BEYOND POST-HOC REWARDS: PRIVILEGED REASONING EXPLORATION FOR MULTIMODAL EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Advances in Multimodal Large Language Models (MLLMs) have driven substantial progress in multimodal embeddings. Recent methods exploit generative capabilities by augmenting embedded inputs with self-generated reasoning to produce more informative embeddings. However, conventional retrieval supervision for training embedding models specifies pairwise relevance relations without explicitly guiding the reasoning used for input augmentation. Such reasoning is neither directly observed nor uniquely determined by the input, allowing multiple plausible traces to introduce semantic cues of varying retrieval utility. Existing approaches largely optimize reasoning conditioned on the input alone, limiting the exploration of retrieval-relevant traces. We introduce **PILOT**, a framework that models reasoning as a latent variable and instantiates its prior and posterior distributions as distinct reasoning modes of a shared MLLM. The prior conditions only on the input, whereas the posterior uses additional training-time relational evidence to guide reasoning exploration. To translate the posterior's broader exploration into inference-time improvements, we derive an importance-weighted variational objective and instantiate it as a prior–posterior co-evolution framework that improves both modes under a shared inference-compatible reward while selectively transferring posterior trajectories to the prior according to their utility and prior reachability. On the 78-task MMEB-V2 benchmark, PILOT outperforms the primary baseline across all 12 task groups at both 2B and 7B parameters, improving overall scores by 3.2 and 3.3 points, respectively, with the largest gains in visual-document retrieval (+6.1 and +6.4 points). Our results show that relational supervision is most valuable not as a post-hoc reward but as a signal for shaping how reasoning is explored in training reasoning-enhanced embedding models.

1 INTRODUCTION

Multimodal embeddings map heterogeneous inputs such as text, images, and videos into a shared dense representation space, supporting applications from cross-modal retrieval to clustering and recommendation (Zhai et al., 2022; Frome et al., 2013; Wei et al., 2019; Yang et al., 2022). Recent advances in multimodal large language models (MLLMs) (Singh et al., 2025; Comanici et al., 2025; Bai et al., 2025; Wang et al., 2025) have shifted the field beyond the dual-encoder vision–language models (Radford et al., 2021; Jia et al., 2021) toward instruction-aware universal multimodal embeddings (UME) models, which follow nuanced instructions across tasks such as visual question answering and video moment retrieval (Meng et al., 2026), and generalize better in zero-shot settings (Jiang et al., 2025; Meng et al., 2026; Gu et al., 2025; Lan et al., 2025). Yet most UME models still use the MLLM purely as an encoder, leaving its generative reasoning capacity untapped.

Chain-of-thought (CoT) reasoning improves MLLMs performance across diverse multimodal tasks by enabling a deeper understanding of complex inputs (Zhang et al., 2024b). Several works extend this idea to UME by augmenting each query or candidate with a self-generated reasoning trace and encoding the input together with its trace into a single embedding (Cui et al., 2026; Lan et al., 2026; Sun et al., 2026a). Despite their effectiveness, existing approaches face a fundamental mismatch between reasoning generation and retrieval supervision. During training, queries are paired with relevant candidates with other candidates serving as negatives. These positive–negative relations

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

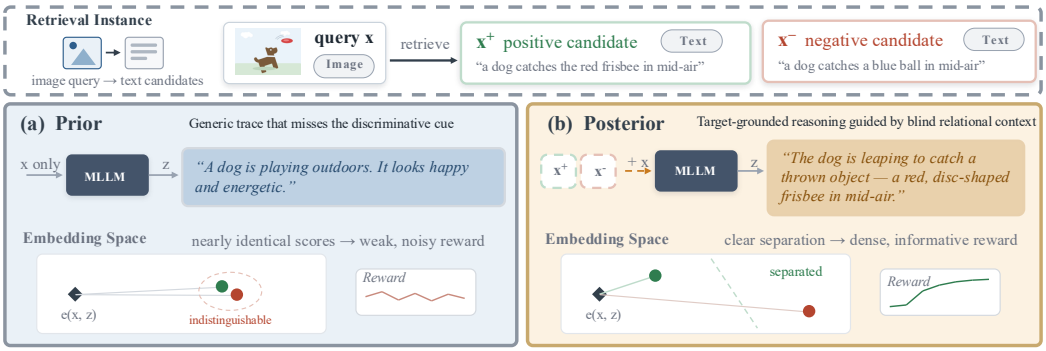

Figure 1: **Relationally guided reasoning exploration in PILOT**. (a) Input-only prior reasoning may miss discriminative cues, yielding weak embedding separation. (b) The posterior uses training-time relational evidence to discover target-grounded cues that help distinguish positive and negative candidates. Embedding layouts and reward curves are schematic.

identify relevant matches but leave the reasoning needed to distinguish them unspecified. Consequently, multiple reasoning traces may be relevant to the same input yet differ substantially in their ability to distinguish positive candidates from negative ones. Existing methods sample reasoning from the individual inputs alone, with relational supervision applied only through post-hoc rewards. This creates a fundamental exploration bottleneck. Without access to relational evidence, the model is unlikely to discover the discriminative reasoning needed to distinguish relevant candidates from distractors, and post-hoc rewards can only reinforce already sampled sub-optimal trajectories.

We address this bottleneck with **PILOT** (Privileged-Information-guided Latent Reasoning Optimization and Transfer), a framework that models reasoning traces as latent variables. This formulation naturally distinguishes an inference-time **prior** conditioned only on the target input from a training-time privileged **posterior** that additionally uses relational evidence to infer retrieval-relevant reasoning. We instantiate both distributions as distinct reasoning modes of a single MLLM. Explicitly identifying the positive and negative counterparts allows the model to explain a given relevance relation without first determining which counterpart better matches the target. To encourage more fine-grained, discriminative reasoning, we construct a *blind contrastive posterior* that observes the target alongside a randomly ordered positive–negative pair with relevance labels withheld. The model must compare both candidates against the target and predict which is positive, encouraging it to identify subtle semantic distinctions that determine relevance. Using mode-specific prompt templates, a stronger annotation MLLM synthesizes reasoning traces for both distributions. Posterior traces are retained only when the annotator correctly identifies the positive counterpart. The prior traces and verified posterior traces jointly initialize the two modes in the shared embedding MLLM through supervised fine-tuning (SFT).

Although the privileged posterior broadens the exploration of retrieval-relevant reasoning, only the input-conditioned prior is available at inference. Moreover, not every posterior trajectory is useful or reproducible from the target input alone, making indiscriminate distillation undesirable. We therefore derive an importance-weighted autoencoding (iWAE) objective (Burda et al., 2015) with a mixture of the prior and posterior as the proposal distribution. Its gradient naturally decomposes into on-policy improvement of the prior, on-policy improvement of the posterior, and importance-weighted transfer into the prior. We instantiate these signals in a prior–posterior co-evolution procedure that uses GRPO (Shao et al., 2024) for the two policy-improvement terms and weighted SFT for posterior-to-prior transfer. Together, these updates allow privileged and inference-time reasoning to co-evolve under a unified objective, with downstream task reward and prior reachability governing which trajectories are exchanged rather than imposing a fixed teacher–student hierarchy.

We evaluate our overall approach on MMEB-v2 (Meng et al., 2026). PILOT outperforms the primary baseline across all 12 task groups at both 2B and 7B scales, improving overall scores by 3.2 and 3.3 points, respectively, with the largest gains in visual-document retrieval. Ablations studies further demonstrate the benefits of dual-mode initialization, adaptive posterior exploration, and importance-weighted transfer.

Our main contributions can be summarized as follows:

- We identify a fundamental exploration bottleneck in reasoning-enhanced multimodal embeddings. Existing methods generate reasoning from individual inputs while using positive–negative relations only for post-hoc evaluation, limiting the discovery of discriminative traces.
- We introduce **PILOT**, a framework that models reasoning as a latent variable, pairing an input-conditioned prior for inference with a privileged posterior constructed from blind relational evidence during training.
- We derive a mixture importance-weighted variational objective that motivates prior–posterior co-evolution, combining joint policy improvement with selective transfer to the prior based on retrieval utility and prior reachability.
- Experiments on the 78-task MMEB-V2 benchmark show that PILOT outperforms primary multimodal embedding baselines at both 2B and 7B scales, highlighting the value of relational supervision not only for evaluating reasoning, but also for guiding its exploration and selective transfer to the inference-time prior.

2 Related Work

2.1 Universal Multimodal Embedding

Recent methods adapt MLLMs into UME models through contrastive learning and hard-negative mining, but typically encode inputs directly (Jiang et al., 2025; Meng et al., 2026; Lin et al., 2025). Think-Then-Embed (Cui et al., 2026) and Embed-RL (Jiang et al., 2026) introduce separate reasoner–embedder pipelines with the embedder encoding each input together with a reasoning trace generated by the reasoner. UME-R1 unifies reasoning generation and embedding extraction within a single model trained with SFT and RL, while PLUME replaces explicit textual traces with compact continuous reasoning (Lan et al., 2026; He et al., 2026). MMEmb-R1 and UniME-R1 further incorporate relational signals through pair-aware trace selection and retrieval-centric reasoning construction (Wang et al., 2026; Sun et al., 2026b). These signals primarily evaluate, select, or refine reasoning after generation, whereas our privileged posterior uses training-time relational evidence to directly guide trajectory exploration while preserving input-only inference.

2.2 Learning with Privileged Information

Privileged information refers to auxiliary evidence available during training that can guide learning but is unavailable to the deployed model at inference (Vapnik & Izmailov, 2015). Prior work in language-model reasoning uses such information either to distill a fixed privileged teacher into a deployable model or to jointly and iteratively improve privileged and deployable reasoning models (Agarwal et al., 2024; Penaloza et al., 2026; Lin et al., 2026; Liang et al., 2026; Zhou et al., 2026; Wen et al., 2026). However, these methods primarily address answer-generation tasks, where privileged information reveals or supports the reconstruction of a target output. In multimodal embedding, a relevant counterpart is not an output to reconstruct but a separate input whose relation to the input, relative to negatives, defines the supervision. This difference changes both how relational evidence should condition privileged reasoning and how reasoning utility should enter the learning objective. We address this setting by constructing a privileged posterior from positive–negative relations and co-evolving it with the input-conditioned prior under a unified latent variable objective.

3 Methodology

3.1 Problem Formulation

We consider multimodal retrieval where queries and candidates may contain text, images, videos, or interleaved multimodal content. Let x denote a target input to be embedded, which can be either a query or a candidate. Its observed retrieval relation is denoted by $\mathcal{R} = (x^+, \mathcal{X}^-)$, where x^+ is a positive counterpart and $\mathcal{X}^-$ is a set of negatives. Following the unified architecture of UME-R1, the model first samples a reasoning trace z and then encodes it together with the target input:

$$z \sim p_\theta(z \mid x), \qquad \mathbf{e}_\theta(x, z) = E_\theta(x, z). \tag{1}$$

Here, $\mathbf{e}_\theta(x, z) \in \mathbb{R}^d$ is the normalized embedding, and E_θ extracts and normalizes the final hidden state obtained by encoding x together with z. Retrieval supervision specifies $\mathcal{R}$ but does not provide

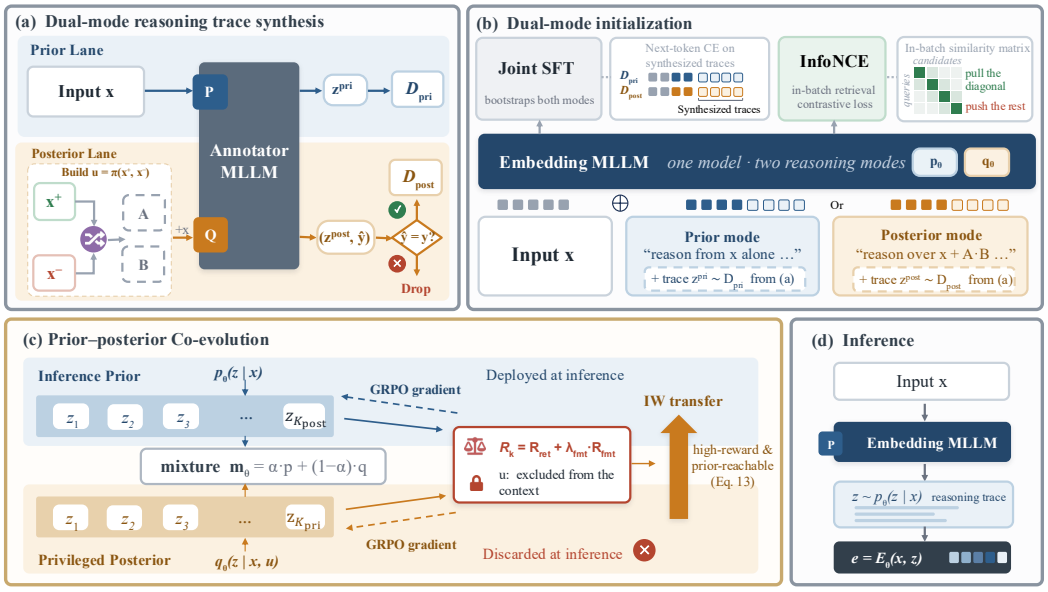

Figure 2: **Overview of PILOT. (a)** A stronger offline annotator, given only a shuffled, unlabeled candidate pair u, synthesizes prior and consistency-filtered posterior reasoning traces. **(b)** Joint SFT and contrastive learning initialize both reasoning modes in a shared MLLM (Eq. 8). **(c)** GRPO optimizes both modes using rewards evaluated under the prior context (Eq. 12), while importance-weighted fitting transfers trajectories to the prior according to retrieval utility and prior reachability (Eq. 13). **(d)** Inference runs the prior alone.

a corresponding reasoning trace. Moreover, multiple traces may be plausible for the same input while emphasizing different semantics and yielding substantially different retrieval outcomes. We therefore propose to treat z as a latent variable and marginalize over possible reasoning traces:

$$p_\theta(\mathcal{R} \mid x) = \sum_z p_\theta(z \mid x) p(\mathcal{R} \mid x, z). \tag{2}$$

Here, $p_\theta(z \mid x)$ serves as the reasoning *prior*. It generates each trace from its corresponding input alone, matching the independent encoding procedure required at inference time. We connect the conditional likelihood to the downstream retrieval reward through

$$p(\mathcal{R} \mid x, z) \propto \exp\left(\beta R_{\text{ret}}(\mathbf{e}_\theta(x, z), \mathcal{R})\right), \tag{3}$$

where R_{ret} evaluates the embedding induced by z and β controls the concentration on high-utility trajectories. Equation 2 exposes the exploration bottleneck of existing methods. In practice the sum is approximated with a few traces from the input-only prior, and retrieval rewards can only score these after the fact. Reasoning that separates the positive from plausible negatives receives no signal unless it is sampled, yet it must be discovered without ever observing those candidates. Conditioning on the observed relation induces an ideal posterior distribution over the reasoning trace:

$$p_\theta^\star(z \mid x, \mathcal{R}) = \frac{p_\theta(z \mid x) p(\mathcal{R} \mid x, z)}{p_\theta(\mathcal{R} \mid x)}, \tag{4}$$

which reweights the input-only prior according to the retrieval utility of each reasoning trajectory. This posterior therefore places greater probability on reasoning trajectories that explain the observed retrieval relation and that the prior should ultimately learn to recover. However, it is intractable to compute exactly for sampling. We therefore introduce a privileged variational *posterior* $q_\theta(z \mid x, u)$ where u denotes the relational context associated with $\mathcal{R}$. This posterior serves as an informed proposal for exploring reasoning trajectories that are difficult to discover from individual inputs alone. Both distributions are instantiated by the same MLLMs under different conditioning prompts.

3.2 Dual-Mode Initialization

The latent-variable formulation motivates a privileged posterior that leverages retrieval relations to explore reasoning trajectories less likely under the input-only prior. To elicit more discriminative

reasoning, we condition it on randomly ordered, unlabeled positive and negative counterparts, yielding a *blind contrastive posterior*. For each target input x, we sample its positive counterpart x^+ and a negative counterpart $x^- \in \mathcal{X}^-$, randomly permute them, and assign anonymous identifiers:

$$u = (c_A, c_B) = \pi(x^+, x^-), \qquad y \in \{A, B\}, \tag{5}$$

where π is a random permutation and y denotes the hidden position of the positive candidate. The privileged posterior observes the target and the unlabeled candidate pair, but not y:

$$z^{\text{post}} \sim q_\theta(z \mid x, u). \tag{6}$$

By withholding relevance labels and randomizing candidate order, this construction encourages the model to compare the counterpart against the input and identify the semantic distinctions that determine relevance, eliciting more discriminative reasoning. However, compact embedding MLLMs may struggle to generate such reasoning in a zero-shot setting. Indiscriminately distilling privileged traces into $p_\theta(z \mid x)$ ignores privileged traces' varying retrieval utility and prior reachability. We thus synthesize traces offline with a stronger model $\mathcal{A}$ and initialize both reasoning modes through SFT. Conditioned only on the target input x, $\mathcal{A}$ generates a prior trace z^{pri}. Conditioned on x and the blind contrastive context u, it jointly generates a posterior trace and predicts the positive candidate:

$$(z^{\text{post}}, \hat{y}) \sim \mathcal{A}(\cdot \mid x, u). \tag{7}$$

We retain a synthesized posterior annotation only if $\hat{y} = y$. This consistency check filters traces generated from incorrect candidate comparisons or hallucinated relations, ensuring that retained posterior traces reflect a correct interpretation of the retrieval supervision.

Finally, We use the synthesized prior and verified posterior traces $\mathcal{D}_{\text{pri}} \cup \mathcal{D}_{\text{post}}$ to initialize the embedding model with a joint generation-and-retrieval objective:

$$\mathcal{L}_{\text{init}} = -\mathbb{E}_{\mathcal{D}_{\text{pri}}} \log p_\theta(z^{\text{pri}}|x) - \mathbb{E}_{\mathcal{D}_{\text{post}}} \log q_\theta(z^{\text{post}} \mid x, u) + \lambda_{\text{ret}} \ell_{\text{NCE}}\big(\mathbf{E}_\theta(\mathcal{B})\big) \tag{8}$$

where $\mathbf{E}_\theta(\mathcal{B})$ denotes the collection of reasoning-conditioned query and candidate embeddings in minibatch $\mathcal{B}$, $\ell_{\text{NCE}}(\cdot)$ is the standard in-batch contrastive loss and λ_{ret} controls its weight. The first two terms bootstrap the inference-time prior and privileged posterior mode, respectively, while the InfoNCE loss aligns the reasoning-augmented representations with retrieval discrimination.

3.3 Prior–Posterior Co-Evolution

Dual-mode initialization establishes the prior and privileged posterior, but leaves open how privileged exploration should improve inference-time reasoning. We therefore develop a prior–posterior co-evolution procedure that jointly optimizes both reasoning modes with retrieval rewards and couples them through importance-weighted prior fitting.

To combine inference-compatible reasoning with privileged exploration, we construct the variational proposal as a mixture of the prior and posterior:

$$m_\theta(z \mid x, u) = \alpha p_\theta(z \mid x) + (1 - \alpha) q_\theta(z \mid x, u), \tag{9}$$

where $\alpha \in (0, 1)$ determines the proportion of prior rollouts in the overall proposals. In practice, we draw K_{pri} trajectories from the prior and K_{post} trajectories from the posterior, with $\alpha = K_{\text{pri}}/(K_{\text{pri}} + K_{\text{post}})$. Using the reward-induced likelihood introduced in Equation 3, the latent evidence can be estimated with $K = K_{\text{pri}} + K_{\text{post}}$ samples from the mixture proposal through the importance-weighted lower bound:

$$\begin{aligned} \mathcal{J}_{IW} &= \mathbb{E}_{z_{1:K} \sim m_\theta} \left[\log \frac{1}{K} \sum_{k=1}^{K} w_k \right] \leq \log p_\theta(\mathcal{R} \mid x), \\ w_k &= \frac{p_\theta(z_k \mid x) p(\mathcal{R} \mid x, z_k)}{m_\theta(z_k \mid x, u)} \propto \frac{p_\theta(z_k \mid x) \exp\left(\beta R_{\text{ret}}(x, z_k)\right)}{m_\theta(z_k \mid x, u)}. \end{aligned} \tag{10}$$

The importance weight combines two complementary criteria. The reward term measures the retrieval utility of a trajectory, while the ratio $p_\theta(z_k \mid x)/m_\theta(z_k \mid x, u)$ measures its reachability under the inference-time prior.. Let $\bar{w}_k = \frac{w_k}{\sum_j w_j}$ denote the normalized importance weight. The gradient of Equation 10 separates into three types of learning signals:

$$\nabla_\theta \mathcal{J}_{\text{IW}} = \mathbb{E}_{p_\theta}[A^p \nabla_\theta \log p_\theta(z|x)] + \mathbb{E}_{q_\theta}[A^q \nabla_\theta \log q_\theta(z|x,u)] + \mathbb{E}_{m_\theta}[\bar{w} \nabla_\theta \log p_\theta(z|x)]\,. \tag{11}$$

Detailed derivation can be seen in Appendix A.3. Here, A^p and A^q denote the score-function learning signals for prior- and posterior-sampled trajectories, respectively, and the expectations are taken over the prior, posterior, and mixture distributions. Samples originating from the prior improve the prior as an on-policy reasoning distribution, while samples originating from the posterior similarly improve privileged exploration. In addition, the prior likelihood in the numerator provides an importance-weighted learning signal that transfers useful mixture samples toward the prior.

Directly optimizing the full score-function gradient can cause high variance. We therefore approximate the two policy-improvement terms using GRPO. For every sampled trajectory, we compute an inference-compatible reward

$$R_k = R_{\text{ret}}\big(\mathbf{e}_\theta(x, z_k), \mathcal{R}\big) + \lambda_{\text{fmt}} R_{\text{fmt}}(z_k). \tag{12}$$

where R_{fmt} denotes the format reward function and λ_{fmt} controls its effect. Although posterior trajectories are generated using u, the privileged evidence is excluded when constructing $\mathbf{e}_\theta(x, z_k)$. Thus, privileged information can improve the reward only by inducing a more useful reasoning trace, rather than by directly entering the evaluated embedding. Rewards are then normalized within each reasoning mode to obtain group-relative advantages, producing the losses $\mathcal{L}^{\text{pri}}_{\text{GRPO}}$ and $\mathcal{L}^{\text{post}}_{\text{GRPO}}$. We instantiate the cross-distribution transfer term using weighted supervised fine-tuning and retain only the posterior-sampled component to avoid overly reinforcing the prior samples. The objective is defined as:

$$\mathcal{L}_{\text{transfer}} = -\sum_{k:s_k=\text{post}} \text{sg}\,[\bar{w}_k] \log p_\theta(z_k \mid x), \tag{13}$$

where $\text{sg}[\cdot]$ stops gradients through the transfer weights. Unlike conventional distillation, this objective does not require the prior to imitate every posterior trajectory. Posterior trajectories contribute strongly only when they achieve high reward and remain sufficiently reachable under the prior.

The final PILOT objective is

$$\mathcal{L}_{\text{PILOT}} = \mathcal{L}^{\text{pri}}_{\text{GRPO}} + \lambda_{\text{post}} \mathcal{L}^{\text{post}}_{\text{GRPO}} + \lambda_{\text{transfer}} \mathcal{L}_{\text{transfer}}, \tag{14}$$

where λ_{post} and $\lambda_{\text{transfer}}$ control privileged policy improvement and posterior-to-prior transfer, respectively. This optimization allows the prior and posterior to co-evolve: the posterior uses relational evidence to explore retrieval-relevant reasoning, the prior improves directly under inference-compatible rewards, and importance weighting selectively internalizes privileged discoveries that are both useful and reproducible.

4 Experiments

4.1 Experimental Setup

Dataset and metrics. We build on the datasets released by UME-R1, comprising 1.46M SFT examples and 11,136 RL examples curated from MMEB-Train (Jiang et al., 2025). For dual-mode initialization, we adapt the SFT dataset. We synthesize posterior traces and retain approximately 676K examples after label-prediction verification for posterior-mode supervision, while preserving the original traces of the remaining 787K examples annotated by GLM-4.1V-thinking (Hong et al., 2025) for prior-mode supervision. Co-evolution uses the released RL dataset. We evaluate PILOT on the full MMEB-V2 benchmark (Meng et al., 2026), which comprises 78 tasks spanning image, video, and visual-document retrieval. Following the standard protocol, we report Hit@1 for image and video tasks and nDCG@5 (Järvelin & Kekäläinen, 2002) for visual-document tasks. In the analysis experiment, we additionally report Pass@K that measures the proportion of queries for which at least one of K sampled reasoning traces produces an embedding that ranks an official positive candidate first.

Baselines. Our primary baseline, UME-R1, shares PILOT's underlying training data, Qwen2-VL (Wang et al., 2024) backbones, and unified embedding setup. DUME is its encoding-only variant without reasoning generation. TTE$_s$ and MMEmb-R1 (reasoning-enhanced baselines) differ in data curation and reasoning supervision, with TTE$_s$ using separate reasoner and embedder models. We also compare against GME (Zhang et al., 2024a), VLM2Vec (Jiang et al., 2025), VLM2Vec-V2 (Meng et al., 2026), LamRA (Liu et al., 2025), CAFe (Yu et al., 2025), and RzenEmbed (Jian et al., 2025), alongside the document-specialized multi-vector retriever ColPali-V1.3 (Faysse et al., 2025) and proprietary Seed-1.6-Embedding.

Table 1: Performance comparison across image, video, and visual-document (VisDoc) retrieval benchmarks. Shaded rows denote PILOT. Blue values in parentheses indicate absolute gains over UME-R1 at the same backbone size.

Model	Training		Image					Video					VisDoc					All
	# of Pairs	# of Annotators	CLS	QA	RET	GD	Avg.	CLS	QA	RET	MRET	Avg.	VDRv1	VDRv2	VR	OOD	Avg.	
# of Datasets			10	10	12	4	36	5	5	5	3	18	10	4	6	4	24	78
Closed-Source Models																		
Seed-1.6-embedding	NR	NR	76.1	74.0	77.9	91.3	77.8	55.0	60.9	51.3	53.5	55.3	85.5	56.6	84.7	43.1	73.4	71.3
Small Backbones (2–3B)																		
ColPali-V1.3	127K	—	40.3	11.5	48.1	40.3	34.9	26.7	37.8	21.6	25.5	28.2	83.6	52.0	81.1	43.1	71.0	44.4
VLM2Vec	1.03M*	—	58.7	49.3	65.0	72.9	59.7	33.4	30.5	20.6	33.0	29.0	49.8	13.5	51.8	33.5	41.6	47.0
DUME	1.46M	—	59.3	55.0	66.3	78.0	62.5	37.7	46.6	17.1	30.0	33.2	67.6	43.3	47.1	33.8	52.8	52.7
GME	8M	—	54.4	29.9	66.9	55.5	51.9	34.9	42.0	25.6	32.4	33.9	86.1	54.0	82.5	43.1	72.7	54.1
VLM2Vec-V2	2M*	—	62.9	56.3	69.5	77.3	64.9	39.3	34.3	28.8	38.5	34.9	75.5	44.9	79.4	39.4	65.4	58.0
RzenEmbed	≥5M	—	68.5	66.3	74.5	90.3	72.3	50.4	49.7	46.6	38.9	47.3	87.1	55.1	87.2	43.4	74.5	67.2
TTE$_s^\dagger$	2M*	1	67.9	66.6	70.2	84.1	70.1	47.3	49.1	34.4	33.2	41.9	77.5	53.2	83.2	41.1	68.8	63.1
MMEmb-R1	1.2M	3	64.5	68.1	70.0	88.9	70.0	56.3	52.8	47.6	42.5	50.6	74.1	56.0	78.9	48.3	68.0	65.0
UME-R1	1.46M	1	64.4	62.0	67.8	75.7	66.1	44.3	50.0	32.9	39.3	41.8	72.4	46.0	79.2	37.2	63.8	60.1
PILOT	1.46M	1	65.2	63.8	69.0	78.1	67.5 (+1.4)	47.7	58.3	34.8	40.8	45.9 (+4.1)	80.6	51.9	82.5	42.1	69.9 (+6.1)	63.3 (+3.2)
Large Backbones (∼ 7B)																		
LamRA	1.38M*	—	59.2	26.5	70.0	62.7	54.1	39.3	42.6	24.3	34.6	35.2	22.0	11.5	37.4	21.0	23.9	40.4
GME	8M	—	57.7	34.7	71.2	59.3	56.0	37.4	50.4	28.4	38.2	38.6	89.4	55.6	85.0	44.4	75.2	57.8
VLM2Vec	1.03M*	—	62.7	56.9	69.4	82.2	65.5	39.1	30.0	29.0	40.6	34.0	56.9	9.4	59.1	38.1	46.4	52.3
DUME	1.46M	—	64.2	57.0	70.8	81.8	66.4	32.9	47.4	8.6	28.0	29.4	67.1	35.2	82.6	34.9	60.3	55.9
CAFe	NR	—	63.6	61.7	69.1	87.6	67.6	35.8	58.7	34.4	39.5	42.4	70.7	49.6	79.5	38.1	63.9	60.6
VLM2Vec-V2	2M*	—	65.7	61.5	70.0	85.2	68.1	45.9	33.9	27.6	39.3	36.4	78.8	52.6	82.7	42.1	69.3	61.2
RzenEmbed	≥5M	—	70.6	71.7	78.5	92.1	75.9	58.8	63.5	51.0	45.5	55.7	89.7	60.7	88.7	44.4	77.1	71.6
TTE$_s^\dagger$	2M*	1	69.7	72.4	74.0	90.6	74.2	49.1	60.6	36.4	37.2	46.8	84.1	62.7	91.9	47.6	76.4	68.6
MMEmb-R1	1.2M	3	68.1	71.8	77.7	87.1	74.4	60.2	56.0	51.2	41.1	53.3	79.6	65.3	84.7	58.0	74.9	69.7
UME-R1	1.46M	1	66.4	69.1	71.8	84.1	70.9	48.2	60.4	38.5	38.9	47.3	75.8	50.0	83.7	37.5	67.1	64.3
PILOT	1.46M	1	68.7	70.2	73.2	87.2	72.7 (+1.8)	50.1	62.8	40.2	41.4	49.4 (+2.1)	83.6	60.8	85.1	43.3	73.5 (+6.4)	67.6 (+3.3)

† TTE$_s$ uses separate reasoner and embedder models. — indicates no trace supervision, NR an unknown total, and * an estimate from released data. Within each size group, **encoding-only** models precede the internal rule and **reasoning-enhanced** models follow it.

Implementation details. We use Qwen2-VL-2B/7B-Instruct backbones and GLM-4.1V-thinking for posterior trace synthesis. Initialization runs for 5,000 steps with a global batch size of 1,024 via gradient accumulation, a learning rate of 5×10^{-5}, and a maximum sequence length of 12,288 tokens. Unless otherwise specified, co-evolution samples 8 trajectories per input per mode, with a global batch size of 64 and a learning rate of 10^{-6}. For efficient evaluation, we modify vLLM (Kwon et al., 2023) to return the normalized hidden state at a designated special token after generation. Additional implementation details are provided in Appendix A.1.

4.2 Main Results

Table 1 shows that PILOT outperforms UME-R1 across all 12 task groups at both scales, reaching overall scores of 63.3 (2B) and 67.6 (7B) using only the inference-time prior. Tasks in the VisDoc domain benefits most, with gains of 6.1 and 6.4 points. Our 2B model even outperforms UME-R1-7B in this modality (69.9 vs 67.1), suggesting that improved training can partly compensate for a smaller backbone. These gains over an already reasoning-enhanced baseline highlight the importance of how reasoning is learned for embedding. Against encoding-only methods, PILOT surpasses VLM2Vec-V2 by 5.3 and 6.4 points overall and DUME by 10.6 and 11.7 points at the two scales. GME nevertheless achieves higher score in the VisDoc domain, while RzenEmbed performs better overall, indicating that the effectiveness of the complete training recipe remains important. Among reasoning-enhanced methods, PILOT performs comparably to TTE$_s$ at 2B (63.3 vs 63.1) and achieves a higher video average at 7B (49.4 vs 46.8), although TTE$_s$ and MMEmb-R1 score higher overall at 7B. These results reveal task-dependent strengths across different supervision and inference configurations, while the consistent gains over UME-R1 support using training-time relational evidence to improve inference-time embeddings.

4.3 Ablation Study

Table 2 reports ablations using Qwen2-VL-2B-Instruct, with results averaged over three random seeds. Dual-mode SFT improves the overall score from 59.0 to 60.3, with gains across all three modalities. Direct distillation, which pairs input-only prior prompts with annotated posterior traces

Table 2: Component ablation on MMEB-V2. Each row specifies a complete training configuration. All variants are evaluated using only the inference-time prior.

	Training configuration					Performance			
Variant	**SFT initialization**	**Prior GRPO**	**Posterior during RL**	**IW prior fitting**	**Rollouts/query (prior + posterior)**	**Image**	**Video**	**VisDoc**	**All**
PILOT	Dual-mode	✓	GRPO-updated	✓	$8+8$	67.5	45.9	69.9	63.3
Joint GRPO	Dual-mode	✓	GRPO-updated	×	$8+8$	66.8	45.1	68.1	62.2 (−1.1)
Frozen-posterior variant	Dual-mode	✓	Frozen	×	$8+0$	67.1	44.8	66.1	61.6 (−1.7)
Prior GRPO only	Dual-mode	✓	Not used	×	$16+0$	66.5	43.4	67.2	61.4 (−1.9)
Dual-mode SFT	Dual-mode	—	—	—	—	65.7	42.1	65.9	60.3 (−3.0)
Direct Distillation	Prior	—	—	—	—	64.7	41.6	64.8	59.4 (−3.9)
Prior SFT	Prior	—	—	—	—	65.3	41.0	63.2	59.0 (−4.3)

✓: enabled; ×: disabled; —: not applicable. IW denotes importance weighting. Rollouts/query reports the number of trajectories sampled per query during RL, written as prior + posterior.

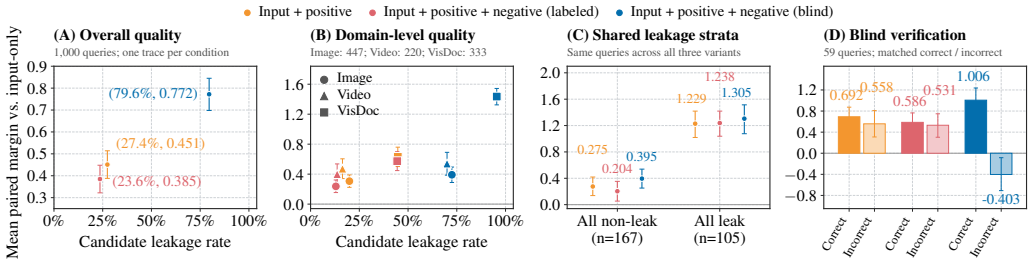

Figure 3: **Reasoning quality, candidate leakage, and verification under different synthesis prompts.** (A–B) Overall and domain-level judged retrieval usefulness versus candidate leakage. (C) Comparison on shared query subsets with no leakage flagged under any variant or leakage flagged under every variant. (D) Comparison conditioned on blind-prediction correctness, with matched sample indices across conditions.

as SFT targets, achieves only 59.4. Its limited gain is consistent with a conditioning mismatch, as posterior traces may depend on candidate information unavailable to the prior. Prior-only GRPO further raises the score to 61.4, confirming benefits from both privileged initialization and direct optimization of the deployed reasoning mode. Adding a frozen posterior yields only 0.2 points overall and reduces visual-document performance from 67.2 to 66.1. Joint GRPO instead reaches 62.2 overall and 68.1 on visual documents, outperforming prior-only GRPO across all modalities. This supports adapting the posterior during training rather than relying on fixed privileged guidance. Importance-weighted prior fitting adds another 1.1 points overall, improving image, video, and visual-document scores by 0.7, 0.8, and 1.8 points, respectively. Full PILOT exceeds prior-only GRPO by 1.9 points, showing that explicit transfer complements joint policy optimization.

4.4 Further Analysis

To better understand how PILOT uses privileged information, we investigate three questions: **(RQ1)** whether blind synthesis and label-prediction verification improve the quality of reasoning supervision; **(RQ2)** whether the posterior improves exploration of useful reasoning trajectories; and **(RQ3)** whether posterior discoveries transfer to the inference-time prior. We conduct these analyses on 1,000 examples sampled approximately equally across all MMEB-V2 subsets, each comprising a query and its ground-truth positive candidate. More details can be seen in Appendix A.4.

RQ1. We use the annotator to generate query-side reasoning under four conditions: input only, input with a positive candidate, input with a labeled positive–negative pair, and input with a blind positive–negative pair (our blind contrastive posterior). GPT-5.6 sol is used to assign retrieval-usefulness margins in $[-2, 2]$ (higher is better) and separately flag candidate information unsupported by the query (Appendix A.5). Figure 3(A–B) shows that blind synthesis achieves the highest mean margin (0.772), particularly on visual documents, but also the highest leakage rate (79.6%). It retains its advantage on the shared non-leakage subset (C), scoring 0.395 versus 0.275 for positive-only and 0.204 for labeled-pair prompting. Matched comparisons on 59 queries (D) yield margins of

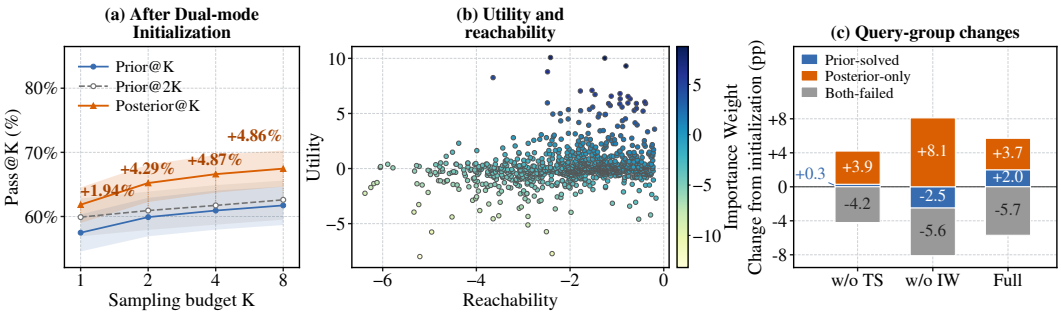

Figure 4: Posterior exploration and transfer. (a) Pass@K after dual-mode initialization. (b) Utility–reachability trade-off, colored by transfer score. (c) Changes in prior-solved, posterior-only, and both-failed query proportions relative to dual-mode initialization comparing the full method and ablations without transfer supervision (w/o TS) or importance weighting (w/o IW).

1.006 for correct blind predictions versus -0.403 for incorrect ones, with much smaller differences in the controls. Blind synthesis thus elicits useful reasoning, while verification provides a quality filter. Neither guarantees that the resulting traces are reproducible from the input alone, motivating transfer that also accounts for prior reachability.

RQ2. We assess posterior exploration at the dual-mode SFT checkpoint by sampling 16 traces per query under each prior and blind posterior prompt, then replay all traces under the prior prompt and evaluate their embeddings against the same candidate pool. Figure 6 (a) shows that Posterior@K outperforms both Prior@K and Prior@$2K$ across all tested budgets. As K increases from 1 to 8, posterior Pass@K rises from 61.8% to 67.5%, exceeding Prior@$2K$ by 1.9–4.9 percentage points. Thus, conditioning exploration on relational evidence yields gains that doubling input-only sampling does not recover, motivating its use to guide reasoning exploration during training.

RQ3. We investigate this question from two complementary perspectives: how transfer prioritizes posterior trajectories and how transfer supervision and importance weighting affect retrieval coverage. Figure 6(b) plots one valid posterior trajectory per query by its utility and reachability components, colored by the un-normalized importance weight. The distribution illustrates the trade-off encoded by our weighting: the retrieval utility increases the transfer priority, whereas a large reachability penalty can offset this benefit. Thus, high-reward posterior trajectories are not automatically assigned high transfer scores. Panel (c) compares Full PILOT with variants removing transfer supervision (w/o TS) or replacing importance weighting with uniform weighting (w/o IW). Using eight traces per query and mode, we count success if any trace retrieves a relevant candidate at rank 1 and partition queries into prior-solved, posterior-only, and both-failed groups. Compared with w/o IW, Full raises prior-solved coverage from 59.3% to 63.8% and lowers posterior-only coverage from 15.8% to 11.4%, with nearly unchanged coverage by either mode (75.1% vs 75.2%). Against w/o TS, Full improves prior-solved coverage by 1.7 points and reduces both-failed queries by 1.5 points. These aggregate results suggest that selective transfer helps make retrieval success accessible to the inference-time prior, whereas greater posterior-only coverage need not improve input-only retrieval. More details of the experiment can be seen in Appendix A.6.

5 Conclusion

We presented PILOT, a latent-variable framework that uses training-time relational evidence to improve reasoning for multimodal embeddings while retaining input-only inference. It outperforms main baselines across all 12 MMEB-V2 task groups at both 2B and 7B scales. Our findings suggest that retrieval supervision should inform which reasoning trajectories are explored, alongside evaluating their embedding utility. Yet increased candidate leakage and limited gains from direct distillation highlight the need for selective transfer. The benefits of posterior adaptation and importance-weighted prior fitting point to a practical training strategy that jointly improves exploration and transfers its discoveries according to retrieval utility and prior reachability. For settings with relational evidence available only during training, this provides a basis for developing better exploration policies and transfer objectives without requiring candidate context at inference.

AI USE STATEMENT

In this work, we used generative AI tools to polish manuscript drafts and assist with experimental code implementation. We have reviewed all AI-assisted work. Specifically, we manually check the generated manuscript paragraphs for potential semantic deviation or grammar errors. LLM-generated experimental code was verified by the author manually and tested to ensure the correctness before being used in the practical experiments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our experiments use publicly available datasets and backbone models. We describe the training data, model configurations, optimization settings, and evaluation protocols in the experimental section, with additional implementation details in Appendix A.1. Synthesis prompts and the reasoning evaluation protocol are provided in Appendix A.2. Appendix A.3 details the derivation from the importance-weighted lower bound to the three gradient components instantiated in prior–posterior co-evolution. About the further analysis experiment, Appendix A.4 documents probe dataset construction. More detailed experiments about the model-based evaluation of the synthesized posterior reasoning trace can be seen in Appendix A.5. More experiments about the effect of the importance weight are presented in Appendix A.6. All the quantitative experiment results are averaged over three random seeds. We will release our training and evaluation code, configurations, and synthesized reasoning data to facilitate reproduction.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

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.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Xuanming Cui, Jianpeng Cheng, Hong-You Chen, Satya Narayan Shukla, Abhijeet Awasthi, Xichen Pan, Chaitanya Ahuja, Shlok Kumar Mishra, Taipeng Tian, Qi Guo, Ser-Nam Lim, Aashu Singh, and Xiangjun Fan. Think then embed: Generative context improves multimodal embedding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AKXXMK5YTI.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, CELINE HUDELOT, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ogjBpZ8uSi.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

Tiancheng Gu, Kaicheng Yang, Ziyong Feng, Xingjun Wang, Yanzhao Zhang, Dingkun Long, Yingda Chen, Weidong Cai, and Jiankang Deng. Breaking the modality barrier: Universal embedding learning with multimodal llms. In *Proceedings of the 33rd ACM International*

Conference on Multimedia, MM '25, pp. 2860–2869, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3754845. URL https://doi.org/10.1145/3746027.3754845.

Chenwei He, Xiangzhao Hao, Tianyu Yang, Yuxiang Ma, Yuheng Jia, Lingxiang Wu, Chaoyang Zhao, Haiyun Guo, and Jinqiao Wang. Plume: Latent reasoning based universal multimodal embedding, 2026. URL https://arxiv.org/abs/2604.02073.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002. ISSN 1046-8188. doi: 10.1145/582415.582418. URL https://doi.org/10.1145/582415.582418.

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

Weijian Jian, Yajun Zhang, Dawei Liang, Chunyu Xie, Yixiao He, Dawei Leng, and Yuhui Yin. Rzenembed: Towards comprehensive multimodal retrieval. *arXiv preprint arXiv:2510.27350*, 2025.

Haonan Jiang, Yuji Wang, Yongjie Zhu, Xin Lu, Wenyu Qin, Meng Wang, Pengfei Wan, and Yansong Tang. Embed-rl: Reinforcement learning for reasoning-driven multimodal embeddings. *arXiv preprint arXiv:2602.13823*, 2026.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2vec: Training vision-language models for massive multimodal embedding tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TE0KOzWYAF.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. LLaVE: Large language and vision embedding models with hardness-weighted contrastive learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 13721–13735, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.740. URL https://aclanthology.org/2025.findings-emnlp.740/.

Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. Ume-r1: Exploring reasoning-driven generative multimodal embeddings. In *International Conference on Learning Representations*, volume 2026, pp. 115779–115806, 2026.

Zihan Liang, Yufei Ma, Ben Chen, Zhipeng Qian, Xuxin Zhang, Huangyu Dai, and Lingtao Mao. Search-e1: Self-distillation drives self-evolution in search-augmented reasoning. *arXiv preprint arXiv:2605.22511*, 2026.

Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. MM-EMBED: UNIVERSAL MULTIMODAL RETRIEVAL WITH MULTIMODAL LLMS. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=i45NQb2iKO.

Tianqianjin Lin, Xi Zhao, Xingyao Zhang, Rujiao Long, Yi Xu, Zhuoren Jiang, Wenbo Su, and Bo Zheng. RAVR: Reference-answer-guided variational reasoning for large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the*

Association for Computational Linguistics: ACL 2026, pp. 18757–18775, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.936. URL https://aclanthology.org/2026.findings-acl.936/.

Yikun Liu, Yajie Zhang, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiangchao Yao, Yanfeng Wang, and Weidi Xie. Lamra: Large multimodal model as your advanced retrieval assistant. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4015–4025. IEEE, 2025.

Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Raghuveer Thirukovalluru, Xuan Zhang, Zeyuan Chen, Ran Xu, Caiming Xiong, Yingbo Zhou, Wenhu Chen, and Semih Yavuz. VLM2vec-v2: Advancing multimodal embedding for videos, images, and visual documents. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=TpU38jbKIJ.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ebZcMQImhG.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Jiashuo Sun, Shixuan Liu, Zhaochen Su, Xianrui Zhong, Pengcheng Jiang, Bowen Jin, Peiran Li, Weijia Shi, and Jiawei Han. GRACE: Generative representation learning via contrastive policy optimization. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=hs9lwjH1bJ.

Zelong Sun, Jun Wang, Kaicheng Yang, Tiancheng Gu, Ziyong Feng, and Zhiwu Lu. Learning from failures: Retrieval-centric cot via hard negatives for unified multimodal retrieval. *arXiv preprint arXiv:2608.06060*, 2026b.

Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: similarity control and knowledge transfer. *J. Mach. Learn. Res.*, 16(1):2023–2049, January 2015. ISSN 1532-4435.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Yuchi Wang, Haiyang Yu, Weikang Bian, Jiefeng Long, Xiao Liang, Chao Feng, and Hongsheng Li. Mmemb-r1: Reasoning-enhanced multimodal embedding with pair-aware selection and adaptive control. *arXiv preprint arXiv:2604.06156*, 2026.

Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, pp. 1437–1445, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368896. doi: 10.1145/3343031.3351034. URL https://doi.org/10.1145/3343031.3351034.

Xueru Wen, Jie Lou, Yanjiang Liu, Hongyu Lin, Ben He, Xianpei Han, Le Sun, Yaojie Lu, and Debing Zhang. Coupled variational reinforcement learning for language model general reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YgQ3MV7OtR.

Dingkang Yang, Haopeng Kuang, Shuai Huang, and Lihua Zhang. Learning modality-specific and -agnostic representations for asynchronous multimodal language sequences. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 1708–1717, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3547755. URL https://doi.org/10.1145/3503161.3547755.

Hao Yu, Zhuokai Zhao, Shen Yan, Lukasz Korycki, Jianyu Wang, Baosheng He, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, and Hanchao Yu. Cafe: Unifying representation and generation with contrastive-autoregressive finetuning. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 6345–6356, 2025. doi: 10.1109/ICCVW69036.2025.00659.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18123–18133, 2022.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Gme: Improving universal multimodal retrieval by multimodal llms. *arXiv preprint arXiv:2412.16855*, 2024a.

Zhuosheng Zhang, Aston Zhang, Mu Li, hai zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*, 2024b. ISSN 2835-8856. URL https://openreview.net/forum?id=y1pPWFVfvR.

Xiangxin Zhou, Zichen Liu, Haonan Wang, Chao Du, Min Lin, Chongxuan Li, Liang Wang, and Tianyu Pang. Variational reasoning for language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fGGcovg6oW.

A Appendix

A.1 Implementation Details

A.1.1 Training Settings

Supervised fine-tuning. During SFT initialization, we freeze the vision encoder while updating the visual merger and language model without LoRA. The training objective combines autoregressive losses on the query and positive-candidate responses with symmetric contrastive losses on the normalized representations at the generative and discriminative embedding tokens. Both contrastive losses use cross-device in-batch negatives and a temperature of 1/30, and all four loss terms are summed with unit coefficients. We train for 5,000 optimizer steps using AdamW with a learning rate of 5×10^{-5}, zero weight decay, a global batch size of 1024 via gradient accumulation and $\epsilon = 10^{-6}$. We use cosine learning-rate decay with a 3% warmup and clip the gradient norm at 0.5. The per-device batch size is two query–candidate pairs, with six gradient-accumulation steps. Training uses a maximum sequence length of 12,288 tokens, bfloat16 precision, gradient checkpointing, and DeepSpeed ZeRO-3. We set both the training and data seeds to 42 and save checkpoints every 500 steps. The fine-tuning of the 2B and 7B model is conducted on 64 NVIDIA A100 80 GB GPUs.

Coevolution. We train for one epoch, comprising 192 rollout iterations, with a global batch size of 64 and eight sampled responses per input under each prompt mode. Responses are generated at temperature 1.0 without top-k or nucleus truncation, with maximum prompt and response lengths of 7,168 and 1,024 tokens, respectively. We use AdamW with a constant learning rate of 10^{-6}, momentum coefficients (0.9,0.999), weight decay 0.01, and gradient clipping at 1.0. GRPO uses within-group standardized advantages, a policy-ratio clipping threshold of 0.2, and one optimization epoch per rollout batch. The prior GRPO, posterior GRPO, and weighted SFT objectives are normalized separately and combined with coefficients 1.0, 1.0, and 0.3, respectively. For both prior and

posterior branch, we set $\lambda_{\text{fmt}} = 0.02$. We apply KL regularization against the initialization checkpoint with coefficient 0.05 and no entropy bonus. Retrieval rewards use an InfoNCE temperature of 0.2.

$$R_{\text{ret}}(\mathbf{e}_\theta(x, z_k), \mathcal{R}) = \log \frac{\sum_{m=1}^{M} \exp(\text{sim}(\mathbf{e}_\theta(x, z_k), \mathbf{e}_\theta(x^+, z_m))/T)}{\sum_{x^- \in \mathcal{X}^-} \sum_{m=1}^{M} \exp(\text{sim}(\mathbf{e}_\theta(x, z_k), \mathbf{e}_\theta(x^-, z_m))/T)}, \tag{15}$$

Here and M is the number of rollouts per candidate. To induce the conditional likelihood, we set $\beta = -10$. The similarity function $\text{sim}(\mathbf{e}_\theta(x, z_k), \mathbf{e}_\theta(x^+, z_m)) = \mathbf{e}_\theta(x, z_k)^\top \mathbf{e}_\theta(x^+, z_m)$ computes cosine similarity, with temperature $T = 0.2$. The reward is the log-probability assigned to the positive set; higher values indicate better retrieval performance. Training runs on 16 NVIDIA A100 80GB GPUs for the 2B backbone and 32 NVIDIA A100 80 GB GPU for the 7B backbone using FSDP2, bfloat16 mixed precision, gradient checkpointing, and vLLM-based sampling.

A.1.2 Evaluation Settings

We evaluate on MMEB-V2 using the subset-specific task configurations and candidate pools provided by the its official repo. Both queries and candidates are independently encoded in generative mode using only their original inputs and task instructions, without additional positive or negative examples. For each input, we generate a single reasoning trace using greedy decoding (temperature 0) with a maximum of 1,024 new tokens. Our customized vLLM backend directly returns the L_2-normalized final-layer hidden state at the <gen_emb> token as the embedding. Candidates are ranked by cosine similarity against either the complete subset-level candidate pool or the query-specific candidate set, as specified by the task configuration. We use an evaluation batch size of 96, image-resizing bounds of 784–2,359,296 pixels, and eight sampled frames for video inputs according to the task configuration. Retrieval metrics are computed separately for each subset using its ground-truth relevance annotations.

A.1.3 Posterior Reasoning Trace Synthesis

Posterior reasoning synthesis. We use GLM-4.1V-9B-Thinking as the annotation model, served through vLLM with a maximum context length of 32,768 tokens. We sample annotations with temperature 1.0, top-p 0.95, top-k 20, and a presence penalty of 1.5. For each query, the annotator receives its positive candidate and a randomly sampled negative candidate from the same source subset; symmetrically, each candidate is accompanied by a positive and a negative query. Auxiliary items are presented in randomized order without revealing their relevance labels. The annotator identifies the positive item and generates a content-grounded reasoning trace and concise summary, with instructions to avoid explicit option references and details unsupported by the focal input. We retain a posterior training pair only when positive-item identification is correct on both sides and the outputs pass format validation. Predicted option labels are excluded from the supervised responses, while reasoning and summaries are serialized into the training format ending with <gen_emb>. The final SFT examples use a simplified, role-independent posterior prompt rather than the detailed annotation instructions. Table 3 summarizes the dataset used for dual-mode initialization. It contains 1,463,360 query–candidate pairs, comprising 787,202 prior-mode pairs (53.79%) and 676,158 posterior-mode pairs (46.21%). Each pair provides reasoning supervision for both the query and its positive candidate. Prior-mode prompts contain only the focal input and instructions, whereas posterior-mode prompts additionally include positive and negative auxiliary examples.

Table 3: Composition of the dual-mode SFT dataset. Each pair contains a query and its positive candidate, with reasoning supervision on both sides.

Mode	Number of pairs	Proportion
Prior	787,202	53.79%
Posterior	676,158	46.21%
Total	1,463,360	100.00%

A.2 PROMPT TEMPLATES

We distinguish prompts consumed by the embedding model during SFT or RL from prompts sent to the expert annotation model. The former are intentionally short, whereas the latter contain detailed grounding and leakage-control instructions. In the templates below, angle-bracketed uppercase strings such as <MAIN_TEXT> and <MAIN_VISUAL> are exposition-only placeholders. Literal model tokens, including <disc_emb>, <think>, <answer>, and <gen_emb>, are shown without substitution. Image and video objects are inserted at the positions marked by the corresponding visual placeholders.

A.2.1 EMBEDDING-MODEL TRAINING PROMPTS

The prior receives only the focal input. The same template is used on the query and candidate sides, with the appropriate text and visual content substituted for the focal input.

Prior prompt used for embedding-model training

```
<MAIN_VISUAL>
<MAIN_TEXT>
<disc_emb>
Represent the above input text, images, videos, or any combination of
    the three as embeddings.
First output the thinking process in <think> </think> tags and then
    summarize the entire input in a word or sentence. Finally, use the
    <gen_emb> tag to represent the entire input.
```

The posterior receives the same focal input followed by one positive and one negative in randomized order. The labels do not expose which item is positive. For query-side training, the auxiliary items are candidate inputs; for candidate-side training, they are queries. Multimodal objects are interleaved at the indicated positions.

Blind posterior prompt used for embedding-model training

```
<MAIN_VISUAL>
<MAIN_TEXT>
<disc_emb>
Represent the above input text, images, videos, or any combination of
    the three as embeddings. A few auxiliary items are provided below
    in randomized order (each may also include text, images, or videos)
     -- one is a true match for the above input and the other is not.
    Infer which is which during your reasoning to identify the
    discriminative features. The embedding must represent the above
    input alone.

Auxiliary item [A]:
<AUXILIARY_A_VISUAL>
<AUXILIARY_A_TEXT>

Auxiliary item [B]:
<AUXILIARY_B_VISUAL>
<AUXILIARY_B_TEXT>

First output the thinking process in <think> </think> tags and then
    summarize the entire input in a word or sentence. Finally, use the
    <gen_emb> tag to represent the entire input.
Do not copy exact answer spans or unique details from the auxiliary
    items. Do not treat the auxiliary items as the object to be
    represented.
```

A.2.2 EXPERT PROMPTS FOR SFT ANNOTATION

The query-side blind posterior annotation prompt used for the multimodal SFT data is formed by concatenating the following input wrapper with the query-side instruction. Candidate labels are randomized before the request is sent to the expert. Samples for which the returned choice is incorrect are discarded.

Expert blind posterior annotation wrapper: query side

```
You are generating training data for a retrieval embedding model. You
    will be shown a QUERY and a few candidate documents labeled with [A
    ] [B] [C] ... sequentially. One is a true positive match for the
    query; the other is not. Infer which is the true positive and write
     reasoning and an answer ABOUT THE QUERY.
The "choice" field is used to filter training data. "reasoning" and "
    answer" are used as training targets for the student model.

MULTIMODAL INPUT
Each side (query or candidate) may carry an image or a video,
    interleaved at its position in this message. The text and visual
    together constitute one side.
- When the text reads "describe the content of the image" / "describe
    the content of the video" or a similar placeholder, treat the
    accompanying visual as the primary content of that side.
- When the text is substantive, the visual -- if present -- is
    supplementary context.
- When no visual is present, the text is the entire content.

INPUT
Query:
<QUERY_TEXT_AND_VISUAL>

Candidate [A]:
<CANDIDATE_A_TEXT_AND_VISUAL>

Candidate [B]:
<CANDIDATE_B_TEXT_AND_VISUAL>
```

Expert blind posterior annotation instruction: query side

```
CHOICE FIELD
- "choice" identifies the candidate that is the true positive match
    for the query.
- Output exactly "A" or "B" or "C" or ... -- single letter, uppercase,
     no other text. Commit to one; do not output "neither" or "
    uncertain."
- A true positive satisfies the query's information need. If both seem
     plausible, pick the one with stronger and more specific alignment.
- Choice must be consistent with your internal reasoning, but
    reasoning/answer must NOT reveal which candidate you saw.

REASONING REQUIREMENTS
- Be content-rich: derive answers, compute values, identify entities,
    describe visible content concretely. Richer reasoning yields a
    stronger embedding.
- If the query has a derivable answer (factual, numerical, descriptive
    , identification), perform the derivation and state the result.
- The discriminative features that determined your choice should
    appear as positive specifications of what a true match must have --
     NOT as comparisons between candidates.
- Overlap with either candidate's content is acceptable when
    independently derivable from the query alone. The leakage
```

```
    constraint applies only to content unreachable without seeing the
    candidates.
- Summarize your reasoning in the answer field with one or two
    sentence.

LEAKAGE -- your "reasoning" and "answer" must not:
- include information knowable only from seeing either candidate (
    textual or visual);
- describe, paraphrase, quote, or mirror either candidate's wording or
     visual specifics;
- use comparative phrasing ("the better match," "of the two," "rather
    than," "as opposed to") or reference multiplicity ("among the
    options," "either of them," "between them");
- name the focal's role or the auxiliary's role ("the query," "the
    user," "the candidate," "the match");
- introduce visual vocabulary when the query itself has no visual
    content;
- describe the query's own visual at greater specificity than it
    observably supports.

A reader given only "reasoning" and "answer" should be unable to tell
    whether you saw zero, one, or many candidates, and unable to infer
    visual features of any candidate.

STYLE
STYLE (for "reasoning" and "answer")
- Vary your opening sentence across examples.

OUTPUT
Return a single JSON object, nothing else:
{
  "choice": "A" or "B" or ...,
  "reasoning": "...",
  "answer": "..."
}
```

A.2.3 UPDATED CANDIDATE-SIDE EXPERT ANNOTATION PROMPTS

The distinction between the query and candidate side is substantive: a text query should not acquire an unseen page's answer, whereas a candidate page should preserve its own visible names, facts, numbers, and evidence relationships. Auxiliary queries may redirect inspection, but must not supply absent content or reduce a general candidate representation to the answer to one particular query.

System prompt for candidate annotation

```
You generate supervised reasoning targets for a candidate-side
    multimodal retrieval embedding model. Produce an informative
    semantic representation of the CANDIDATE, preserving its content
    and important details.

Distinguish candidate evidence from auxiliary query content. Queries
    may suggest interpretations or details worth checking; retain those
     suggestions only when the candidate independently supports them.
    Query selection is an intermediate verification task, and its label
     does not establish facts about the candidate.

Be concrete and faithful. Preserve useful information even when it is
    unnecessary for distinguishing the supplied queries. Describe
    observable content and evidence relationships instead of generic
    statements about retrieval or relevance. The target represents the
    candidate, not a response tailored to an auxiliary query.
```

User-message wrapper for candidate annotation

```
Generate a retrieval-oriented reasoning trace and embedding summary
    for the CANDIDATE.

MULTIMODAL INPUT
The candidate may contain text, an image, a video, or a combination.
    Its text and visual content jointly define the candidate.
- When the candidate text is a placeholder instruction, inspect the
    accompanying visual as the primary content.
- When the candidate text is substantive, preserve its entities,
    relations, modifiers, and constraints. Use any candidate visual as
    additional evidence.
- When the candidate is text-only, represent its stated content
    without importing an auxiliary query's image, video, reference
    object, or requested transformation.
- When the candidate is a document page, inspect the supplied page
    itself. Distinguish visible evidence from content that might exist
    on an unseen page.

CANDIDATE-FIRST ANALYSIS
Before evaluating any auxiliary query, identify the candidate's main
    entities, attributes, actions, relations, subject matter, and
    explicit claims. Inspect all available candidate modalities. Keep
    track of what is directly supported, what is a supported
    interpretation, and what remains unknown.

INPUT
Candidate:
<CANDIDATE_TEXT_AND_VISUAL>

<OPTIONAL_AUXILIARY_QUERY_BLOCK>

END OF INPUT

ANNOTATION INSTRUCTIONS

<CONDITION_WORKFLOW>

<SHARED_CANDIDATE_TARGET_RULES>

<OUTPUT_SCHEMA>
```

Auxiliary-query block for blind candidate annotation

```
Query [A]:
<QUERY_A_TEXT_AND_VISUAL>

Query [B]:
<QUERY_B_TEXT_AND_VISUAL>
```

In the blind posterior, one query is positive for the focal candidate and the other is negative. Their order is randomized and relevance labels are not shown. The prompt does not assert that the negative is hard: the same wording supports either random or mined negatives. The returned choice identifies the positive query; correctness filtering, when applied, is a separate processing step.

Candidate blind posterior annotation workflow

```
BLIND VERIFICATION AND CANDIDATE SYNTHESIS
```

```
The goal is an informative semantic representation of the candidate.
    Query selection is an intermediate verification task. The candidate
     is a true positive for one of Query [A] and Query [B].

1. Understand the candidate before evaluating either query.
Identify its entities, attributes, actions, relations, explicit claims
    , and main subject matter. Inspect relevant candidate visuals.
    Preserve both what is supported and what remains unknown.

2. Evaluate A and B independently.
For each query, check which requirements the candidate satisfies,
    which it contradicts, and which cannot be verified from the
    supplied content. Prioritize explicit entity, relation, modifier,
    and temporal constraints over broad topical similarity. An unknown
    match is not confirmed support; an assumption in a query is not
    evidence about the candidate.
Respect the retrieval unit: a document cover can establish document
    identity without displaying the requested internal fact. Missing
    evidence in a preview is not automatically a contradiction, but it
    does not license claims about unseen content.

3. Investigate meaningful ambiguities.
When the queries suggest different interpretations, return to the
    candidate and inspect evidence that could resolve the difference.
    Use this comparison to discover overlooked candidate details. If a
    difference is already obvious or irrelevant to the candidate's
    content, do not force it into the final representation.

4. Make the best A/B choice.
Choose the query whose information need the candidate satisfies more
    specifically and completely. Do not revise the candidate's content
    to accommodate a query. If the available evidence cannot resolve
    the choice reliably, still return the best choice, but leave the
    unresolved interpretation out of the final reasoning. A choice does
     not make uncertain query assumptions true.

5. Synthesize the candidate representation.
Return to the candidate content identified before the comparison.
    Integrate query-suggested details only when independently supported
     by the candidate. Preserve useful connections between entities,
    attributes, actions, claims, and evidence.
The representation should remain informative for many plausible
    retrieval queries, not merely sufficient to choose between A and B.
     Do not include the query-selection process in reasoning or answer.
     Do not replace a broad candidate summary with a short answer to
    the selected query.
```

Candidate-grounding and output rules

```
CANDIDATE-GROUNDED TARGET
1. Preserve the full retrieval-relevant meaning of the candidate.
    Retain its important content even when some details are unnecessary
     for a local query decision.
2. Follow the condition-specific workflow internally; do not narrate
    the annotation workflow in reasoning or answer.
3. In reasoning, describe concrete supported entities, attributes,
    actions, relations, claims, and spatial or temporal cues. For
    documents, preserve visible evidence relationships, including the
    association of a value with its entity, metric, unit, or time
    period when readable.
4. Include useful discriminative details when supported, without
    forcing a fixed number of details or inventing specificity. The
```

```
    representation should remain informative for many plausible
    retrieval queries.
5. The answer is a concise semantic embedding summary of the candidate
    . Preserve its central content and important qualifiers; do not
    merely repeat the task instruction or answer an auxiliary query.
6. Preserve uncertainty locally: omit or qualify an uncertain detail
    while keeping other supported content specific and informative.

USEFUL OUTPUT CONTENT
Retain:
- Concrete details directly supported by the candidate text or visual,
     including relevant names, numbers, dates, and factual answers
    actually present in the candidate.
- Interpretations supported by identifiable evidence within the
    candidate. Connect an interpretation to the evidence supporting it.
- Meaningful relationships between depicted objects, actions, events,
    or stated facts.
- For document pages, readable headings, entities, claims, units,
    tables, charts, and their relationships when informative. Describe
    layout only when it helps identify or interpret content; do not
    replace semantic content with generic layout descriptions.
- For videos, observable action states, event order, and temporal
    relationships supported by the available frames. Do not invent
    motion or events between unobserved frames.

Exclude:
- Query-specific names, numbers, dates, locations, requested changes,
    or assumptions not independently established by the candidate.
- Claims about unseen pages, obscured text, hidden image regions,
    unobserved events, or unavailable audio.
- Broad task descriptions that replace concrete candidate content,
    such as 'retrieve relevant information' or 'focus on visual
    features'.
- Fine-grained visual, OCR, spatial, numerical, or temporal details
    that cannot be inspected reliably.

A question's premise does not become a candidate fact. Conversely, a
    factual answer visible in the candidate should not be suppressed
    merely because an auxiliary query asks for it. Preserve the fact in
     its candidate-supported context, not as a narrated response to the
     query. Do not compute a new value solely to satisfy an auxiliary
    request unless the calculation and all its inputs are independently
     justified by the candidate; clearly distinguish a supported
    derivation from an explicitly stated value.

FINAL CHECK
- Remove references to auxiliary queries, A/B labels, selection,
    rejection, and the annotation procedure from reasoning and answer.
    A separately required top-level choice field is exempt.
- Preserve comparisons, negations, exclusions, and relationships that
    are explicitly part of the candidate itself.
- For every factual clause, identify its support in the candidate.
    Remove clauses supported only by auxiliary content.
- Check that no important candidate entity, action, relation, claim,
    or qualifier disappeared while following the condition-specific
    workflow.
- The final reasoning and answer must stand on their own as an
    analysis of the candidate after every auxiliary query is removed.
```

Candidate blind posterior output schema

```
OUTPUT
Return one JSON object and nothing else. Set choice to exactly "A" or
    "B", a single uppercase letter identifying the positive query:

{
  "choice": "A",
  "reasoning": "...",
  "answer": "..."
}
```

A.2.4 DIAGNOSTIC ANNOTATION PROMPT-VARIANT EXPERIMENT

The four conditions are input only, input + positive, input + positive + negative (labeled), and input + positive + negative (blind). All four include the query input. The negative is sampled from the same subset rather than selected by hard-negative mining. In the blind condition, the positive and negative are assigned randomized A/B labels without revealing their ground-truth relevance.

All conditions share the system prompt, query-first introduction, task adapter, and target rules below. Only the auxiliary-input blocks, condition-specific workflow, and requirement for a separate choice field differ. The text is transcribed from the generation code, not rewritten for presentation. The uppercase angle-bracketed placeholders indicate inserted content; bracketed condition headings in the candidate-block box are explanatory and are not sent to the expert. The query and candidates can each contain text and visual inputs. The explicit `END OF INPUT` boundary separates the final input from `ANNOTATION INSTRUCTIONS`, preventing candidate text from running into the task-adapter heading.

Candidate blocks for the four diagnostic conditions

```
[INPUT ONLY]
No candidate block is included.

[INPUT + POSITIVE]
Positive candidate:
<POSITIVE_TEXT_AND_VISUAL>

[INPUT + POSITIVE + NEGATIVE (LABELED)]
Positive candidate:
<POSITIVE_TEXT_AND_VISUAL>

Negative candidate:
<NEGATIVE_TEXT_AND_VISUAL>

[INPUT + POSITIVE + NEGATIVE (BLIND)]
Candidate [A]:
<CANDIDATE_A_TEXT_AND_VISUAL>

Candidate [B]:
<CANDIDATE_B_TEXT_AND_VISUAL>
```

Diagnostic workflow: Input only

```
DIRECT QUERY WORKFLOW
Analyze the query directly. Cover its most retrieval-relevant
    observable content, constraints, and evidence relationships.
    Preserve what is known and what is requested without inventing
    missing facts.
```

Diagnostic workflow: Input + positive

```
POSITIVE-GUIDED WORKFLOW
First establish the query's meaning and important details
    independently.
Use the positive candidate to suggest interpretations, overlooked
    details, or evidence relationships worth checking. Return to the
    query to verify each suggestion. The positive label establishes a
    retrieval match, not independent support for every candidate fact.
Integrate verified suggestions with the query's other important
    content. If none can be verified, use a detailed direct analysis of
     the query. Do not narrow the query to the positive candidate's
    incidental content.
```

Diagnostic workflow: Input + positive + negative (labeled)

```
LABELED CONTRASTIVE WORKFLOW
First establish the query's meaning and important details
    independently.
Use the supplied positive and negative labels. Inspect their
    differences to identify meaningful ambiguities or overlooked query
    evidence. Return to the query to verify any proposed distinction.
If the negative differs only in broad topic, or its differences add
    nothing useful to understanding the query, do not force a contrast
    into the final representation. Continue with positive-guided
    inspection and direct query analysis.
Retain verified discriminative properties alongside the query's other
    important entities, actions, relations, and constraints. Do not
    reduce the representation to the minimum features needed to
    distinguish these two candidates. Express retained information as
    self-contained query content or requirements.
```

Diagnostic workflow: Input + positive + negative (blind)

```
BLIND VERIFICATION AND QUERY SYNTHESIS
The goal is an informative semantic representation of the query.
    Candidate selection is an intermediate verification task.

1. Understand the query before evaluating either candidate.
Identify its entities, attributes, actions, relations, and explicit
    constraints. Inspect relevant query visuals. Preserve both what is
    specified and what remains unknown.

2. Evaluate A and B independently.
For each candidate, check which query requirements it satisfies, which
     it contradicts, and which cannot be verified from the supplied
    content. Prioritize explicit entity, relation, modifier, and
    temporal constraints over broad topical similarity. An unknown
    match is not confirmed support; extra candidate detail is not
    evidence of correctness.
Respect the retrieval unit: a document cover can establish document
    identity without displaying the requested internal fact. Missing
    evidence in a preview is not automatically a contradiction.

3. Investigate meaningful ambiguities.
When the candidates suggest different interpretations, return to the
    query and inspect evidence that could resolve the difference. Use
    this comparison to discover overlooked query details. If a
    difference is already obvious or irrelevant to the query's meaning,
     do not force it into the final representation.
```

```
4. Make the best A/B choice.
Do not revise the query's meaning to accommodate a candidate. If the
    available evidence cannot resolve the choice reliably, still return
     the best choice, but leave the unresolved interpretation out of
    the final reasoning. A choice does not make uncertain candidate
    details true.

5. Synthesize the query representation.
Return to the query content identified before the comparison.
    Integrate candidate-suggested details only when independently
    supported by the query. Preserve useful connections between
    entities, attributes, actions, and constraints.
The representation should remain informative among many plausible
    retrieval candidates, not merely sufficient to choose between A and
     B. Do not include the candidate-selection process in reasoning or
    answer.
```

Shared query-grounding and output rules for diagnostic generation

```
QUERY-GROUNDED TARGET
1. Preserve the full retrieval-relevant meaning of the query. Retain
    its important content even when some details are unnecessary for a
    local candidate decision.
2. Follow the condition-specific workflow internally; do not narrate
    the annotation workflow in reasoning or answer.
3. In reasoning, describe concrete supported entities, attributes,
    actions, relations, and spatial or temporal cues. For information-
    seeking queries, explain the evidence relationships needed to
    resolve the request.
4. Include useful discriminative details when supported, without
    forcing a fixed number of details or inventing specificity. The
    representation should remain informative among many plausible
    retrieval candidates.
5. The answer is a concise semantic embedding summary. Preserve the
    central meaning and constraints; do not merely repeat the task
    instruction.
6. Preserve uncertainty locally: omit or qualify an uncertain detail
    while keeping other supported content specific and informative.

USEFUL OUTPUT CONTENT
Retain:
- Concrete details directly supported by the query text or visual.
- Interpretations supported by identifiable evidence within the query.
     Connect an interpretation to the evidence supporting it.
- Evidence relationships implied by the request, such as linking an
    entity to an attribute, comparing specified years, or locating the
    numerator and denominator of a requested ratio.

Exclude:
- Candidate-specific answers, names, numbers, quotations, dates, or
    locations that cannot be established from the query.
- Claims that a particular chart, table, page, or layout must exist
    solely because it appears in a candidate. A plausible evidence
    format may be described as a possibility, not an observed fact.
- Broad task descriptions that replace concrete query content, such as
     'retrieve relevant information' or 'focus on visual features'.
- Fine-grained visual, OCR, spatial, or temporal details that cannot
    be inspected reliably.

For information-seeking queries, preserve the requested fact type and
    relationship. State a factual answer only when the query itself
    supports it. A candidate's answer does not become a query fact.
```

```
FINAL CHECK
- Remove references to auxiliary candidates, A/B labels, selection,
    rejection, and the annotation procedure from reasoning and answer.
    A separately required top-level choice field is exempt.
- Preserve comparisons, negations, exclusions, and transformations
    that are explicitly part of the query itself.
- For every factual clause, identify its support in the query. Remove
    clauses supported only by auxiliary content.
- Check that no important query entity, action, relation, or
    constraint disappeared while following the condition-specific
    workflow.
- The final reasoning and answer must stand on their own as an
    analysis of the query.
```

Non-blind output schema

```
OUTPUT
Return one JSON object and nothing else:

{
  "reasoning": "...",
  "answer": "..."
}
```

Blind output schema

```
OUTPUT
Return one JSON object and nothing else:

{
  "choice": "A",
  "reasoning": "...",
  "answer": "..."
}
```

The task adapter is selected from the following exact strings according to the query's task type. Only one adapter is inserted into each request. The general retrieval adapter is the fallback when no task-specific adapter applies.

Task adapters for diagnostic generation

```
TASK ADAPTER: CLASSIFICATION
Describe concrete query-visible category evidence such as shape, parts
    , material, texture, scene context, behavior, or composition. Do
    not stop at saying that the input needs classification. A category
    hypothesis is useful only when accompanied by visible query
    evidence.

TASK ADAPTER: QUESTION ANSWERING
Preserve exactly what is being asked and identify the query evidence
    needed to answer it. Inspect the query visual for relevant entities
    , attributes, relations, text, or events. State a proposed answer
    only if the query itself supports it; otherwise represent the
    requested answer type and evidence without guessing.

TASK ADAPTER: TEXT-TO-IMAGE RETRIEVAL
Preserve named entities, objects, actions, scene, style, modifiers,
    and relations from the text. Describe the concrete visual evidence
```

```
    a relevant image should satisfy without inventing details absent
    from the query.

TASK ADAPTER: IMAGE-TO-TEXT RETRIEVAL
Describe concrete observable image content and the textual semantics
    that should represent it, including salient entities, actions,
    scene, and relations.

TASK ADAPTER: COMPOSED IMAGE RETRIEVAL
Separate what must be retained from the reference image from what the
    modification text changes. State both retained visual identity and
    requested transformation concretely.

TASK ADAPTER: VISUAL GROUNDING
Represent the referred entity using query-visible attributes and
    spatial relations. Include location or relation cues that
    distinguish the referent from nearby alternatives.

TASK ADAPTER: TEXT-TO-VIDEO RETRIEVAL
Preserve entities, actions, event order, setting, and temporal
    qualifiers. Describe the observable event a relevant video should
    contain.

TASK ADAPTER: VIDEO MOMENT RETRIEVAL
Represent the target event, participating entities, action state,
    temporal order, and useful start/end cues supported by the query.

TASK ADAPTER: VISUAL DOCUMENT RETRIEVAL
Preserve the query's exact entities, qualifiers, requested fact type,
    and evidence relationship. Describe what a relevant page must
    establish without inventing the page's answer, OCR text, layout,
    table values, or chart values.

TASK ADAPTER: GENERAL RETRIEVAL
Preserve concrete query content, relations, constraints, and the
    evidence a relevant result should satisfy.
```

A.2.5 BLINDED TRACE-AUDIT INSTRUCTION

This is a model-based qualitative judgment. The boxes below reproduce the evidence, rating, and per-trace flag instructions from the operational review protocol. Each packet provides the full query, positive, random negative, available media, and anonymous left/right traces. Variant names and predicted choice labels are masked, although the prose can sometimes reveal the condition. The judge model considers reasoning and answer together. The signed margin measures judged query-grounded retrieval usefulness, not an observed change in retrieval accuracy.

The expanded audit also records that prior results had been unblinded before expansion; it should therefore not be described as fully independent blinding. Leakage flags include explicit narration of candidate selection or comparison, not only the importation of candidate-only facts. These flags inform but do not mechanically determine the pair rating.

Operational audit instruction: evidence and masking

```
Read every assigned source's full query, positive, random negative,
    reasoning,
and answer. Display and inspect its actual media, including sampled
    video
frames, not merely paths or encoded bytes. Enlarge/crop original
    document pages
when the contact sheet cannot support a claimed detail. A contact
    sheet with
```

```
unreadable OCR does not establish that an OCR claim is false. Record
    limitations.
Do not look at annotations.jsonl, private_pair_mapping.jsonl, old
    score files,
or occurrence_mapping.json while scoring. Variant names and predicted
    choice
labels are masked in public packets, though prose can sometimes reveal
     them.

Packets are in canonical/packets/source_NNNN.json. They contain
    distinct
trace IDs and left/right pairs; exact overlaps across the two audit
    designs
have been deduplicated. Never infer ratings from length, keywords, or
    variant
identity. Reading via a tool must return full text; reread any
    truncated output.
```

Operational audit instruction: paired ratings

```
Assess usefulness as query-grounded reasoning for retrieval after the
    auxiliary
candidates are removed. The observed corpus positive and negative help
     evaluate
relevance; do not reward simply guessing a candidate or repeating its
    label.

- L2/R2: the indicated side is clearly better, with an important
    grounded gain
  or correction of a central entity, action, relation, constraint,
    omission,
  factual error, or candidate-dependent answer. Explain the concrete
    difference.
- L1/R1: a smaller but meaningful grounded gain or reduction in
    misleading
  content, not stylistic preference alone.
- T0: no meaningful net difference. Equal quality can be equally good
    or bad.

Consider the reasoning and answer together. Correct contextual
    interpretation
can be useful even when the answer is concise. Longer is not
    automatically
better; extra faithful detail is not automatically harmful. A directly
     grounded
class name is not automatically leakage, and naming a class already
    well
described by the other trace is not automatically a win. Preserve
    meaningful
uncertainty in ambiguous queries. Do not count sample difficulty as a
    causal
effect of verification. Genuine query negations/comparisons are
    permitted.
```

Operational audit instruction: per-trace flags

```
Enter all three flags as explicit booleans once per distinct trace:

- candidate_only_leak: final reasoning/answer imports concrete
    auxiliary-only
```

```
  information not independently supported by the query, or explicitly
    narrates
  candidate identities, labels, choice, rejection, or auxiliary
    comparison.
  A queried fact, plausible evidence relationship, or visibly
    supported class
  hypothesis is not leakage merely because it aligns with the positive
    .
- generic_or_vacuous: the output is predominantly reusable task
    boilerplate
  and fails to retain this query's concrete content/requirements. A
    short but
  specific summary is not vacuous.
- unsupported_claim: a substantive factual clause is contradicted or
    not
  established by available evidence and is presented as fact; includes
     invented
  visual/OCR/action relations. Clearly qualified plausible
    interpretations are
  not automatically false. Flags can co-occur. Candidate-supported but
  query-unsupported claims can carry both leakage and unsupported
    flags when
  asserted as facts about the query.

Flags inform, but do not mechanically determine, the pair rating.
    Describe
tradeoffs explicitly. Identical traces retain identical flags across
    all pairs
and repeated analyses. Identical left/right text must receive T0, not
    blank flags.
```

A.3 Derivation of the Mixture-IWAE Gradient

We derive the gradient of the mixture importance-weighted objective used in PILOT. For conciseness, we suppress the conditioning variables and define

$$p_\theta(z) \coloneqq p_\theta(z \mid x), \qquad q_\theta(z) \coloneqq q_\theta(z \mid x, u), \qquad m_\theta(z) \coloneqq \alpha p_\theta(z) + (1-\alpha) q_\theta(z), \tag{16}$$

where the mixture coefficient $\alpha \in (0,1)$ is independent of θ. We further define

$$w_k = \frac{p_\theta(z_k)\, p_R(y \mid x, z_k)}{m_\theta(z_k)}, \qquad \widehat{Z}_K = \frac{1}{K}\sum_{k=1}^{K} w_k, \qquad \bar{w}_k = \frac{w_k}{\sum_{j=1}^{K} w_j}. \tag{17}$$

Using the reward-induced likelihood $p_R(y \mid x, z) \propto \exp(\beta R_{\mathrm{ret}}(x, z))$, the mixture-IWAE objective is

$$\mathcal{J}_{\mathrm{IW}} = \mathbb{E}_{z_{1:K} \sim m_\theta}\left[\log \widehat{Z}_K\right] \le \log p_\theta(y \mid x). \tag{18}$$

We assume that R_{ret} is treated as a stop-gradient reward. Consequently, $p_R(y \mid x, z)$ has no direct gradient with respect to θ.

Step 1: Differentiate the sampling distribution. Because the samples are drawn from the parameterized mixture m_θ, the score-function identity gives

$$\begin{aligned} \nabla_\theta \mathcal{J}_{\mathrm{IW}} &= \mathbb{E}_{m_\theta^K}\left[\log \widehat{Z}_K \nabla_\theta \log \prod_{k=1}^{K} m_\theta(z_k) + \nabla_\theta \log \widehat{Z}_K\right] \\ &= \mathbb{E}_{m_\theta^K}\left[\sum_{k=1}^{K} \log \widehat{Z}_K \nabla_\theta \log m_\theta(z_k) + \nabla_\theta \log \widehat{Z}_K\right]. \end{aligned} \tag{19}$$

Step 2: Differentiate the importance-weighted estimator. The gradient of the log-average importance weight is

$$\nabla_\theta \log \widehat{Z}_K = \frac{\sum_{k=1}^K \nabla_\theta w_k}{\sum_{j=1}^K w_j} = \sum_{k=1}^K \bar{w}_k \nabla_\theta \log w_k. \tag{20}$$

Under the stop-gradient reward assumption,

$$\begin{aligned} \nabla_\theta \log w_k &= \nabla_\theta \left[\log p_\theta(z_k) + \log p_R(y \mid x, z_k) - \log m_\theta(z_k)\right] \\ &= \nabla_\theta \log p_\theta(z_k) - \nabla_\theta \log m_\theta(z_k). \end{aligned} \tag{21}$$

Substituting Eq. 21 into Eq. 20 gives

$$\nabla_\theta \log \widehat{Z}_K = \sum_{k=1}^K \bar{w}_k \nabla_\theta \log p_\theta(z_k) - \sum_{k=1}^K \bar{w}_k \nabla_\theta \log m_\theta(z_k). \tag{22}$$

Combining Eqs. 19 and 22, we obtain

$$\nabla_\theta \mathcal{J}_{\mathrm{IW}} = \mathbb{E}_{m_\theta^K}\left[\sum_{k=1}^K \left(\log \widehat{Z}_K - \bar{w}_k\right) \nabla_\theta \log m_\theta(z_k) + \sum_{k=1}^K \bar{w}_k \nabla_\theta \log p_\theta(z_k)\right]. \tag{23}$$

A baseline $b_k(z_{-k})$ that does not depend on z_k may be subtracted from the score-function coefficient because

$$\mathbb{E}_{z_k \sim m_\theta}\left[b_k(z_{-k}) \nabla_\theta \log m_\theta(z_k)\right] = 0. \tag{24}$$

We therefore define the mixture score-function signal as

$$A_k^{\mathrm{mix}} \coloneqq \log \widehat{Z}_K - \bar{w}_k - b_k(z_{-k}). \tag{25}$$

The baseline can, for example, be a leave-one-out estimate constructed from the remaining $K-1$ trajectories.

Step 3: Decompose the mixture score. Define the posterior responsibilities of the two mixture components:

$$\rho_k^p = \frac{\alpha p_\theta(z_k)}{m_\theta(z_k)}, \qquad \rho_k^q = \frac{(1-\alpha) q_\theta(z_k)}{m_\theta(z_k)}, \qquad \rho_k^p + \rho_k^q = 1. \tag{26}$$

The score of the mixture can then be decomposed as

$$\begin{aligned} \nabla_\theta \log m_\theta(z_k) &= \frac{\alpha \nabla_\theta p_\theta(z_k) + (1-\alpha) \nabla_\theta q_\theta(z_k)}{m_\theta(z_k)} \\ &= \rho_k^p \nabla_\theta \log p_\theta(z_k) + \rho_k^q \nabla_\theta \log q_\theta(z_k). \end{aligned} \tag{27}$$

Substituting Eq. 27 into Eq. 23 yields the exact three-term decomposition

$$\begin{aligned} \nabla_\theta \mathcal{J}_{\mathrm{IW}} = &\underbrace{\mathbb{E}_{m_\theta^K}\left[\sum_{k=1}^K A_k^{\mathrm{mix}} \rho_k^p \nabla_\theta \log p_\theta(z_k)\right]}_{\text{prior policy improvement}} \\ &+ \underbrace{\mathbb{E}_{m_\theta^K}\left[\sum_{k=1}^K A_k^{\mathrm{mix}} \rho_k^q \nabla_\theta \log q_\theta(z_k)\right]}_{\text{posterior policy improvement}} \\ &+ \underbrace{\mathbb{E}_{m_\theta^K}\left[\sum_{k=1}^K \bar{w}_k \nabla_\theta \log p_\theta(z_k)\right]}_{\text{importance-weighted prior learning}}. \end{aligned} \tag{28}$$

The first two terms arise because the prior and posterior jointly define the parameterized proposal from which latent reasoning trajectories are explored. The third term arises from the prior in the numerator of the importance weight and increases the prior likelihood of trajectories that receive high normalized importance weight.

Step 4: Recover the compact expression. For any function $f(z)$, the mixture responsibilities satisfy

$$\mathbb{E}_{z\sim m_\theta}\left[\rho^p(z)f(z)\right] = \alpha\mathbb{E}_{z\sim p_\theta}[f(z)], \qquad \mathbb{E}_{z\sim m_\theta}\left[\rho^q(z)f(z)\right] = (1-\alpha)\mathbb{E}_{z\sim q_\theta}[f(z)]. \tag{29}$$

Therefore, after absorbing α, $1-\alpha$, the sum over K samples, and the expectation over the remaining trajectories into the definitions of A^p and A^q, Eq. 28 can be written compactly as

$$\nabla_\theta \mathcal{J}_{\mathrm{IW}} = \underbrace{\mathbb{E}_{p_\theta}\left[A^p \nabla_\theta \log p_\theta\right]}_{\text{prior improvement}} + \underbrace{\mathbb{E}_{q_\theta}\left[A^q \nabla_\theta \log q_\theta\right]}_{\text{posterior improvement}} + \underbrace{\mathbb{E}_{m_\theta}\left[\bar{w} \nabla_\theta \log p_\theta\right]}_{\text{weighted prior learning}}. \tag{30}$$

This is the compact gradient expression reported in the main text.

Connection to the practical objective. The first two terms are score-function policy gradients for the prior and posterior. In practice, we replace their mixture learning signal A_k^{mix} with group-relative advantages computed from the inference-compatible reward R_k and optimize them with GRPO, which reduces variance. This substitution is an approximation rather than an unbiased estimator of Eq. 28. Since A_k^{mix} is computed from $\log w_k = \log p_\theta(z_k) + \beta R_{\mathrm{ret}}(x, z_k) - \log m_\theta(z_k)$, it favors trajectories that are both useful and reachable under the prior, whereas the GRPO advantage retains only the reward. In the practical objective, reachability therefore enters only through the third term, which we implement as an importance-weighted likelihood loss,

$$\mathcal{L}_{\mathrm{IW}} = -\mathbb{E}_{z_{1:K}\sim m_\theta}\left[\sum_{k=1}^{K} \mathrm{sg}[\bar{w}_k] \log p_\theta(z_k \mid x)\right], \tag{31}$$

where $\mathrm{sg}[\cdot]$ stops gradients through the normalized importance weights. The resulting practical objective combines prior GRPO, posterior GRPO, and importance-weighted prior learning.

Remark on differentiable rewards. If gradients are propagated through the retrieval reward, then

$$\nabla_\theta \log w_k = \nabla_\theta \log p_\theta(z_k) - \nabla_\theta \log m_\theta(z_k) + \beta \nabla_\theta R_{\mathrm{ret}}(x, z_k), \tag{32}$$

and the exact gradient contains the additional term

$$\beta\, \mathbb{E}_{m_\theta^K}\left[\sum_{k=1}^{K} \bar{w}_k \nabla_\theta R_{\mathrm{ret}}(x, z_k)\right]. \tag{33}$$

The three-term decomposition in Eq. 28 therefore assumes that the retrieval reward is detached, as is standard in policy-gradient optimization. If the alternative temperature parameterization $p_R \propto \exp(R_{\mathrm{ret}}/\beta)$ is used, the derivation remains unchanged after replacing β by $1/\beta$.

A.4 Probe Dataset Construction

We construct MMEB-2-Probe, a compact diagnostic dataset for controlled mechanism analyses. The probe contains 1,000 query–candidate triplets drawn from all 78 subsets of MMEB-2, with a nearly uniform allocation of 12 or 13 queries per subset. This prevents large subsets from dominating the analysis while preserving coverage of all benchmark domains and task types. The resulting dataset contains 447 image, 220 video, and 333 visual-document queries. The image portion covers classification, question answering, image-to-text retrieval, text-to-image retrieval, composed-image retrieval, and visual grounding; the video portion covers classification, question answering, text-to-video retrieval, and moment retrieval; and the visual-document portion covers document retrieval. Each example retains its source subset, task type, benchmark row index, and original text, image, or video modalities.

For each query, we construct a subset-specific candidate pool containing candidates that serve as official positives for other queries. Queries are then deterministically sampled within each subset according to the assigned quota. After excluding every official positive associated with the current query, we uniformly sample one candidate as its random negative. Sampling is deterministic and performed independently across queries, allowing the same candidate to be selected for multiple queries. Among the 730 sampled negatives present in the saved retrieval rankings, the median rank is 316.5 and only 50 appear within the top 10, confirming that the resulting negatives are not

systematically selected for difficulty. The corresponding query, positive, and negative contents are loaded through the official MMEB-2 dataset loaders, ensuring that the stored multimodal inputs match those used during benchmark evaluation.

The final probe therefore provides one query, one official positive, and one qrel-safe random negative per instance. Retrieval rewards or other relative metrics are computed against the expanded subset-level candidate pool rather than against this single negative alone. Because the probe is sampled from MMEB-2 evaluation data, it is used exclusively for diagnostic and mechanism analyses and is never used for model training, hyperparameter selection, or checkpoint selection.

A.5 Detailed RQ1 Experiment

This experiment examines whether privileged candidate information helps an annotation model produce higher-quality query-side reasoning traces. In particular, we ask whether **a positive candidate**, **a labeled positive–negative pair**, or **an unlabeled candidate pair** can reveal retrieval-relevant information that is difficult to identify from the query alone. We additionally distinguish transferable improvements, which remain supported by the query after the candidates are removed, from improvements caused by copying candidate-only information into the reasoning trace.

Prompt conditions. For every query, we compare four query-side annotation conditions:

1. **Input only**: the annotation model observes only the original multimodal query.
2. **Input + positive**: the annotation model observes the query and its positive candidate.
3. **Input + labeled positive/negative**: the annotation model observes the query, a candidate explicitly labeled as positive, and a candidate explicitly labeled as negative.
4. **Input + blind candidate pair**: the annotation model observes the query and the same two candidates in a randomly shuffled A/B order, without being told which candidate is positive. It must first predict the positive candidate and then generate the reasoning trace.

All conditions use the same task-specific instructions and output schema. The candidate-conditioned prompts instruct the annotation model to treat candidates as privileged attention cues rather than authoritative factual sources. Candidate information may be used to propose hypotheses about discriminative entities, attributes, relations, actions, spatial configurations, temporal events, or evidence types. However, the annotation model must re-inspect the query and retain only properties independently supported by the query. The final reasoning is instructed to be self-contained and must not mention candidates, positive/negative labels, comparisons, elimination, or A/B identifiers. The blind condition additionally outputs a separate top-level `choice` field, which is removed from the reasoning content used in the pairwise comparison. Details of the used prompt can be seen in Appendix A.2.

Reasoning generation. We use `GLM-4.1V-thinking` as the annotation model, served through a vLLM-compatible OpenAI API. We independently sample eight reasoning traces for each query under each of the four prompt conditions, producing $1{,}000 \times 4 \times 8 = 32{,}000$ annotations in total. Sampling uses temperature 1.0, top-$p = 0.95$, top-$k = 20$, presence penalty 1.5, and repetition penalty 1.0. For video inputs, at most eight frames are supplied to the expert. The model returns a JSON object containing `reasoning` and `answer`; the blind condition additionally returns its predicted A/B choice.

The output-token limit was increased during generation to handle occasional empty responses from the serving backend. The limits used across generation segments were 4,096, 8,192, 80,000, and finally 16,384 tokens, while all prompts and sampling parameters remained unchanged. Only three successful generations reached 4,096 tokens and one reached 8,192 tokens, so truncation affected at most four of the 32,000 generated records.

Blinded pairwise audit. We conduct a blinded, model-based pairwise audit at two scales. For the main trace-quality analysis, we evaluate all 1,000 queries in the probe dataset. For each query, we select one of the eight sampled traces from each prompt condition, using the same sampling index across conditions. Each candidate-conditioned trace is paired with the corresponding input-only trace, yielding $1{,}000 \times 3 = 3{,}000$ pairwise judgments over 4,000 selected traces. This audit

supports the main comparison (Table 4), candidate-leakage stratification (Table 5), domain-wise results (Table 6), and blind-prediction correctness analysis (Table 7) reported in the first four tables. Separately, we evaluate all eight samples per condition on an 81-query subset for the repeated-sampling analysis. Pairing each candidate-conditioned trace with the correspondingly indexed input-only trace yields $81 \times 3 \times 8 = 1{,}944$ pairwise judgments over 2,592 selected traces. This analysis supports the mean-of-eight, single-sample variability, and oracle best-of-eight results reported in Table 8.

For each pair, the input-only and candidate-conditioned traces are randomly assigned to the left and right positions. Pair order is also randomized. The reviewer has access to the complete multimodal query, positive candidate, hard negative, and both reasoning traces, but does not observe the prompt-condition identity or the hidden left/right mapping. The mapping is revealed only after all judgments are completed.

The reviewer answers the following question: *Which trace would produce the better query embedding for retrieving the positive candidate while rejecting the hard negative after the candidates are removed?* A trace is preferred when it contains more query-supported discriminative information, omits fewer important query constraints, and contains fewer unsupported claims, generic statements, or candidate-only details.

Pairwise rating scale. Each comparison receives one of five ordinal ratings:

$$r_i \in \{-2, -1, 0, +1, +2\},$$

where the sign is defined from the candidate-conditioned trace's perspective. A score of $+1$ indicates a small but meaningful advantage for the conditioned trace, and $+2$ indicates a clear advantage likely to affect positive-versus-negative ranking. Scores of -1 and -2 analogously favor the input-only trace, while 0 denotes no meaningful retrieval-quality difference.

Let N be the number of evaluated pairs for a prompt condition. We report:

$$W = \sum_i \mathbb{I}[r_i > 0], \quad T = \sum_i \mathbb{I}[r_i = 0], \quad L = \sum_i \mathbb{I}[r_i < 0].$$

The **net win rate** measures the excess frequency of wins over losses:

$$\mathrm{NetWin} = \frac{W - L}{N}.$$

Unlike the ordinary win rate, this metric treats losses symmetrically and assigns ties zero contribution. Its range is $[-1, 1]$.

The **mean margin** incorporates both preference direction and strength:

$$\mathrm{Margin} = \frac{1}{N} \sum_{i=1}^{N} r_i,$$

with range $[-2, 2]$. A positive value favors candidate-conditioned reasoning. We compute 95% confidence intervals for the mean margin using 10,000 bootstrap replicates clustered by source query. Specifically, we first average the eight judgment margins for each query and then resample the 81 query-level means with replacement. This avoids treating repeated generations from the same query as independent observations.

Failure indicators. In addition to pairwise preferences, the reviewer independently marks three binary properties for each trace. **Candidate-only leakage** indicates that the trace contains a concrete entity, answer, number, date, quotation, category, or visual detail that is available from a candidate but cannot be independently inferred from the query. Its rate is

$$\mathrm{LeakRate} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[\text{conditioned trace } i \text{ leaks}].$$

This is an operational measure of train–test information mismatch rather than benchmark contamination: such information is available to the posterior annotator but unavailable to the query-only model at deployment.

Table 4: Model-based pairwise audit of query-side reasoning on 1,000 MMEB-2 queries (v3-delimited prompts; random negatives). Each candidate-conditioned trace is compared with the input-only trace for the same query and sample index. W/T/L counts wins, ties, and losses; net win is $(W - L)/N$. Margin is the mean signed preference score in $\{-2, -1, 0, 1, 2\}$. Intervals are query-level bootstrap 95% confidence intervals for mean margin. Leak denotes the fraction flagged for candidate-only information.

Prompt condition	N	W/T/L	Net win	Margin	95% CI	Leak
Input + positive	1000	445/393/162	0.283	0.451	[0.388, 0.514]	0.274
Input + positive + negative (labeled)	1000	406/416/178	0.228	0.385	[0.322, 0.448]	0.236
Input + positive + negative (blind)	1000	608/216/176	0.432	0.772	[0.698, 0.845]	0.796

Table 5: Pairwise reasoning quality stratified by candidate-only information leakage. Scores compare each trace with its matched input-only baseline. These post-generation strata are descriptive and do not identify the causal effect of leakage.

Prompt condition	Stratum	N	W/T/L	Net win	Margin
Input + positive	No leak	726	238/367/121	0.161	0.249
	Leak	274	207/26/41	0.606	0.985
Input + positive + negative (labeled)	No leak	764	226/392/146	0.105	0.173
	Leak	236	180/24/32	0.627	1.072
Input + positive + negative (blind)	No leak	204	71/104/29	0.206	0.328
	Leak	796	537/112/147	0.490	0.886

Generic or vacuous marks traces dominated by statements such as "retrieve relevant information" or "focus on visual features" without concrete query semantics. **Unsupported claim** marks factual, visual, OCR, spatial, or temporal assertions not justified by the query. Candidate specificity is counted as a genuine improvement only when it is independently supported by the query.

Repeated-sampling analyses. The primary result averages all eight stochastic samples per condition. We also report a **single-sample** estimate using the predefined sample with index zero. Finally, we report an **oracle best-of-eight** upper bound:

$$\mathrm{OracleMargin}_s = \max_{k \in \{1,\ldots,8\}} r_{s,k}.$$

This oracle uses the pairwise audit score to select the best conditioned trace for each query and therefore should not be interpreted as a deployable result. It quantifies the potential benefit available to a sufficiently accurate verifier or reasoning-selection mechanism.

Blind verification analysis. For the blind condition, we additionally stratify generations by whether the expert correctly identifies the positive candidate. This is a conditional association rather than a causal estimate of verification: queries on which the expert predicts correctly may be intrinsically easier than those on which it predicts incorrectly. Within the 81-query audit set, 521 of 648 blind generations are prediction-correct and 127 are prediction-incorrect.

A.6 Detailed RQ3 Experiment

To check whether the transfer weights actually discriminate among posterior traces, we track their effective sample size during training of the Full model. For each query with at least one eligible posterior trace, we compute $\mathrm{ESS} = 1/\sum_k w_k^2$ over its posterior traces, where a trace is eligible if its reward exceeds the mean reward of the prior group by more than $\max(0.05,\ 4\sigma_{\text{prior}})$, and average over such queries at every step. As a reference we also report the mean number of eligible traces per query, which is the ESS that uniform weights would attain. The ESS rises from about 2.1 in the first 20 steps to about 3.6 in the last part of training, while the number of eligible traces grows to 6–7.5 per query. Throughout training the ESS stays at roughly half of this bound (mean ratio 0.53). The

Table 6: Pairwise reasoning-quality audit by MMEB-2 domain. All prompt conditions include the query input and are compared against matched input-only traces. Rates are reported as fractions.

Domain	Prompt condition	N	W/T/L	Net win	Margin	Leak
Image	Input + positive	447	156/239/52	0.233	0.306	0.199
	Input + positive + negative (labeled)	447	140/239/68	0.161	0.237	0.130
	Input + positive + negative (blind)	447	196/150/101	0.213	0.391	0.725
Video	Input + positive	220	101/87/32	0.314	0.473	0.164
	Input + positive + negative (labeled)	220	94/87/39	0.250	0.400	0.136
	Input + positive + negative (blind)	220	122/57/41	0.368	0.541	0.700
VisDoc	Input + positive	333	188/67/78	0.330	0.631	0.447
	Input + positive + negative (labeled)	333	172/90/71	0.303	0.574	0.444
	Input + positive + negative (blind)	333	290/9/34	0.769	1.435	0.955

Table 7: Blind-conditioned reasoning stratified by whether the expert correctly selected the positive candidate. Unsupported denotes the fraction of traces flagged for unsupported claims. Differences are descriptive: correct and incorrect groups may differ in query difficulty and do not establish a causal effect of verification.

Blind prediction	N	W/T/L	Net win	Margin	Leak	Unsupported
Correct	971	602/212/157	0.458	0.818	0.803	0.092
Incorrect	29	6/4/19	-0.448	-0.759	0.552	0.483

weights therefore do not spread evenly over the eligible traces but concentrate on a subset of them, so transfer is dominated by a few selected traces per query rather than by averaging over all traces that clear the eligibility threshold.

We measure how strongly the posterior traces depend on the privileged information u through the signed log-probability gap. For each posterior trace z we compute the token-averaged difference $\frac{1}{|z|}\sum_t \left[\log \pi_\theta(z_t \mid x, u, z_{<t}) - \log \pi_\theta(z_t \mid x, z_{<t})\right]$ between the posterior and prior prompts and average it over the batch. A lower gap means the trace is more likely under the prior prompt alone, i.e., it can be reproduced without access to u. We plot each dataset over its own block. We compare Full with w/o TS, which disables transfer SFT. Full has a lower gap on all ten datasets, e.g., 0.30 versus 0.60 on CIRR, 0.33 versus 0.68 on Visual7W and 0.55 versus 0.83 on ColPali (averaged over each block). Under w/o TS the gap also grows within several blocks, most visibly on LLaVA-Hound Caption and Visual7W, whereas under Full it stays flat or decreases. Transfer SFT thus keeps the posterior traces reachable by the prior, which is the property the transfer step relies on.

Table 8: Repeated-sampling audit on 81 queries with eight samples per prompt condition. Top: aggregate results over 648 paired comparisons per condition; confidence intervals resample queries as clusters. Bottom: sample index 0 versus oracle-best-of-8, each with 81 comparisons per condition. The oracle maximizes each query's paired margin against the input-only trace at the corresponding sample index. It is a retrospective paired-score upper bound, not an executable selection policy or an absolute-quality oracle.

Prompt condition	N	W/T/L	Net win	Margin	95% CI	Leak
Input + positive	648	296/243/109	0.289	0.503	[0.350, 0.660]	0.250
Input + positive + negative (labeled)	648	265/262/121	0.222	0.395	[0.248, 0.545]	0.211
Input + positive + negative (blind)	648	441/117/90	0.542	0.937	[0.739, 1.131]	0.753

Prompt condition	Selection	Net win	Margin
Input + positive	Sample index 0	0.358	0.593
	Oracle-best-of-8	0.926	1.506
Input + positive + negative (labeled)	Sample index 0	0.099	0.259
	Oracle-best-of-8	0.889	1.383
Input + positive + negative (blind)	Sample index 0	0.568	0.988
	Oracle-best-of-8	0.951	1.580



Figure 5: Effective sample size (ESS) of the transfer weights during training (Full).

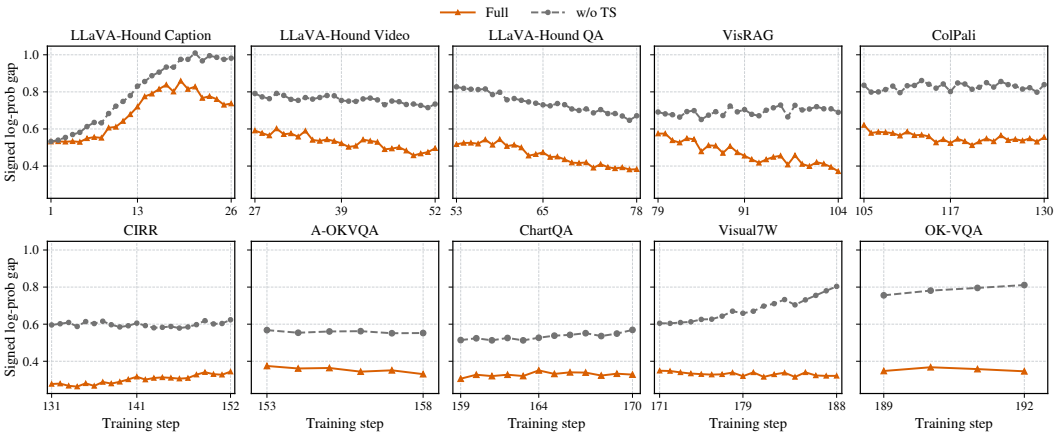

Figure 6: Signed log-probability gap between the posterior and prior prompts on posterior traces, per training subset.