PRICAI 2026 · PAPER 247 · ORAL PRESENTATION SCRIPT (~12 MINUTES)

AutoConcept: Training-Free Concept-Guided Reranking for Metadata-Available Composed Image Retrieval

Tianyu Wang (Soochow University), Tianjiao Wu (INSA Lyon) · Session F11.2, 19 Nov 2026, Room R215 · ~1,520 words

Slide 1: Title Slide

0:00 – 0:30 (30s)

Good afternoon, everyone. Today, I'm going to share our work, **AutoConcept**. It's a training-free, concept-guided reranking method for composed image retrieval. This is joint work by Tianyu Wang from Soochow University and Tianjiao Wu from INSA Lyon.

The main question we're asking is simple: when product metadata is available, can we build an explicit, inspectable concept memory to rerank candidate images—without training a reranker or calling an MLLM for every query?

Slide 2: 01. Motivation — Composed Image Retrieval

0:30 – 1:20 (50s)

Let's start with the task. In composed image retrieval, or CIR, the user gives two inputs: a reference image and a text modification. Look at this example. The user starts with a sleeveless black lace dress and asks for a dress that *"has short sleeves, is less fitted, and is solid black with white accents."* So the system needs to keep the main look of the reference dress, while changing those exact details.

Most CIR models merge the image and text into one vector and search the whole gallery. The problem is, the top results often look similar overall, yet miss the exact attributes the user asked for. That's where we step in—we rerank this top candidate list. Just one quick note: the target image on the right is only here to show the goal; our reranker never sees it.

Slide 3: 01. Motivation — Why Add Explicit Concepts?

1:20 – 2:15 (55s)

So why add explicit concepts? After the first stage, we already have a composed query, a fixed top-K candidate pool, and a similarity score for each item. At the same time, in e-commerce or product galleries, items also come with metadata—like titles, tags, and attribute descriptions. These give us clear, named concepts such as sleeve length, pattern, or neckline.

Instead of just matching raw text or paying for slow MLLM calls on every query, we organize this evidence into a structured concept memory. And we want to test whether this concept structure actually beats simple text-to-metadata matching. Our method, AutoConcept, acts as a lightweight second-stage layer: the base retriever stays frozen, there's zero reranker training, and zero per-query MLLM calls.

Slide 4: 02. Method — The AutoConcept Framework

2:15 – 3:30 (75s)

> **Pointer Cue:** 先从左到右指上方的主检索流，再从左下角的 Concept Source 指向右下角的 Output。

Let's walk through how AutoConcept works. Look at the top row first. That's the standard retrieval pipeline. We encode the reference image x_r and modification text t_q, fuse them, and a frozen CIR model gives us the top-K candidates with their base scores. Only after we get this candidate pool do we bring in their metadata m_i.

Now look at the bottom row. We build an explicit concept memory from item texts. When a query comes in, a relevance gate picks out only the concepts related to the user's request. Then, we match those active concepts against each candidate's metadata to get positive and negative alignment scores. Finally, we calibrate and fuse these scores with the base retrieval score to get our reranked top-K list. Because the base retriever stays completely untouched, we can plug AutoConcept into different retrieval models.

Slide 5: 02. Method — Building an Inspectable Concept Memory

3:30 – 4:30 (60s)

How do we build this concept memory? It takes four straightforward steps. First, we extract short attribute phrases from item texts. Second, we cluster near-duplicates using their embeddings so similar phrases get grouped together. Third, we score their positive and negative evidence. Fourth, we run quality filters to drop generic words, conflicting labels, and weak concepts.

As you can see in the table, each entry in memory stores five things: its attribute **name**, its **polarity**, its **evidence strength**, a **CLIP visual grounding** score, and the **supporting items**. Why does this matter? Because the memory is completely transparent—we can inspect which concept fired and which items supported it. Plus, this interface is flexible: it can take controlled training-side texts, query-only phrases, or real human tags.

Slide 6: 02. Method — Query Activation and Score Calibration

4:30 – 5:45 (75s)

> **Pointer Cue:** 依次指向左侧第 1 到第 4 步；具体标定公式留到 Q&A 环节展示 Backup Slide 4。

Once we have the memory, how do we score a candidate at test time? There are four steps here. First is the **query-relevance gate**. We check the cosine similarity between the modification text and each concept name, and only keep concepts above a threshold τ. That way, irrelevant concepts don't add noise.

Second is **metadata alignment**. We match each candidate's metadata against the active concepts and use **max pooling**. We choose max pooling because product metadata is sparse—if a candidate clearly matches one requested attribute, we don't want to dilute that signal just because other concepts aren't mentioned.

Third, we combine the base score, the positive concept score, and a small negative penalty. Fourth, we do a **closed-form outer interpolation** with weight λ_q. Both the internal weights and λ_q adjust automatically for each query based on concept activation strength and the score gap between the top-1 and top-2 base candidates. Basically, if concept evidence is weak, or the base retriever is already super confident, we keep more weight on the base score. And all bounds are fixed beforehand—zero training needed.

Slide 7: 03. Experiments — Evaluation on Fixed Candidate Pools 5:45 – 6:40 (55s)

Let's move to the experiments. Our main dataset is **FashionIQ**, with 6,016 queries and fixed top-100 candidate pools. We test two different first-stage retrievers—**WeiMoCIR** and **LinCIR**—across five random seeds for concept evidence. We encode text and concepts with `all-MiniLM-L6-v2`, use `CLIP ViT-B/32` for visual grounding, and report Recall@10, Recall@50, MRR, and NDCG@10.

To make sure there is zero data leakage, we enforce strict information boundaries. Our main concept evidence comes only from same-category training-side item texts. We never touch the target ID, target image, target metadata, or candidate ranks when building evidence. Candidate metadata only enters after retrieval, and is scored uniformly across all candidates.

Slide 8: 03. Experiments — Gains on Two First-Stage Retrievers 6:40 – 7:45 (65s)

Here are our main Recall@10 results. On **WeiMoCIR**, AutoConcept boosts Recall@10 from **11.25%** to **13.79%**—a gain of **2.54 percentage points**. On **LinCIR**, which is a much stronger baseline, it jumps from **26.05%** to **30.09%**, giving us **4.04 percentage points** of improvement.

Notice how stable this is across the five seeds: the standard deviations are only 0.11 and 0.18 percentage points. Under a query-level paired bootstrap test, the gains across all four metrics are statistically significant with $p < 0.001$. The main takeaway here is that the exact same training-free module improves two very different retrievers while keeping their candidate pools fixed.

Slide 9: 03. Experiments — Beyond Direct Metadata Matching 7:45 – 8:55 (70s)

Now, a very fair question is: does this gain come from our structured concept memory, or just from adding metadata in general? To check this, Table 5 compares several metadata-aware rerankers on the exact same WeiMoCIR pool.

Simply matching the raw query text to metadata already brings Recall@10 up to 13.18%. Extracting query attributes and doing constraint matching push it to 13.63% and 13.83%. However, under the exact same query-only input, **Query-only AutoConcept** hits **14.00%** Recall@10, and also beats those baselines on MRR and NDCG@10.

We also include two reference rows. Controlled-evidence AutoConcept gets 13.75% on this seed, while concatenating the full reference-item metadata with the query text reaches 14.30% simply because it feeds in much richer text. Also, in the paper, we show that shuffling the query-evidence alignment causes a big drop—proving that matching the right concepts to the right query really matters.

Slide 10: 03. Experiments — What Drives the Improvement? 8:55 – 9:50 (55s)

Next, we ran ablations to see which parts of AutoConcept actually drive the gains. Three clear findings stand out.

First, **positive concepts do most of the heavy lifting**. If we remove the negative penalty, Recall@10 is 13.80%, almost identical to the full model at 13.75%. Making the negative penalty stronger actually drops it slightly to 13.63%. So negative concepts work best as a gentle, inspectable guardrail.

Second, **max pooling is critical**. Switching to mean pooling drops Recall@10 all the way down to 12.40%, because averaging dilutes sparse attribute matches.

Third, **outer interpolation matters**. Without it, performance drops to 13.02%, showing that balancing concept scores with base retriever confidence keeps early ranks safe.

Slide 11: Qualitative Trace — A Concept Trace Explains a Change in Ranking

9:50 – 10:45 (55s)

> **Pointer Cue:** 先指中间的 Active Concepts，再指右上角第 22 名的虚线框，最后指右下角第 3 名的虚线框。

Let's look at a concrete example to see how this works under the hood. Going back to our opening query on the left, the memory holds several concepts. Our gate activates **"black-white pattern"** and **"shorter sleeves"** as positive constraints—while ignoring **"brighter color"** since it doesn't apply—and activates **"cream sheer sleeves"** as a negative constraint.

In the baseline WeiMoCIR ranking at the top right, the true target dress is buried at **rank 22**, while several sheer or light-colored dresses sit in the top 10. After AutoConcept reranks the pool, those mismatched items drop out, and the target dress jumps from **rank 22 all the way to rank 3**. Best of all, we can trace the exact concepts that caused this shift.

Slide 12: 04. Discussion — A Pilot with Human Concept Labels

10:45 – 11:35 (50s)

Finally, we ran a small pilot study to see if our memory interface can also use real human labels. Participants labeled FashionIQ items and picked optional tags from a fixed vocabulary, which we turned into user-level concept memories. Even though human tags are sparse and noisy, all four metrics still improved—nudging Recall@10 from 11.25% to **11.59%**.

This pilot also highlights our practical boundaries. First, the gallery needs reasonably good metadata. Second, sparse human histories naturally bring smaller gains. Third, since we only rerank the top-K pool, if the first-stage retriever misses the target completely, we can't recover it.

Slide 13: 04. Discussion — Conclusion

11:35 – 12:00 (25s)

To wrap up, AutoConcept adds an explicit, training-free concept memory on top of composed image retrieval:

1. **Inspectable memory:** It makes second-stage reranking transparent and editable without expensive MLLM calls.
2. **Consistent early-rank gains:** Query-relevant activation and score calibration improve early-rank metrics across two fixed retrievers.
3. **Clear empirical positioning:** Our metadata controls show where structured concepts add value and where richer text remains strong.

Thank you all for listening! I'd be happy to take any questions.

Q&A Quick-Response Cue Cards (Backup Slides 1–7) 3:00 Discussion

Q1: What is the runtime / latency overhead? (Backup Slide 3)

> *"On an Apple M5 MacBook Air, once embeddings are cached, the scoring loop takes just 0.32 milliseconds per query—that's only about 5.6% of the base retrieval time. Even with full uncached text encoding, it's under 2 milliseconds. Complexity scales linearly as $O(K(|A_q^+| + |A_q^-|)d)$."*

Q2: How sensitive is it to the relevance gate or candidate pool size? (Backup Slide 2)

> *"The relevance gate is pretty robust—anything between $\tau = 0.30$ and 0.40 gives very similar gains. As for pool size $K \in \{50, 100, 200, 500\}$, top-100 hits the sweet spot: top-50 misses too many targets, while top-200 or 500 brings in too much metadata noise."*

Q3: Does concept evidence use any ground-truth target information? (Slide 7)

> *"Not at all. When building concept evidence, we strictly exclude the target ID, target image, target metadata, and candidate ranks. Candidate metadata is only used at reranking time and scored uniformly across the whole pool."*

Q4: Does it generalize to other datasets? (Backup Slide 6)

> *"We also ran a secondary check on Fashion200K, where AutoConcept boosts WeiMoCIR Recall@10 from 65.60% to 70.49%."*