/14 backbone, `SkillCIR` reaches 52.51% average R@10 on FashionIQ, improving over WISER by 7.92 points and over CIReVL by 20.32 points. On CIRCO, `SkillCIR` reaches 41.80% mAP@5 and 41.83% mAP@10, improving over WISER by 5.27 and 3.69 points. On CIRR, `SkillCIR` reaches 88.64% R@10 and 80.17% R@5, improving over WISER by 2.88 and 2.77 points. It trails WISER slightly by 0.48 points on CIRR R@1 and 0.68 points on subset R@1, reflecting the top-1 sensitivity of the rerank-free dual-encoder setting. On the smaller B/32 and L/14 backbones the trends are consistent, with FashionIQ average R@10 improving over WISER by 4.83 and 6.58 points and CIRCO mAP@10 by 4.02 and 4.65 points. For the additional ViT-G/14 baselines, LDRE [49] achieves 36.15 CIRR R@1 and 31.12 CIRCO mAP@5, while LDRE+IP-CIR [24] reaches 39.25 and 32.75, respectively; `SkillCIR` exceeds the latter by 9.81 and 9.05 points. In addition, `SkillCIR` outperforms OSrCIR and WISER on Refined-CIRR and Refined-FashionIQ [19], improving over WISER by 5.9 R@1 points on Refined-CIRR and 5.5 average R@10 points on Refined-FashionIQ, as reported in Table 14 of Appendix B.5.

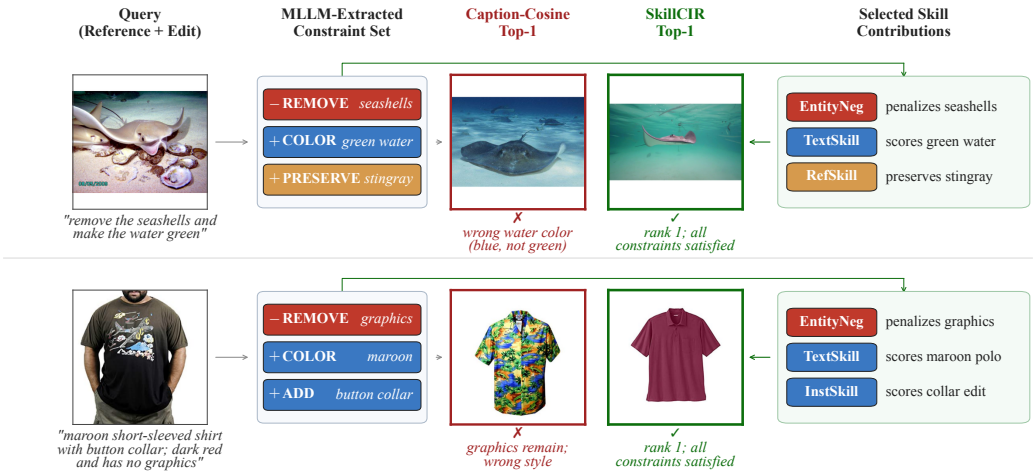


Figure 3: **Qualitative comparison of caption-cosine and `SkillCIR` top-1 results.** Typed skills execute the green-water edit (row 1) and graphic-removal edit (row 2). `TextSkill` is always active; highlighted skills show their contributions. More examples are in Appendix B.6.

Table 3: FashionIQ average R@10 (val, ViT-bigG/14) with and without Qwen3-VL-Reranker-8B.

Candidate generator	8B reranker	Avg R@10
TextSkill-only	No	44.76
TextSkill-only	Yes	50.77
`SkillCIR`	No	52.51
`SkillCIR`	Yes	54.58

Qualitative comparison. Figure 3 illustrates where per-skill execution changes the top-1. In the first row, caption-cosine returns a result that matches the dominant subject but misses both the negation and the green-water edit; `SkillCIR`'s `EntityNeg` penalizes the removed-entity evidence while `TextSkill` and `RefSkill` score the target description and preserve the subject identity, recovering the correct target. In the second row, caption-cosine matches the dominant attributes but retains graphics despite the *no-graphics* instruction; `SkillCIR`'s `EntityNeg` penalizes candidates that violate the negation while positive channels score the requested appearance and collar detail. Extended examples spanning attribute, removal, type-change, add, and replace edits are in Appendix B.6.

Reranking comparison. Table 3 applies Qwen3-VL-Reranker-8B [23] to `TextSkill`-only and `SkillCIR` with the same candidate-pool size, prompt, and inference settings. Reranking improves the two paths by 6.01 and 2.07 average R@10 points, respectively; with the reranker fixed, the `SkillCIR` path remains 3.81 points higher. The optional reranker adds a local 8B model with a measured peak GPU-memory footprint of 17.5 GB and is excluded from the base latency.

4.2 Ablation Study

If `SkillCIR` effectively narrows the intent execution gap, each component of its execution stack should contribute, and Table 4 ablates the stack in four groups. **(1) Module ablations.** Removing any ablated skill or verification operator reduces CIRR R@1 by 1.23–1.88 points in line with each benchmark's intent mix in Figure 4: removing `SceneSkill` lowers CIRR R@1 most by 1.88 points, whereas removing `PatchVerify` lowers FashionIQ R@10 most by 1.55 points. Overall, the full system improves over the matched `TextSkill`-only control by 3.04 CIRR R@1 points from 46.02 to 49.06, while `RefSkill` is evaluated in the MegaPairs leave-one-out audit in Figure 4(b). **(2) Signed composition.** Replacing the signed sum with a uniform positive sum drops CIRR R@1 by 2.63 points and FashionIQ average R@10 by 2.45 points, indicating that the negative sign on constraint skills repels violating candidates. **(3) Planner alternatives.** Activating all skills reduces CIRR R@1 by 1.26 points, random 50% masking reduces it by 2.86 points, and a deterministic keyword parser retaining the MLLM target description trails by 0.66 CIRR R@1 and 0.36 FashionIQ average R@10, showing

Table 4: Ablation on CIRR (test) and FashionIQ (val), ViT-bigG/14, with GPT-4o as the default planner. Routing alternatives retain the MLLM-generated target description.

	CIRR			FashionIQ	
Configuration	R@1	R@5	R@10	R@10	R@50
Full `SkillCIR`	49.06	80.17	88.64	52.51	71.04
Module ablations					
Without GenSkill	47.43	78.21	87.18	51.27	69.91
Without InstSkill	47.69	78.49	87.31	51.13	69.62
Without EntityNeg	47.51	78.32	87.22	51.45	69.93
Without SceneSkill	47.18	78.05	86.93	52.08	70.61
Without PatchVerify	47.83	78.71	87.49	50.96	69.71
Without CPM	47.71	78.58	87.39	51.42	70.13
All-positive composition	46.43	77.18	86.31	50.06	69.43
Routing alternatives					
All skills active	47.80	78.84	87.62	51.74	70.18
Random 50% skill mask	46.20	76.95	86.13	49.83	69.21
Keyword skill selection	48.40	79.27	88.06	52.15	70.71
GenSkill image generator					
FLUX.1 Kontext [6]	48.85	79.94	88.41	52.34	70.86
SDXL-Turbo [34]	47.63	78.71	87.18	51.42	69.87
Planner MLLM substitutions					
Claude Opus 4.6 [1]	49.23	80.36	88.79	52.68	71.21
GPT-4.1 [30]	49.17	80.28	88.72	52.62	71.13
Qwen3-VL [3]	48.45	79.56	87.91	51.83	70.31

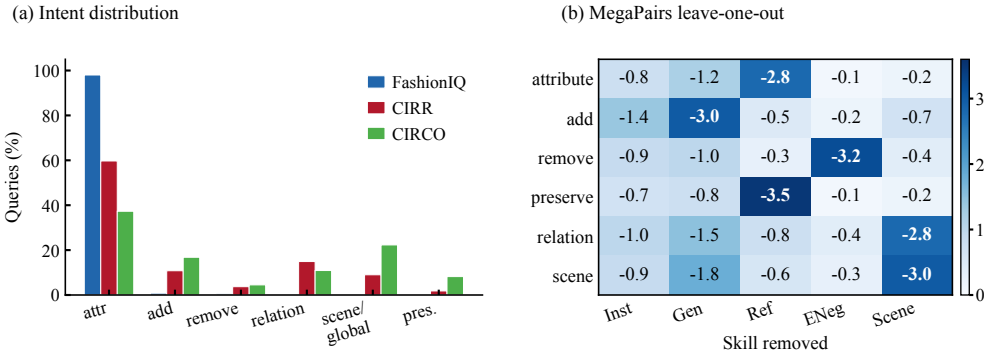


Figure 4: **Intent distribution and skill specialization.** **(a)** Per-benchmark intent distribution; *scene/global* covers background, viewpoint, layout, or spatial context. **(b)** MegaPairs per-intent ΔR@10 after removing each conditional skill. Bold marks the largest decrease per row; `TextSkill` remains active.

that routing contributes beyond skill availability. **(4) Upstream engine substitutions.** Replacing GPT-image-1 with FLUX.1 Kontext or SDXL-Turbo changes CIRR R@1 by -0.21 or -1.43 points, and substituting the planner across MLLMs keeps CIRR R@1 within 48.45–49.23 and FashionIQ average R@10 within 51.83–52.68, as detailed in Appendix B.3. Together, typed skills, signed composition, and query-specific MLLM routing all contribute to closing the gap.

4.3 Planner and Verification Analysis

Figures 4 and 5 diagnose intent specialization, per-intent CIRR recall, and 170+ post-hoc design runs.

Skill induction transfers. Figure 4(b) shows conditional-skill specialization on the external 50K MegaPairs pool, matching the CIRR removal pattern in Table 4: for every intent except *attribute*, where removing `RefSkill` hurts most at -2.8 R@10, the largest drop aligns with the dedicated channel for that constraint, namely -3.2 from `EntityNeg` on *remove*, -3.5 from `RefSkill` on *preserve*, and -3.0 from `SceneSkill` on *scene/global*, with full results including `TextSkill` in

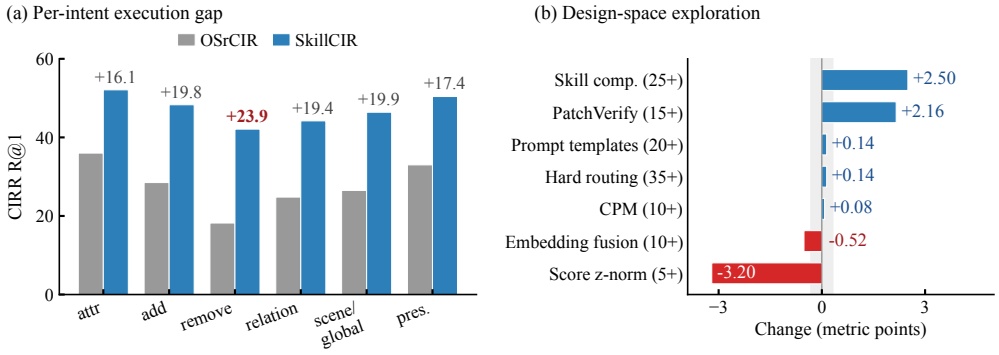


Figure 5: **Per-intent performance and design analysis. (a)** CIRR R@1 and gains over OSrCIR, using the intent groups in Figure 4. **(b)** Post-hoc design comparisons after freezing the configuration on MegaPairs (run counts in parentheses): `PatchVerify` uses CIRCO mAP@5, `CPM` CIRR R@10, normalization FashionIQ average R@10, and the other four categories CIRR R@1.

Appendix B.1. The fixed pool transfers across FashionIQ, CIRR, and CIRCO via per-query subset selection.

Per-intent comparison. Figure 5(a) compares per-intent R@1 against OSrCIR [41] on CIRR val, where per-query labels are available: the gain widens from $+16.1$ on *attribute* to $+23.9$ on *remove* and $+19.4$ on *relation*, where execution requires explicit suppression or relational evidence. Alongside the routing controls in §4.2, this indicates that both dedicated channels and selective activation contribute.

Design-space exploration. Figure 5(b) summarizes 170+ post-hoc design runs detailed in Appendix C: skill-pool composition adds $+2.50$ CIRR R@1 and `PatchVerify` adds $+2.16$ CIRCO mAP@5, dominating their respective comparisons, whereas prompt templates and hard routing saturate near $+0.14$. Per-benchmark variation is planner-driven: hypothesis skills on FashionIQ, `EntityNeg` and `PatchVerify` on CIRR removal and relation queries, and `SceneSkill` and `PatchVerify` on CIRCO. Bootstrap intervals and the GPT-image-2 study are in Appendix B.

Planner diagnostics. Including always-active `TextSkill`, mean active-skill counts are 2.61 on FashionIQ and 3.07 on CIRR and CIRCO, as reported in Appendix B.4. `EntityNeg` activates on only 4.8% of non-removal queries (138/2,886), while raising CIRR val removal R@1 from 39.86 when inactive to 48.60 when active in Appendix B.2; rule-based removal-phrase extraction reaches $\sim$85% precision relative to the MLLM planner in Appendix D. Across 15+ configurations, soft intent-dependent weighting improved pre-verification CIRR R@1 by up to 1.60 points without further post-verification gain.

Runtime. Over a 100-query CIRR trace with cached gallery embeddings and visual hypotheses, `SkillCIR` averages 3.0 s/query, comprising 1.8 s for planning, 0.7 s for conditional top-10 verification, and 0.05 s for score composition. Against default WISER at 70.6 s/query on CIRCO, `SkillCIR` achieves higher CIRCO mAP@5 (41.80 vs. 36.53) at a $23.5\times$ lower latency, and spans a $3.8\times$–$23.5\times$ speedup across WISER operating points in Table 11.

5 Conclusion

In this paper, we introduce `SkillCIR`, a training-free framework for composed image retrieval that addresses the *intent execution gap* by routing inferred edit constraints to a fixed pool of typed retrieval skills and combining their scores with explicit attract/repel signs at ranking time. Across three benchmarks and three CLIP backbones, `SkillCIR` improves the primary metrics by 2.8 to 7.9 points over the prior training-free state of the art without model-parameter updates; selective skill routing outperforms all-on and random activation, replacing the GPT-4o planner across MLLMs varies CIRR R@1 within 0.61 points of the default without retraining, and pairing `SkillCIR` with a multimodal reranker adds a further 2.07 average R@10 gain on FashionIQ.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62572465). The authors declare no additional funding or competing financial interests beyond the support and affiliations disclosed in this paper.

References

[1] Anthropic. Introducing Claude Opus 4.6. Anthropic, February 2026. URL https://www.anthropic.com/news/claude-opus-4-6.

[2] Anthropic. Introducing Claude Opus 5. Anthropic, July 2026. URL https://www.anthropic.com/news/claude-opus-5.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

[4] Yang Bai, Xinxing Xu, Yong Liu, Salman Khan, Fahad Khan, Wangmeng Zuo, Rick Siow Mong Goh, and Chun-Mei Feng. Sentence-level prompts benefit composed image retrieval. In *ICLR*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8204f54f6cd8f0db1765a274e94ceaac-Paper-Conference.pdf.

[5] Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Alberto Del Bimbo. Zero-shot composed image retrieval with textual inversion. In *ICCV*, pages 15338–15347, 2023.

[6] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[7] Jaeseok Byun, Young Kyun Jang, Seokhyeon Jeong, Donghyun Kim, and Taesup Moon. MA-CIR: A multimodal arithmetic benchmark for composed image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21342–21352, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Byun_MA-CIR_A_Multimodal_Arithmetic_Benchmark_for_Composed_Image_Retrieval_ICCV_2025_paper.html.

[8] Zhangtao Cheng, Yuhao Ma, Jian Lang, Kunpeng Zhang, Ting Zhong, Yong Wang, and Fan Zhou. Generative thinking, corrective action: User-friendly composed image retrieval via automatic multi-agent collaboration. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pages 334–344, 2025. doi: 10.1145/3711896.3736982. URL https://doi.org/10.1145/3711896.3736982.

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

[10] Guofeng Ding, Yiding Lu, Peng Hu, Mouxing Yang, Yijie Lin, and Xi Peng. Visual abstraction: A plug-and-play approach for text-visual retrieval. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 13825–13844. PMLR, 2025. URL https://proceedings.mlr.press/v267/ding25b.html.

[11] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23 (120):1–39, 2022. URL http://jmlr.org/papers/v23/21-0998.html.

[12] Yue Feng, Ziyi Yan, Yizhen Jia, Ethan Q Chen, and Jie Qin. Training-free dense video captioning with large-scale pretrained models. *Machine Intelligence Research*, pages 1–16, 2026.

[13] Tianhong Gao, Yannian Fu, Weiqun Wu, Haixiao Yue, Shanshan Liu, and Gang Zhang. MMAT-1M: A large reasoning dataset for multimodal agent tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1484–1494, 2025.

[14] Xuri Ge, Chunhao Wang, Xindi Wang, Zheyun Qin, Zhumin Chen, and Xin Xin. MCoT-MVS: Multi-level vision selection by multi-modal chain-of-thought reasoning for composed image retrieval. In *Proceedings of the ACM Web Conference 2026*, pages 2105–2113, 2026. doi: 10.1145/3774904.3792287. URL https://arxiv.org/abs/2603.17360.

[15] 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*, 2024. URL https://openreview.net/forum?id=mKtlzW0bWc.

[16] Geonmo Gu, Sanghyuk Chun, Wonjae Kim, Yoohoon Kang, and Sangdoo Yun. Language-only training of zero-shot composed image retrieval. In *CVPR*, 2024.

[17] Bohan Hou, Haoqiang Lin, Haokun Wen, Meng Liu, Mingzhu Xu, and Xuemeng Song. Pseudo triplet guided few-shot composed image retrieval. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2025.

[18] Ziniu Hu, Ahmet Iscen, Chen Sun, Kai-Wei Chang, Yizhou Sun, David A. Ross, Cordelia Schmid, and Alireza Fathi. AVIS: Autonomous visual information seeking with large language model agent. In *NeurIPS*, pages 867–878, 2023.

[19] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4004, 2025.

[20] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP. Zenodo, 2021. URL https://doi.org/10.5281/zenodo.5143773.

[21] Shyamgopal Karthik, Karsten Roth, Massimiliano Mancini, and Zeynep Akata. Vision-by-language for training-free compositional image retrieval. In *ICLR*, 2024. URL https://openreview.net/forum?id=EDPxCjXzSb.

[22] Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

[23] Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. arXiv:2601.04720, 2026. URL https://arxiv.org/abs/2601.04720.

[24] 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 (CVPR)*, pages 3984–3993, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Li_Imagine_and_Seek_Improving_Composed_Image_Retrieval_with_an_Imagined_CVPR_2025_paper.html.

[25] Zixu Li, Zhiwei Chen, Haokun Wen, Zhiheng Fu, Yupeng Hu, and Weili Guan. ENCODER: Entity mining and modification relation binding for composed image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5101–5109, 2025. doi: 10.1609/aaai.v39i5.32541. URL https://ojs.aaai.org/index.php/AAAI/article/view/32541.

[26] Jiyoung Lim, Heejae Yang, and Jee-Hyong Lee. G-MIXER: Geodesic mixup-based implicit semantic expansion and explicit semantic re-ranking for zero-shot composed image retrieval. In *CVPR*, pages 33891–33900, 2026. URL https://arxiv.org/abs/2604.14710.

[27] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *ICCV*, 2021.

[28] Yanze Min, Yawei Sun, Yin Zhu, Jun Zhu, and Bo Zhang. Bootstrapping large language models with outsideknowledge for knowledge-based visual question answering. *Machine Intelligence Research*, 23(1):115–132, 2026.

[29] OpenAI. Hello GPT-4o. OpenAI, May 2024. URL `https://openai.com/index/hello-gpt-4o/`.

[30] OpenAI. Introducing GPT-4.1 in the API. OpenAI, April 2025. URL `https://openai.com/index/gpt-4-1/`.

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

[32] 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 *CVPR*, 2023.

[33] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, 2022.

[34] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision (ECCV)*, pages 87–103. Springer, 2024.

[35] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36: 68539–68551, 2023.

[36] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI tasks with ChatGPT and its friends in Hugging Face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

[37] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, 2019.

[38] Zelong Sun, Dong Jing, and Zhiwu Lu. CoTMR: Chain-of-thought multi-scale reasoning for training-free zero-shot composed image retrieval. In *ICCV*, pages 22675–22684. IEEE, 2025. URL `https://arxiv.org/abs/2502.20826`.

[39] 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 *AAAI*, volume 38, pages 5180–5188, 2024.

[40] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24785–24795, 2025. URL `https://arxiv.org/abs/2503.17109`.

[41] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14400–14410, 2025.

[42] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Fei-Fei Li, and James Hays. Composing text and image for image retrieval: An empirical odyssey. In *CVPR*, pages 6439–6448, 2019.

[43] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.

[44] 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 (CVPR)*, pages 16865–16875, 2026. URL https://arxiv.org/abs/2602.23029.

[45] Tong Wang, Yunhan Zhao, and Shu Kong. Generating a paracosm for training-free zero-shot composed image retrieval. In *ECCV*, 2026. URL https://arxiv.org/abs/2602.00813.

[46] Haokun Wen, Xuemeng Song, Haoyu Zhang, Weili Guan, Xiangyu Zhao, and Liqiang Nie. UniCVR: From alignment to reranking for unified zero-shot composed visual retrieval. *arXiv preprint arXiv:2604.20318*, 2026. URL https://arxiv.org/abs/2604.20318.

[47] 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 *CVPR*, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Wu_Fashion_IQ_A_New_Dataset_Towards_Retrieving_Images_by_Natural_CVPR_2021_paper.html.

[48] Xingyu Yang, Daqing Liu, Heng Zhang, Yong Luo, Chaoyue Wang, and Jing Zhang. Decomposing semantic shifts for composed image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6576–6584, 2024. doi: 10.1609/aaai.v38i7.28479. URL https://ojs.aaai.org/index.php/AAAI/article/view/28479.

[49] Zhenyu Yang, Dizhan Xue, Shengsheng Qian, Weiming Dong, and Changsheng Xu. LDRE: LLM-based divergent reasoning and ensemble for zero-shot composed image retrieval. In *SIGIR*, pages 80–90, 2024.

[50] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, 2023.

[51] Zhixin Zhang, Yiyuan Zhang, Xiaohan Ding, and Xiangyu Yue. Vision search assistant: Empower vision-language models as multimodal search engines. *arXiv preprint arXiv:2410.21220*, 2024.

[52] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. GPT-4V(ision) is a generalist web agent, if grounded. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pages 61349–61385, 2024. URL https://proceedings.mlr.press/v235/zheng24e.html.

[53] Junjie Zhou, Yongping Xiong, Zheng Liu, Ze Liu, Shitao Xiao, Yueze Wang, Bo Zhao, Chen Jason Zhang, and Defu Lian. MegaPairs: Massive data synthesis for universal multimodal retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19076–19095, 2025. URL https://aclanthology.org/2025.acl-long.935/.

A Implementation Details

This appendix describes the dataset protocols, scoring operators, and implementation settings. The evaluation configuration is `configs/final_skillcir.yaml`.

A.1 Datasets

FashionIQ [47]. Fine-grained fashion attribute editing with three category subsets: Shirt (2038 query / 6346 gallery), Dress (2017 / 3817), and Toptee (1961 / 5373). Evaluated on the val split with Recall@{10, 50}.

CIRR [27]. General-domain composed retrieval with 4,148 test queries and 2,315 gallery images drawn from NLVR^2 [37]. Evaluated on the test split via the official server at `https://cirr.cecs.anu.edu.au/`; metrics: Recall@{1, 5, 10} and $\text{Recall}_{\text{Subset}}$@{1, 2, 3}.

CIRCO [5]. Open-domain composed retrieval with 800 test queries and 123K+ gallery images derived from MS-COCO. Multiple ground-truths per query mitigate false negatives. Evaluated on the test split via the official server at `https://circo.micc.unifi.it/`; metric: mAP@{5, 10, 25, 50}.

A.2 Models and Hardware

CLIP backbone. The default retriever is frozen OpenCLIP ViT-bigG-14 (LAION-2B, `laion2b_s39b_b160k` checkpoint). It uses 224×224 images and 1280-dimensional embeddings. The backbone for each reported comparison is identified in the result tables.

MLLM engines. GPT-4o [29] is the default planner, with decoding temperature set to 0. Planner substitutions use Claude Opus 4.6 [1], GPT-4.1 [30], and Qwen3-VL [3]. The planner produces the target description used by TextSkill. API responses are cached for reuse across retrieval runs.

Text-to-image generation. GenSkill uses GPT-image-1 via API. Generated images are cached to disk (one image per query), eliminating redundant generation across runs.

Hardware and timing. The inference hardware is specified in §4. Gallery scoring and post-processing use a single GPU. The cache-enabled latency reuses gallery embeddings and generated visual hypotheses. A first-time GenSkill image-generation call is excluded from that latency and is required when the selected hypothesis is not cached.

A.3 Post-Processing Configurations

The dataset-specific post-processing settings are selected on the 50K MegaPairs development pool (§3.2) and held fixed for evaluation; no benchmark split is used for selection. FashionIQ uses DBA, CIRR uses a top-1 tiebreaker, and on CIRR PatchVerify and CPM operate on the top 10 candidates.

The verification contributions in Eq. 4 are $\Pi(c \mid q) = \alpha_P s_{\text{local}}(c \mid q)$ and $\Gamma(c \mid q) = \alpha_C s_{\text{cpm}}(c \mid q)$, with fixed dataset-specific coefficients α_P and α_C, using the operator scores in Eqs. 5 and 6. These dataset-specific post-processing settings are separate from the fixed signs in skill composition.

DBA details. Database-side Augmentation smooths the gallery manifold by averaging each L2-normalized gallery embedding $\Psi_I(c)$ with its N nearest neighbors: $\tilde{\Psi}_I(c) = \text{L2-normalize}\Big(\Psi_I(c) + \beta \cdot \frac{1}{N} \sum_{j \in \text{NN}_N(c)} \Psi_I(j)\Big)$. Parameters: $N = 10$, $\beta = 0.3$. Applied to FashionIQ only, where gallery items cluster tightly by category in CLIP space. In post-hoc measurements with the frozen configuration, DBA improves FashionIQ average R@10 by $+1.85$ points across subsets ($+1.75$ to $+1.94$), and is neutral to slightly negative on CIRR and CIRCO, whose galleries span diverse visual domains.

Tiebreaker. For CIRR, when the margin between the top-1 and top-2 candidates is below $\tau = 0.05$, InstSkill provides a disambiguation boost of 0.3. This resolves ambiguous top-1 rankings where the composite score does not strongly prefer either candidate.

A.4 Skill Activation

The planner activates skills conditionally per query (§3.3). Appendix B.4 reports per-skill activation rates for each benchmark.

A.5 Prompt Templates

The target-description scaffold asks the MLLM to observe reference attributes, analyze the instruction, and describe the target. The full planner extends this scaffold with edit decomposition and skill selection, producing the fields shown in Appendix D.1. A caption-only output by itself does not specify a skill route.

A.6 Full Skill Pool with Implementation Details

Table 5 reports the complete Skill Pool with implementation details (purpose, evidence channel, encoder).

Table 5: Full `SkillCIR` Skill Pool with implementation details. Skill families and roles are summarized inline in §3.2.

Skill	Purpose / why it is needed	Implementation
Hypothesis family ($\sigma_k = +1$, attract candidates toward the target)		
TextSkill	Describe the edited target. Anchors retrieval to the post-edit semantics that caption-based pipelines rely on.	Planner-produced target description (§3.3); CLIP text–image cosine.
InstSkill	Anchor the raw modification text. Preserves the user's literal phrasing when MLLM rewriting drifts away from the original intent.	Modification text encoded directly; CLIP text–image cosine.
GenSkill	Provide a visual hypothesis. Recovers details that text under-specifies for complex or compositional edits.	GPT-image-1 generated target hypothesis; CLIP image–image cosine.
RefSkill	Preserve unchanged reference content. Counters the tendency of caption-based retrieval to drift away from the reference identity.	Reference image combined with a preservation gloss; CLIP cosine.
Removal and scene constraints		
EntityNeg	Suppress entities marked for removal. Without negative evidence, removed entities frequently still appear in top-ranked candidates.	Negative phrase; CLIP text–image cosine, sign-inverted in Eq. 3.
SceneSkill	Score individual scene and relation requirements.	Fixed component signs: positive for required relations/viewpoints, negative for must-avoid content/negative relations.

A.7 Verification Operator Equations

`PatchVerify` decomposes the local score over the 4×4 crop grid $\{c^{(p)}\}_{p=1}^{16}$ into positive (must-contain phrases $\Phi^+(q)$) and negative (must-avoid phrases $\Phi^-(q)$) channels:

$$s_{\text{local}}(c \mid q) = \frac{1}{|\Phi^+(q)|} \sum_{\phi \in \Phi^+(q)} \max_p \cos(\Psi_T(\phi), \Psi_I(c^{(p)})) - \lambda_{\text{neg}} \max_{\phi \in \Phi^-(q)} \max_p \cos(\Psi_T(\phi), \Psi_I(c^{(p)})). \tag{5}$$

Here, Ψ_I and Ψ_T denote the CLIP image and text encoders with cosine similarity $\cos(\cdot, \cdot)$. The positive term averages the strongest crop match for each required phrase, while the negative term penalizes the strongest match to a forbidden phrase. These are soft similarity scores and do not guarantee the presence or absence of an attribute.

CPM detail.

$$s_{\text{cpm}}(c \mid q) = \sum_{\phi \in \Phi^{\text{add}}(q)} \big[a_\phi(c) - a_\phi(I_r)\big] + \sum_{\phi \in \Phi^{\text{remove}}(q)} \big[a_\phi(I_r) - a_\phi(c)\big], \tag{6}$$

where $a_\phi(I) = \max_p \cos\big(\Psi_T(\phi), \Psi_I(I^{(p)})\big)$ is the strongest patch activation of phrase ϕ in image I, and $\Phi^{\text{add}}(q)$, $\Phi^{\text{remove}}(q)$ are the *add* and *remove* phrase sets from the cached edit parse. For a fixed query, the reference-image terms $a_\phi(I_r)$ are constant over candidates and therefore leave the ordering induced by this score unchanged among the verified top-K candidates. The candidate terms favor stronger add-phrase activation and weaker remove-phrase activation, without enforcing a hard candidate–reference inequality.

B Extended Results

This appendix provides the skill-induction audit, uncertainty estimates, planner diagnostics, engine substitutions, and efficiency measurements. It also describes the additional comparison protocols and presents further retrieval examples.

B.1 Component Ablations and Skill-Induction Audit

Component ablations. Table 4 compares component removal, routing, and signed composition under the main evaluation protocol. Each removal is evaluated within the full system.

Intent-stratified skill induction. Table 6 summarizes the largest R@10 decrease for each intent group in the 50K MegaPairs development pool. The decreases use full-pool performance minus performance after skill removal, following the retention criterion in §3.2. Figure 4(b) shows the conditional-skill matrix with TextSkill active, while this table also considers TextSkill removal. TextSkill contributes most strongly to attribute edits, GenSkill to additions, EntityNeg to removals, RefSkill to preservation, and SceneSkill to relation and scene edits.

Table 6: Largest R@10 decrease after skill removal within each intent group of the 50K MegaPairs pool. Decreases use full minus removed performance.

Intent	Removed skill	R@10 decrease
Attribute	TextSkill	3.5
Add	GenSkill	3.0
Remove	EntityNeg	3.2
Preserve	RefSkill	3.5
Relation	SceneSkill	2.8
Scene/global	SceneSkill	3.0

B.2 Uncertainty and Planner Diagnostics

Bootstrap intervals. Table 7 gives 95% bootstrap confidence intervals for FashionIQ category R@10, using 1000 resamples, the percentile method, and seed 42. The intervals quantify evaluation-set variability for `SkillCIR`. Differences against the matching ViT-G/14 WISER results are descriptive comparisons with published point estimates; the FashionIQ category average has no interval.

Table 7: Single-system bootstrap 95% intervals for `SkillCIR` on FashionIQ. Differences use matching G/14 WISER results and are descriptive. A dash denotes an unavailable interval.

Dataset	Metric	`SkillCIR`	95% CI	WISER	Difference
FIQ shirt	R@10	53.39	[51.13, 55.40]	45.98	+7.41
FIQ dress	R@10	47.94	[45.81, 50.17]	38.57	+9.37
FIQ toptee	R@10	56.20	[54.05, 58.29]	49.21	+6.99
FIQ average	R@10	52.51	—	44.59	+7.92

Failure categories. Visual distractors are a recurring failure mode: a candidate can resemble the reference while missing an edited attribute, removal requirement, or relation. The qualitative examples in Figures 3 and 6 show how separate evidence channels distinguish these partial matches. Query construction, skill selection, and verification can each affect the final ranking.

Table 8: EntityNeg diagnostic on CIRR. The activation denominator is the independently annotated non-removal subset. The two removal-group recall values compare different query subsets.

Diagnostic	Result
Active among queries annotated as not involving removal	138/2,886 (4.8%)
R@1 on removal-type queries, EntityNeg active	48.60
R@1 on removal-type queries, EntityNeg inactive	39.86

EntityNeg activation. We examine EntityNeg activation using a fixed six-intent rubric. Claude Opus 5 [2] provides the primary labels and Claude Fable 5 [2] independently labels the same queries, with mean intent-set Jaccard agreement of 0.89. Four authors inspect a stratified subset and representative disagreements. This produces model-assisted labels with author quality control for the diagnostic in Table 8, which uses CIRR val because it requires per-query labels.

The 4.8% rate measures activation on queries labeled as non-removal. On removal queries, the active and inactive groups differ by 8.74 R@1 points. These groups contain different queries, so the difference is observational. The controlled all-skills and random-mask ablations in Table 4 complement this diagnostic by directly changing the activation policy.

B.3 Planner and Generator Substitutions

Table 4 reports the planner substitutions. The generator substitutions in Table 4 retain the same retrieval backbone.

GenSkill generators. Table 4 replaces GPT-image-1 with FLUX.1 Kontext [6] or SDXL-Turbo [34]. The CIRR R@1 differences are -0.21 and -1.43 points, respectively.

GPT-image-2 substitution. On a 200-query CIRR val pilot, replacing GPT-image-1 with GPT-image-2-0421-global improves gen-only ensemble R@10 by 2.50 points. In the full pipeline, the replacement lowers CIRR R@1 by 0.79 and CIRCO mAP@5 by 1.15 (Table 9). The two settings show that a gain in visual-hypothesis retrieval need not carry through composition and verification. The main configuration uses GPT-image-1.

Table 9: Full-pipeline generator comparison. Differences are GPT-image-2 minus GPT-image-1.

Metric	GPT-image-1	GPT-image-2	Difference
CIRR R@1	49.06	48.27	-0.79
CIRR R@5	80.17	79.02	-1.15
CIRR R@10	88.64	87.99	-0.65
CIRR subset R@1	77.42	76.11	-1.31
CIRCO mAP@5	41.80	40.65	-1.15

B.4 Efficiency and Skill Activation

Cache-enabled latency. Table 10 decomposes the 3.0 s/query CIRR latency averaged over a 100-query trace. The timing includes the planner call, query-side encoding, score composition, and conditional top-10 verification. Gallery embeddings and generated hypotheses are cached; first-time GenSkill image generation adds to this cost and is reported separately here: an uncached query requires one additional GPT-image-1 call only when GenSkill is active, which occurs on 18%, 62%, and 74% of FashionIQ, CIRR, and CIRCO queries (Table 13). Verification is active on approximately 60% of the profiled CIRR queries, and its latency is averaged over the full trace.

Table 10: Per-query latency for the cache-enabled SkillCIR base path. Cold GenSkill image generation is excluded.

Component	Seconds	Scope
MLLM planner	1.80	One target-description and skill-plan call
CLIP encoding	0.40	Active query channels, with cached gallery features
Signed composition	0.05	Full-gallery score fusion
Conditional verification	0.70	Top-K PatchVerify/CPM, amortized over queries
Other	0.05	I/O and post-processing
Total	3.00	Cache-enabled base path

WISER operating points. The additional profiling uses the same single-GPU hardware and batch-size setting across the compared systems, with retrieval quality evaluated under the ViT-bigG/14 CIRCO protocol. Table 11 separates editing, candidate verification, and conditional refinement. The latency-oriented setting uses SDXL-Turbo and $K = 10$ candidates per branch. The balanced setting uses BAGEL with 10 diffusion steps and $K = 50$ per branch. The default accuracy-oriented setting uses default BAGEL with $K = 50$ per branch. Refinement costs are amortized over all profiled queries.

Table 11: WISER operating points: CIRCO accuracy and profiled latency in seconds/query. Ratios use `SkillCIR`'s cache-enabled 3.0 s CIRR reference, excluding cold GenSkill generation.

WISER setting	Editing	Verify	Refine	Total	mAP@5	Ratio
Latency-oriented	0.2	8.3	2.9	11.4	29.47	3.8×
Balanced	7.1	24.6	9.4	41.1	32.91	13.7×
Default	35.7	21.3	13.6	70.6	36.53	23.5×

The balanced and latency-oriented WISER settings reduce total latency while lowering mAP@5 by 3.62 and 7.06 points relative to its default. The balanced configuration reduces BAGEL diffusion steps at the same $K = 50$, whereas the latency-oriented configuration changes both the generator and the per-branch depth to $K = 10$. The default comparison is 41.80 versus 36.53 mAP@5 at 3.0 versus 70.6 s/query under the stated cache conditions. The latency ratios range from 3.8 to 23.5 depending on the WISER setting.

The WISER decomposition further locates the runtime costs. Its default configuration spends 35.7 seconds on editing, 21.3 on candidate verification, and 13.6 on amortized refinement. Verification and refinement together account for 34.9 seconds. In the fastest configuration, those two stages still account for 11.2 of the total 11.4 seconds. Candidate-wise verification and refinement therefore remain substantial costs even with faster image generation.

Cross-method context. Table 12 summarizes the broader configuration-level comparison. The timing values are approximate; the `SkillCIR` entry uses the cache-enabled CIRR trace above, while the accuracy column reports CIRCO mAP@5.

Table 12: Additional accuracy–latency context. Latencies are approximate seconds/query and are configuration-specific. The `SkillCIR` timing excludes first-time GenSkill generation.

Method	Latency (s/query)	CIRCO mAP@5
LinCIR	0.01	19.71
CIReVL	1	26.77
OSrCIR	0.8	30.47
G-MIXER	0.94	31.79
WISER	70	36.53
`SkillCIR`	3	41.80

Activation rates. Table 13 reports the fraction of queries activating each conditional skill. Including the always-active TextSkill gives mean active-skill counts of 2.61 on FashionIQ and 3.07 on both CIRR and CIRCO. CIRR and CIRCO have different skill mixtures despite their equal mean count. FashionIQ activates RefSkill most frequently, and CIRCO activates SceneSkill more often than the other datasets. Activation also follows the query intents. On CIRR, EntityNeg activates on 45% of removal-type queries (including explicit removal and replacement instructions) versus 2–4% of attribute-, add-, and relation-tagged queries; because some substitutions can be resolved through positive target evidence, this is an activation statistic rather than a gate-recall estimate. On CIRCO, removal-tagged queries activate the must-avoid and negative-relation SceneSkill components on 89% and 98% of queries, versus 14% and 31% of attribute-tagged queries.

Table 13: Conditional-skill activation rates. The mean count includes TextSkill, which is active on every query.

Dataset	Inst	Gen	Ref	ENeg	Scene	Mean count
FashionIQ	0.46	0.18	0.92	0.04	0.01	2.61
CIRR	0.71	0.62	0.34	0.18	0.22	3.07
CIRCO	0.55	0.74	0.21	0.16	0.41	3.07

Within the model-labeled intent groups, EntityNeg activates on 45% of CIRR removal-type queries and on 2–4% of attribute-, addition-, or relation-tagged queries. On CIRCO, must-avoid and negative-relation channels activate on 89% and 98% of removal-tagged queries, compared with 14% and 31% of attribute-tagged queries. The tags serve as behavioral groupings. These activation rates do not measure gate recall, since a successful replacement may not require an explicit negative channel.

B.5 Additional Comparisons

Refined benchmarks. The Refined-CIRR and Refined-FashionIQ benchmarks are introduced by CoLLM [19]. Our comparisons use the same frozen OpenCLIP ViT-bigG/14 checkpoint for OSrCIR, WISER, and `SkillCIR`, without optimizing model parameters on annotated CIR triplets. Refined-CIRR uses its complete validation split and hard-target Recall. Refined-FashionIQ reports category-averaged R@10 and R@50. We process the refined modification texts afresh, with separate caches from the original-query evaluations.

Table 14 reports the full refined-benchmark comparison.

Table 14: Refined-benchmark comparisons with a matched training-free ViT-bigG/14 setup. Refined-CIRR uses its full validation split and hard-target recall. Refined-FashionIQ reports category-averaged recall. Modification texts are processed afresh.

	Refined-CIRR			Refined-FashionIQ	
Method	R@1	R@10	R@50	R@10	R@50
OSrCIR	52.3	88.5	97.0	55.2	74.5
WISER	61.4	92.9	98.1	60.5	78.0
`SkillCIR`	67.3	94.8	98.7	66.0	83.2

Matched TextSkill and reranking controls. TextSkill-only retains the corresponding query representation and retrieval protocol while disabling conditional skills and verification. On FashionIQ, both candidate paths use Qwen3-VL-Reranker-8B [23] with identical candidate-pool size, prompt, and inference settings. The stacked implementation has a peak GPU-memory footprint of 17.5 GB and is evaluated separately from the 3.0 s cache-enabled base path. Each TextSkill-only control is paired with the full system under the same benchmark and protocol.

B.6 Extended Qualitative Comparison

Figure 6 adds five selected examples to Figure 3. They span attribute change, removal, type change, addition, and replacement. Each row shows the reference image, modification, top-ranked candidates,

and selected skill contributions. The comparisons show how the channels address attribute, removal, and preservation requirements.

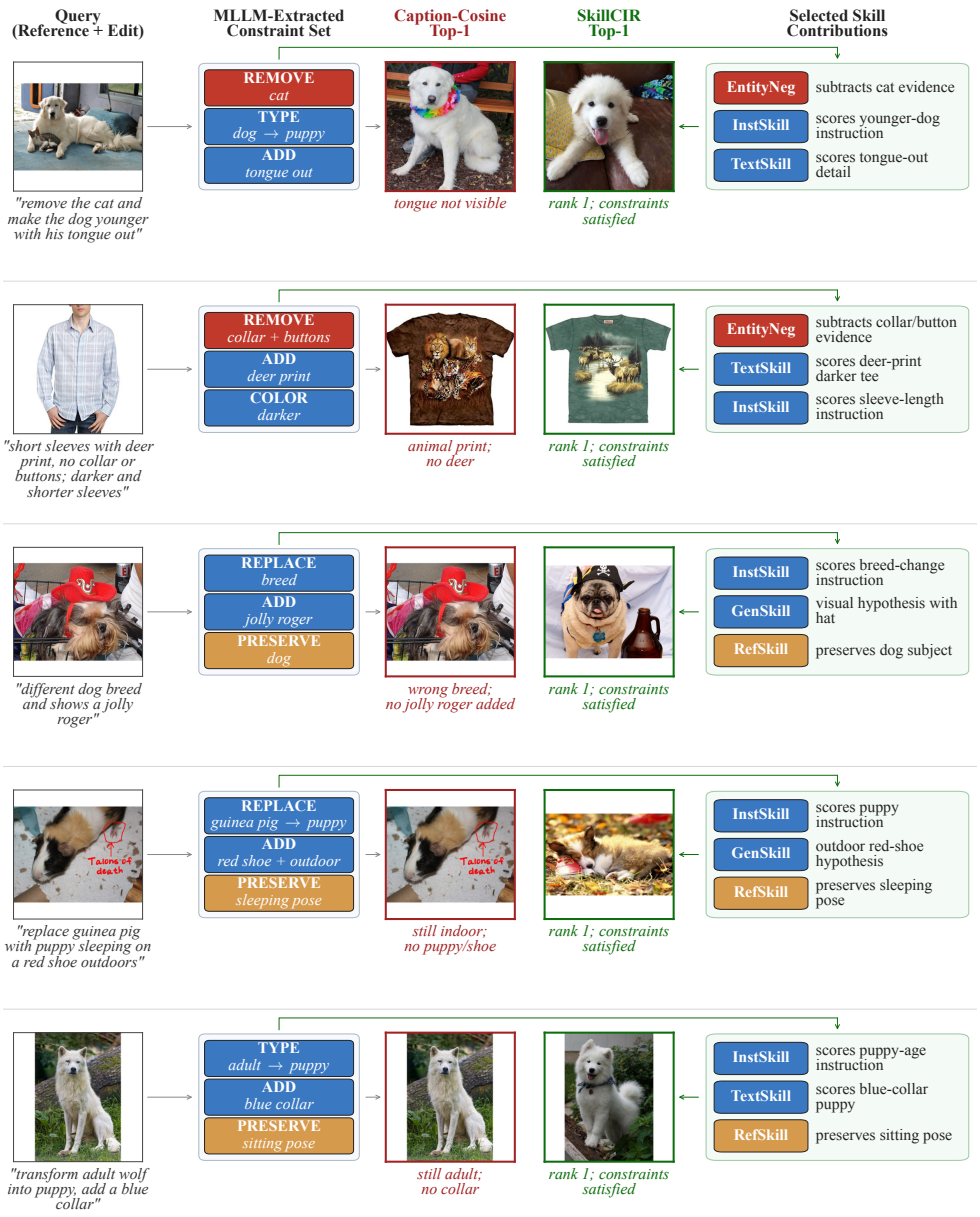


Figure 6: Top-1 qualitative comparisons on CIRR, FashionIQ, and CIRCO, covering five edit types. The figure highlights selected skill contributions; TextSkill is active for every query.

C Design-Space Exploration

This appendix summarizes post-hoc design comparisons on skill composition, routing, and verification. They were run on the benchmarks after the configuration had been selected on the 50K MegaPairs pool and frozen, and are reported for analysis only. Table 15 groups the 170+ runs by design choice. Table 16 describes the alternatives that were not retained. The comparisons use the development baseline and metric associated with each design choice.

C.1 Experiment Summary by Category

Table 15 groups these comparisons. The categories use different metrics and baselines, so they are not ranked by a shared gain column. The retained inference configuration is defined in Sections 3.4–3.5 and Appendix A. Historical score-calibration sweeps are not included as an additional term in that configuration.

Table 15: Post-hoc design categories. The final inference configuration is specified separately from these exploratory comparisons.

Design category	Reported settings
Prompt templates and chain-of-thought variants	20+
Hard routing and discrete skill selection	35+
Skill-pool composition	25+
Score-calibration sweeps	10+
PatchVerify local matching	15+
Counterfactual Presence Matching	10+
Database augmentation	5+
Tiebreaking	5+
Soft intent modulation	15+
Embedding-space fusion	10+
Score normalization	5+

C.2 Alternatives Not Retained

Table 16 records alternatives to the retained composition rules, routing strategy, and skill library. Skill-pool construction uses the external MegaPairs sample. Dataset-specific post-processing settings are also selected on this MegaPairs sample, as described in Appendix A.

Table 16: Alternatives not retained. Counts refer to the reported design comparisons. Numerical observations specify their own metric.

Direction	Expts	Observation
Prompt optimization (templates, CoT structure, system prompts)	20+	Additional variants gave limited gains. The 13–28 R@10 gap to a per-query oracle indicates variation across prompts, but does not identify the cause of the ensemble's errors.
Hard routing (primitive-based skill selection, fixed presets, confidence gates)	35+	Mixed queries can require several evidence channels. The retained planner allows multi-label skill selection (§3.3).
Soft intention modulation ($\lambda > 0$ in the per-skill score)	15+	The best setting improved pre-verification CIRR R@1 by 1.60 points. It gave no further gain after verification; the main configuration uses fixed composition.
Always-active RefSkill (general-domain queries)	5+	The tested reference-preservation setting gave little gain on general-domain queries whose targets can be visually dissimilar to the reference (*e.g.,* dog $\rightarrow$ pirate-pug).
DeltaSkill (compositional queries)	5+	Delta embeddings ($\Psi_T(\text{target}) - \Psi_T(\text{reference})$) gave little improvement on the tested compositional queries.
Reciprocal Rank Fusion (RRF)	5+	The tested rank-fusion variant underperformed the signed-score baseline. Its gain depended on the evaluation configuration.
OR / AND / NOT skill algebra in embedding space	5+	The tested operators gave limited gains. This comparison does not establish that all embedding-space combinations are less expressive.
Auxiliary skill candidates (*e.g.,* EditSkill that scores against an MLLM-edited image, GenEnsemble that averages multiple GenSkill samples)	10+	Each provided insufficient incremental value over the six-skill pool to justify the added complexity.
Lazy scheduling (deferred skill execution by predicted utility)	5+	The tested scheduling rules reduced accuracy. The available comparison does not isolate whether the loss came from inaccurate utility prediction or another scheduling effect.
Z-score normalization across skill scores	5+	Z-score normalization reduced FashionIQ R@10 by 3.2 points in the tested configuration. It was not retained.

C.3 Scope of Score-Space Composition

Score-space composition provides an explicit interface for inspecting and combining the selected evidence channels. For normalized query and gallery vectors, a linear signed cosine sum is algebraically equivalent to a signed sum of query vectors: $\sum_k \sigma_k \langle u_k, v_c \rangle = \langle \sum_k \sigma_k u_k, v_c \rangle$. Thus, linear score fusion alone has no general expressivity advantage over signed embedding fusion. The embedding-space fusion in Figure 5(b) (-0.52 CIRR R@1) instead averaged the CLIP embeddings of the active skills without signs before computing similarity, so it is not the signed sum above. The interface remains useful for reporting each channel's contribution and combining it with crop-based verification, whose max operations are nonlinear. The benefit of this design is an explicit skill interface that supports channel inspection and nonlinear verification.

D Planner Prompt and Skill Details

D.1 Planner Prompt and Output Schema

The planner (§3.3) receives a natural-language skill menu and produces the plan $(\mathcal{I}(q), \mathcal{A}(q), T_t)$. Skill selection is multi-label, allowing overlapping evidence channels to support the same query. The schema below is an excerpt of the core routing and target-description fields.

Skill pool menu.

```
You can activate any subset of the following retrieval skills:

- TextSkill: scores a candidate against the target description.
  Use for: any query (always activate).
- InstSkill: scores a candidate against the raw modification text.
  Use for: instruction-anchored queries where the rewrite may drift.
- GenSkill: generates a visual hypothesis and scores image-to-image.
  Use for: queries with rich attribute or compositional details.
- RefSkill: scores similarity to the reference image
  combined with a preservation gloss.
  Use for: local edits that should keep most of the reference.
- EntityNeg: subtracts evidence of removed entities (sigma = -1).
  Use for: queries containing "remove", "without", "no <X>", etc.
- SceneSkill: decomposes the target into scene-level components.
  Required relations/viewpoints have positive signs; must-avoid
  content and negative relations have negative signs.
  Use for: queries that change relations, viewpoint, or scene.
```

Core output fields.

```
{
  "thoughts": "<chain-of-thought reasoning>",
  "intents": [<subset of {attribute, add, remove, preserve,
                          relation, scene}>],
  "active_skills": [<subset of {TextSkill, InstSkill, GenSkill,
                                RefSkill, EntityNeg, SceneSkill}>],
  "target_description": "<one-sentence post-edit description>",
  "entities_to_remove": [...],    // populated if EntityNeg activated
  "preservation_gloss": "..."     // populated if RefSkill activated
}
```

Scoring inputs. TextSkill encodes the target description, InstSkill encodes the original modification, RefSkill encodes the reference image combined with the preservation gloss, and GenSkill encodes the generated target hypothesis. EntityNeg uses removal phrases from the planner and the extraction rules in §D.3. SceneSkill scores the typed scene and relation phrases defined in §3.2. PatchVerify uses must-contain ($\Phi^{+}(q)$) and must-avoid ($\Phi^{-}(q)$) phrases, while CPM uses the add ($\Phi^{\text{add}}(q)$) and remove ($\Phi^{\text{remove}}(q)$) phrase sets of the cached edit parse (Appendix A.7). These scorer-specific phrase inputs are separate from the routing fields shown in the excerpt. Adding a skill requires an executable scorer over gallery candidates, a menu/schema entry, and an external marginal-contribution audit before the library is frozen. No planner or retriever parameters are updated. The recorded plan makes the selected evidence channels available for inspection.

Example completion. For the query of §3.3 (reference: stingray on sandy seabed; modification: *"remove the seashells and make the water green"*), the planner returns:

```
{
  "thoughts": "The reference shows a stingray on a sandy
   seabed with scattered seashells in blue water. The
   instruction asks to remove the seashells (negation) and
   recolor the water to green (attribute). The stingray and
   the sandy seabed should be preserved. TextSkill anchors
   the recolored target description, EntityNeg handles the
   seashell removal, and RefSkill preserves the stingray
   identity. GenSkill is not needed: the edit is a single
   color change that text already specifies precisely.",
  "intents": ["remove", "attribute", "preserve"],
  "active_skills": ["TextSkill", "EntityNeg", "RefSkill"],
  "target_description": "a stingray underwater in green
                         water without seashells",
  "entities_to_remove": ["seashells"],
  "preservation_gloss": "stingray identity, sandy seabed"
}
```

D.2 Lite vs Full Planner Variants

The lite scaffold below produces a target description. It does not return the intent labels or activated skills needed for a complete plan. The full scaffold adds these fields. The keyword-router ablation retains the MLLM-generated target description and replaces only skill selection with deterministic rules.

Lite scaffold.

```
You are an image retrieval expert. Given a reference image
and a modification instruction, produce a concise target
description.

Step 1 - Observe: list the key visual attributes of the
reference (color, pattern, shape, material, style).
Step 2 - Analyze: identify what the modification asks to
change, add, or remove.
Step 3 - Synthesize: write a single-paragraph description
of the target image, incorporating the requested changes
while preserving unmentioned attributes.

Output: {"target_description": "..."}
```

Full scaffold. The full prompt extends the lite version with steps that decompose the modification and explicitly populate the plan fields:

```
You are a composed image retrieval expert. Given a reference
image and a complex modification instruction, produce a
structured plan and a target description.

Step 1 - Observe: enumerate visual elements of the reference.
Step 2 - Decompose: break the modification into atomic edits
  (attribute changes, add, remove, spatial/relational, scene).
Step 3 - Activate: choose the active skills from the menu
  whose evidence channel addresses each atomic edit.
Step 4 - Synthesize: write the target description and populate
  entities_to_remove and preservation_gloss as needed.

Output: a structured plan with the routing fields above.
```

D.3 Entity Extraction Heuristic

EntityNeg and the negative SceneSkill components require phrases identifying content to suppress. We use the following heuristic alongside the planner's `entities_to_remove` field:

1. **Removal triggers:** "without", "remove", "no", "exclude", "delete", "get rid of" followed by noun phrases.
2. **Replacement triggers:** "replace X with Y", "change X to Y", "swap X for Y"—the entity X is extracted as the entity to suppress.

3. **Negation triggers:** "not", "don't", "shouldn't" followed by noun/adjective phrases.
4. **Noun-phrase extraction:** simple regex matching for determiner + adjective* + noun patterns; no dependency parsing or NER models.

The reported inspection of CIRR samples gives approximately 85% precision for this heuristic. This measures phrase extraction, not the accuracy of the complete skill plan. Incorrect negative phrases can penalize relevant candidates.

E Reproducibility

Reproduction requires the frozen retrieval checkpoint, evaluation configuration, benchmark splits, and cached generative outputs. Caching avoids variation from repeated external-model calls. The code is publicly available at `https://github.com/Pter61/skillcir` under the MIT license.

Frozen backbones.

- CLIP retrieval encoder: OpenCLIP `ViT-bigG-14`, weights from `laion2b_s39b_b160k` (public).
- Planner: the default model in §4, with Claude Opus 4.6 [1], GPT-4.1 [30], and Qwen3-VL [3] substitutions reported separately. The cached target description is used by TextSkill.
- Text-to-image (GenSkill): GPT-image-1. The cache contains 11,426 generated images; a cached image is used only when the planner activates GenSkill for that query.

Hardware. The inference hardware is specified in §4. Retrieval uses one GPU, with generative outputs loaded from cache. The cache-enabled latency excludes first-time image generation.

Evaluation entry point. The reported runs use a single evaluation harness that loads one frozen configuration per benchmark. Reproduction should preserve the split, checkpoint, cached query outputs, and post-processing settings associated with each reported result. Bootstrap intervals use seed 42 and 1000 resamples.

Closed-model handling. MLLM responses and generated images are stored as JSON/PNG under `tool_cache/`. Reusing frozen outputs allows retrieval evaluation without live generative-model calls. Fresh generations should be treated as a separate run because model versions and outputs may change.

Determinism. Skill signs, fusion settings, and verification operators are fixed for evaluation. Cached generative outputs and the bootstrap seed should be held constant when comparing repeated retrieval runs.

F Limitations and Future Work

The current framework has the following scope and limitations.

Latency. The cache-enabled path averages about 3 s/query, including about 1.8 s for the planner. It reuses gallery embeddings and generated hypotheses, excluding cold first-time GenSkill generation. This operating point is suited to offline, batched, or exploratory retrieval. API round-trip time and uncached generation limit interactive use. Comparisons with WISER depend on the chosen generation, candidate-depth, and refinement settings.

Fixed skill pool. The planner selects from six skills frozen before target-benchmark evaluation. Their roles overlap, and mixed queries can activate several channels. A new skill requires an executable scorer, a planner menu/schema entry, and an external marginal-contribution audit. The current library does not cover multi-image queries, temporal video edits, or knowledge-grounded instructions such as finding a country from a hotel image. A tool-backed SearchSkill has been prototyped, but external web evidence can include target or near-target content and changes the

closed-gallery protocol. Evaluating this extension requires a separate protocol for external evidence. Automatic skill discovery and confidence-aware resolution among overlapping skills remain future directions.

Pseudo-triplet sample size. The 50K MegaPairs sample was chosen for the available compute budget. We did not systematically vary its size. Larger samples could expose additional evidence channels for long-tail intents.

Retriever dependence. The system uses CLIP cosine similarities. Transfer to other retriever families has not been established. Score distributions and local-verification behavior may change with the encoder, so the existing coefficients cannot be assumed to transfer unchanged.

Planner reliability and weighting. Incorrect intent labels, skill choices, and negative phrases can all affect ranking. The EntityNeg analysis is a model-assisted diagnostic with partial author checks, and its active/inactive subsets do not provide a causal estimate of routing errors. Confidence-aware weighting may help when overlapping skills disagree. The soft intent-modulation sweep improved the pre-verification stage by up to 1.60 CIRR R@1, but gave no additional end-to-end gain after verification.

Verification limits. Crop similarities can miss small details or activate on visually related content. CPM's reference terms are constant across candidates, so their subtraction alone does not enforce the requested change. The failure categories in Appendix B.2 describe observed retrieval errors and do not identify an unavoidable encoder ceiling.

G Broader Impacts

Potential applications. Training-free composed retrieval may be useful for visual search in domains with limited annotated triplets, such as product catalogs and image archives. The recorded skill plan and per-channel scores can help inspect query behavior. Application-specific reliability, including accessibility or medical use, has not been evaluated here.

Risks and mitigations. Constraint-rich image retrieval could support unwanted surveillance or searches over private galleries. Deployments should obtain appropriate consent, restrict gallery access, and define acceptable query uses. Biases in the planner or generator may affect the target description, visual hypothesis, and retrieved results. GenSkill's generated image is used as retrieval evidence rather than returned as the search result, but remains subject to the generator's content policies. Logs of plans, active skills, and results can support application-specific audits.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe the fixed skill library and its scope. Tables 1 and 2 support the reported ViT-bigG/14 comparisons, including the 2.77-point CIRR R@5 gain and the 0.48-point R@1 deficit; latency claims state the cache convention and comparator configuration.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix F discusses planner errors, the fixed skill pool, the external sample budget, retriever dependence, verification limits, and uncached generation cost.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper presents a retrieval method and empirical evaluation, with no formal theoretical results requiring proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 3–4 and Appendices A, D, and E describe the pipeline, skill pool, planner prompts, composition rule, verification operators, backbones, and evaluation protocols needed to reproduce the main results, and the code is released at `https://github.com/Pter61/skillcir`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The evaluation benchmarks and retrieval backbone are publicly available and cited. The code is publicly available at `https://github.com/Pter61/skillcir` under the MIT license, and Appendices A, D, and E give the configuration, prompts, and cache conventions.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 and Appendix A identify the evaluation splits, backbones, scoring rules, and dataset-specific post-processing settings, which are selected on the external 50K MegaPairs pool rather than on benchmark splits. The method uses frozen models without training or an optimizer; the refined-benchmark and reranking comparisons state their separate protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Appendix B.2 reports percentile bootstrap 95% intervals for FashionIQ category R@10, using 1000 resamples and seed 42. Their evaluation-set scope is explicit: no paired significance test against published baselines is claimed, and unavailable intervals are not supplied.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4 identifies the single NVIDIA H100 inference setting. Appendix B.4 reports the per-stage latency breakdown, the WISER operating points under the same hardware, and the 17.5 GB peak GPU memory of the optional reranker.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work evaluates retrieval on existing benchmarks without recruiting human participants. Appendix G discusses privacy, bias, and misuse concerns and appropriate deployment precautions.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix G discusses possible uses in visual search and image archives, along with risks involving privacy, surveillance, and model bias.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new pretrained model, image generator, or scraped dataset. GenSkill uses an existing generator for retrieval hypotheses; potential deployment risks are discussed in Appendix G.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets, pretrained models, and baseline methods are cited, and each is used for non-commercial research under the license or terms published by its creators. OpenCLIP is released under the MIT license and Qwen3-VL under Apache-2.0; FLUX.1 Kontext and SDXL-Turbo are used under their non-commercial model licenses; GPT-4o, GPT-image-1, and Claude models are accessed through their providers' API terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code at `https://github.com/Pter61/skillcir` is documented with a README and an MIT license. Appendices A, D, and E describe the configuration, prompts, and cache conventions.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper reports no crowdsourcing or participant study. The annotation quality checks described in Appendix B.2 were performed by the authors.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper reports no human-subject experiment or recruitment of study participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Sections 3.2 and 3.3 describe GPT-4o-assisted intent taxonomy construction, labeling, target descriptions, and skill selection. Appendix B.2 describes model-assisted diagnostic annotation, while Appendices A and D document model substitutions and prompts; GenSkill uses GPT-image-1 for visual hypotheses.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.