EACACHE: ENTAILMENT-AWARE CACHING FACILITATING AI IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image generation is computationally expensive, motivating the reuse of previously generated images through caching. Similarity-based cache retrieval is widely used, but image–text similarity may not always accurately indicate whether a cached image contains sufficient content to satisfy a user request. This distinction is especially important for general prompts, where a valid cached image may contain substantial additional content while still satisfying the requested content, which reflects the entailment effect. We introduce **EACache**, an entailment-aware caching framework that combines similarity-based retrieval with selective entailment verification (e-verification). EACache derives two similarity thresholds with the objective of minimizing the expected system cost of serving a query. When the similarity score exceeds the higher threshold, the cached image is returned directly; when the score falls below the lower threshold, a new image is generated; and when the score lies between the two thresholds, e-verification is performed to determine whether the cached image can be reused. The threshold derivation jointly accounts for the costs of e-verification, image generation, transmission, and recovery from an incorrect cache-hit decision. To reduce online e-verification overhead, EACache captions each image when it is added to the cache and performs textual entailment between the stored caption and the incoming query at runtime. Experiments show that similarity remains an informative signal but cannot cleanly separate reusable from non-reusable candidates, while selective e-verification recovers valid cache hits missed by similarity-only reuse without requiring verification for every request. The optimal threshold analysis further provides practical guidance for selecting an appropriate entailment model for caching system design.

1 INTRODUCTION

Text-to-image generation is computationally expensive, motivating the reuse of previously generated results through semantic caching. Existing semantic caches have primarily been developed for text-based LLM applications, where a cached response is reused when the embedding similarity between a new query and a cached query exceeds a decision boundary (Bang, 2023; Schroeder et al., 2026). Extending this idea to image generation appears natural because CLIP-style models provide an efficient measure of cross-modal similarity between a text query and a cached image (Radford et al., 2021).

Image reuse requires directional semantic coverage rather than overall semantic closeness. We define an image–query pair as *entailed* when the visual content of the cached image provides sufficient evidence to support all content requested by the user query. Importantly, the image may contain additional objects, attributes, or actions that are not mentioned in the query without violating entailment. In contrast, an image–query pair is *contradicted* when the visual content provides evidence that conflicts with one or more claims required by the query. This directional notion follows the standard entailment formulation, where evidence serves as the premise and the requested statement as the hypothesis (Xie et al., 2019). Prior work has also shown that symmetric image–text similarity does not fully capture such asymmetric or fine-grained semantic relationships (Vendrov et al., 2016; Yuksekgonul et al., 2023; Kamath et al., 2024; Koishigarina et al., 2026).

This distinction creates a practical failure mode for similarity-based cache reuse. As shown in Figure 1, some valid cache-hit opportunities have relatively low image–query similarity and would

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: **Examples of semantically sufficient cached images with relatively low image–query similarity.** These examples are rejected by a high similarity threshold but recovered by the entailment verification model.

therefore be rejected by a high similarity threshold. In these examples, the *premise* describes the cached image and the *hypothesis* corresponds to the user query; although the cached images contain additional visual information, they still entail the requested content. These examples show that similarity alone can discard semantically valid cache-hit opportunities.

Applying entailment verification (e-verification) to every retrieved candidate, however, would introduce additional inference cost and reduce the efficiency benefit of caching. The key question is therefore not simply whether entailment can improve cache-hit decisions, but *when* e-verification is worth its cost. This decision also depends on the entailment model itself, since different entailment models exhibit different prediction behavior and inference overhead.

To address this problem, we introduce **EACache**, an entailment-aware caching framework for deciding when previously generated images can be reused to serve new user queries. EACache uses image–query similarity as an efficient first-stage signal and derives two model-specific decision thresholds from expected system cost. High-similarity candidates are returned directly, intermediate candidates undergo e-verification, and low-similarity candidates trigger new image generation. If the entailment model rejects an intermediate candidate, a new image is also generated. In this way, EACache applies e-verification only where its additional semantic information can justify its computational cost.

To determine the two thresholds, EACache estimates how these outcomes vary with similarity: how often a candidate is judged satisfactory when returned directly, how often the entailment model predicts entailment, and, among candidates predicted as entailed, how often the candidate is judged satisfactory. These three probabilities are combined with the costs of image generation, e-verification, transmission, and recovery from an incorrect return to derive the two decision thresholds. This allows the thresholds to reflect both the behavior of the selected entailment model and the operating costs of the service.

Our main contributions are:

- We develop **EACache**, an **entailment-aware caching framework** that combines similarity-based retrieval with selective e-verification to improve image reuse while reducing unnecessary generation.
- We introduce a **cost-aware selective policy** in which two similarity thresholds are rigorously derived to define three decision regions: **direct reuse**, **e-verification**, and **new generation**. Our analytical model explicitly incorporates **generation cost**, **e-verification cost**, **communication cost**, and the stochastic relationships among **user satisfaction**, **similarity**, and **entailment**.
- We demonstrate how EACache can guide **entailment model selection** by comparing textual and multimodal entailment models. Our results show that model choice depends not only on standalone entailment accuracy, but also on **inference overhead** and the **reliability of accepted cache returns**, providing practical guidance for service providers.

2 Related Work

Semantic caching. Semantic caching reduces repeated model inference by reusing outputs for semantically related requests. Existing systems primarily target text-based LLM applications. GPTCache (Bang, 2023) stores LLM responses and determines reuse through embedding-based retrieval and similarity evaluation. SCALM (Li et al., 2024) uses hierarchical semantic clustering to identify valuable query patterns and adapt cache storage and eviction, while query matching still relies on similarity thresholding. More recently, vCache (Schroeder et al., 2026) replaces a global fixed threshold with embedding-specific boundaries learned online under a user-defined error-rate guarantee.

These methods differ in threshold estimation and cache management, but focus on textual queries and LLM responses rather than reuse of generated images. Their online decision for a retrieved candidate is also essentially binary: reuse the cached output or invoke the model. EACache instead considers image reuse and introduces e-verification as an intermediate action. The system therefore chooses among generation, e-verification, and direct reuse using two model-specific decision boundaries derived from expected risk.

Image–text similarity and entailment. Contrastive vision–language models such as CLIP (Radford et al., 2021) provide effective image–text similarity signals for cross-modal retrieval, but prior work has identified limitations in modeling asymmetric and fine-grained semantic relationships. Vendrov et al. (Vendrov et al., 2016) model image–language semantics using a partial order, while Yuksekgonul et al. (Yuksekgonul et al., 2023) show that strong retrieval performance can coexist with weaknesses in compositional understanding. Kamath et al. (Kamath et al., 2024) further show that hard-negative fine-tuning can reduce performance on semantically valid hard positives, and Koishigarina et al. (Koishigarina et al., 2026) suggest that some compositional information may be present unimodally but imperfectly aligned across modalities. These findings motivate complementing similarity with explicit e-verification for cache-reuse decisions.

3 EACache

3.1 System Overview

Figure 2 illustrates the overall EACache workflow. We consider a provider-side image-generation service that maintains a cache of previously generated images. Each cached entry contains an image, its detailed caption, and a precomputed image embedding, respectively:

$$\mathcal{C} = \{(I_j, c_j, \mathbf{z}_j)\}_{j=1}^{N}, \tag{1}$$

where I_j, c_j, and $\mathbf{z}_j$ denote the cached image, caption, and image embedding, respectively. The caption is generated once when the image is added to the cache and can be reused for subsequent queries.

Given an incoming query q, EACache computes its text embedding $\mathbf{u}_q$ and retrieves the cached image with the highest image–text similarity:

$$j^* = \arg\max_{j \in \{1,\dots,N\}} \mathrm{sim}(\mathbf{z}_j, \mathbf{u}_q), \qquad s = \mathrm{sim}(\mathbf{z}_{j^*}, \mathbf{u}_q). \tag{2}$$

The top-1 similarity s is then used by EACache to determine whether the retrieved candidate should trigger generation, undergo e-verification, or be returned directly. The corresponding decision thresholds are derived from the expected costs of these three actions below.

3.2 System Cost Evaluating a Query-Cache Pair

This part , we dervie the average system costs when the caching system make decisions on how to serve a query. Let C_g denote the cost of image generation, $C_{e,m}$ the inference cost of model m, and C_{tx} the cost of transmitting one image to the user. We use C_{rec} to denote the additional recovery penalty incurred when a returned cached image fails to satisfy the request. Let K denote the similarity score computation cost.

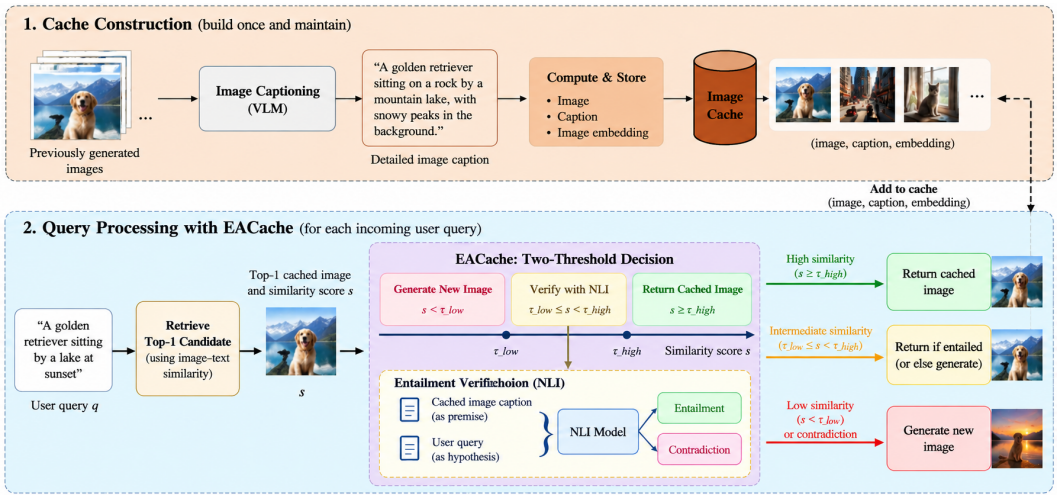

Figure 2: **Overview of EACache.** The cache stores previously generated images together with their captions and image embeddings. When the cache construction is ready, for each incoming query, EACache retrieves the top-1 cached image using image–text similarity and obtains its similarity score s. The two-threshold policy then determines the action: candidates with $s \geq \tau_{\text{high}}$ are returned directly, candidates with $\tau_{\text{low}} \leq s < \tau_{\text{high}}$ undergo e-verification using the cached caption as the premise and the user query as the hypothesis, and candidates with $s < \tau_{\text{low}}$ trigger new image generation. If e-verification rejects the candidate, a new image is also generated. Newly generated images can subsequently be added to the cache.

For a candidate with similarity s, let $\widehat{E}_m$ denote the binary decision of model m, and let G denote the event that the cached image really satisfies the user. We define

$$e_m(s) = P(\widehat{E}_m = 1 \mid s), \tag{3}$$

$$r_m(s) = P(G = 1 \mid \widehat{E}_m = 1, s), \tag{4}$$

$$g(s) = P(G = 1 \mid s). \tag{5}$$

Here, $e_m(s)$ is the probability that model m accepts a candidate, $r_m(s)$ is the satisfaction probability among accepted candidates, and $g(s)$ is the satisfaction probability when a candidate is returned without e-verification.

The expected costs of generation, direct return, and e-verification are

$$R_G(s) = K + C_g + C_{\text{tx}}, \tag{6}$$

$$R_D(s) = K + C_{\text{tx}} + [1 - g(s)]\,(C_g + C_{\text{tx}} + C_{\text{rec}})\,, \tag{7}$$

$$\begin{aligned} R_{V,m}(s) = {} & K + C_{e,m} + C_{\text{tx}} + [1 - e_m(s)]C_g \\ & + e_m(s)[1 - r_m(s)]\,(C_g + C_{\text{tx}} + C_{\text{rec}})\,. \end{aligned} \tag{8}$$

Equation 6 describes immediate generation, which incurs the common cache-search cost K, one image-generation cost C_g, and one image-transmission cost C_{tx}.

Equation 7 describes direct cache reuse. The cached image is always transmitted once, contributing C_{tx}. With probability $1 - g(s)$, the returned image does not satisfy the request. The service must then generate and transmit a replacement and incur the recovery penalty, adding $C_g + C_{\text{tx}} + C_{\text{rec}}$.

Equation 8 describes e-verification. The model first incurs $C_{e,m}$, and one image is eventually transmitted regardless of the e-verification outcome. If the model rejects the candidate, with probability $1 - e_m(s)$, the service generates a new image, contributing C_g. If the model accepts but the returned cached image does not satisfy the user, with probability $e_m(s)[1 - r_m(s)]$, the service must generate and transmit a replacement and incur the recovery penalty, adding $C_g + C_{\text{tx}} + C_{\text{rec}}$.

3.3 Optimal Decision Thresholds

EACache derives the thresholds from the candidate-level expected costs. When the minimum-cost action changes from generation to e-verification and then to direct return as similarity increases, the

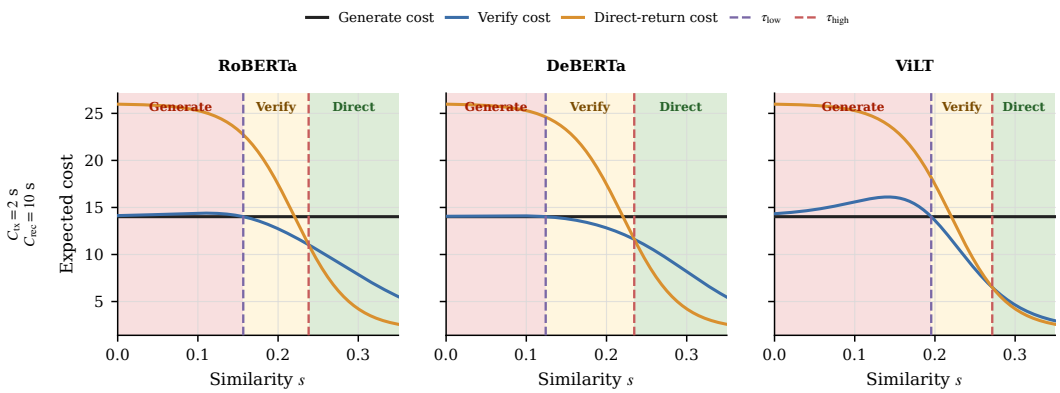

Figure 3: **Learned thresholds from expected-cost intersections.** For each model m, $\tau_{\text{low},m}$ is obtained from the crossing of $R_G(s)$ and $R_{V,m}(s)$, while $\tau_{\text{high},m}$ is obtained from the crossing of $R_{V,m}(s)$ and $R_D(s)$.

lower threshold is defined by

$$R_G(\tau_{\text{low},m}) = R_{V,m}(\tau_{\text{low},m}), \tag{9}$$

and the upper threshold is defined by

$$R_{V,m}(\tau_{\text{high},m}) = R_D(\tau_{\text{high},m}). \tag{10}$$

The two thresholds partition cached candidates into three similarity regions:

$$a_m(s_j) = \begin{cases} \text{invoke generation}, & s_j < \tau_{\text{low},m}, \\ \text{verify}, & \tau_{\text{low},m} \le s_j < \tau_{\text{high},m}, \\ \text{direct reuse}, & s_j \ge \tau_{\text{high},m}. \end{cases} \tag{11}$$

For an incoming query, EACache first checks whether any cached candidate lies in the direct-reuse region. If such a candidate exists, a qualifying image is returned without e-verification. Otherwise, candidates in the e-verification region are evaluated by model m. If a candidate is accepted, the corresponding cached image is returned. If no candidate is accepted, a new image is generated. If all cached candidates satisfy

$$s_j < \tau_{\text{low},m}, \qquad \forall j, \tag{12}$$

EACache directly invokes image generation without running the entailment model.

Figure 3 illustrates how the two thresholds are obtained from these expected-cost functions. For each entailment model, the three curves correspond to generation, e-verification, and direct reuse. At each similarity value, EACache selects the action with the lowest expected cost. The intersections therefore define the decision boundaries: $\tau_{\text{low},m}$ marks the transition from generation to e-verification, while $\tau_{\text{high},m}$ marks the transition from e-verification to direct reuse. As a result, the similarity axis is partitioned into the three minimum-cost regions.

4 Experiments

4.1 Experimental Setup

Data construction. We evaluate EACache using SNLI-VE/Flickr30K (Xie et al., 2019; Young et al., 2014) and MS COCO (Lin et al., 2014). Existing dataset images serve as cached candidates, while their text descriptions serve as incoming queries.

For SNLI-VE/Flickr30K, we construct 5,000 queries from 2,500 SNLI-VE hypotheses and 2,500 Flickr30K captions. For COCO, we similarly combine one caption from each of 2,500 cache images with captions from 2,500 images outside the cache. The active caches contain 2,451 Flickr30K images and 2,445 COCO images, respectively.

Table 1: Primary experimental parameters and learned thresholds.

Policy	C_e	τ_{low}	τ_{high}	K	C_g	$(C_{\text{tx}}, C_{\text{rec}})$
RoBERTa	0.002271	0.156560	0.238094	0.02	12	(2, 10)
DeBERTa	0.003604	0.124107	0.234591	0.02	12	(2, 10)
ViLT	0.014473	0.195427	0.271504	0.02	12	(2, 10)

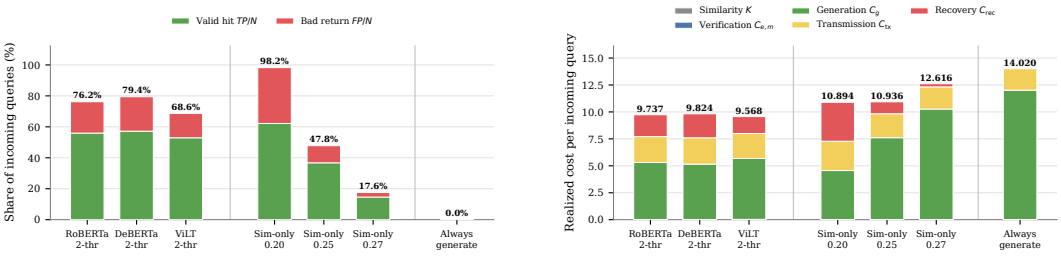

Figure 4: Cache-return quality and realized cost over 4,000 incoming queries. **Left:** valid-hit and bad-return rates; values above the bars report the total cache-return rate. **Right:** realized mean cost per query, decomposed into similarity, e-verification, generation, transmission, and recovery costs. The two-threshold policies achieve lower realized cost than all evaluated similarity-only thresholds.

For each dataset, we sample 1,000 queries to construct similarity-stratified image–query pairs for threshold estimation, yielding 13,308 SNLI-VE/Flickr30K pairs and 13,602 COCO pairs after high-similarity augmentation. The remaining 4,000-query stream is used to evaluate whether the top retrieved image should be returned directly, verified, or rejected in favor of generation.

Models and judgments. We use EVA02-L-14-336 (Fang et al., 2023) with cosine similarity for cache retrieval. We evaluate RoBERTa (Liu et al., 2019), DeBERTa-v3 (He et al., 2023), and ViLT (Kim et al., 2021) as entailment models. RoBERTa and DeBERTa receive the cached image caption and query, whereas ViLT receives the image and query directly. A pair is accepted when entailment is the highest-scoring class among entailment, neutral, and contradiction. We use the fixed GPT-5.2 model (OpenAI, 2025) to judge whether each returned image satisfies its corresponding query, treating scores above 50 as user satisfactory. The shared EVA02-L-14-336 retrieval model contains approximately 428M parameters. The RoBERTa, DeBERTa, and ViLT e-verification models contain approximately 355M, 435M, and 112M parameters, respectively.

Threshold learning and test evaluation. We estimate $g(s)$, $e_m(s)$, and $r_m(s)$ for each model m using the threshold-learning set. The model-specific thresholds $\tau_{\text{low},m}$ and $\tau_{\text{high},m}$ are obtained from the intersections of $R_G(s)$, $R_{V,m}(s)$, and $R_D(s)$. The thresholds are then frozen and evaluated on 4,000 queries using exact top-1 retrieval over the fixed 2,451-image cache.

All cost parameters represent service latency in seconds. We set $K = 0.02$ for the relatively inexpensive similarity search, $C_g = 12$ for image generation, $C_{\text{tx}} = 2$ for transmitting an image, and $C_{\text{rec}} = 10$ for the additional recovery overhead after an unsatisfactory return. These values reflect the expected ordering that similarity search is inexpensive, transmission has a moderate cost, and generation and failure recovery dominate latency. In contrast, $C_{e,m}$ is measured directly from the batched inference time of each verifier. All parameters are fixed before the 4,000-query evaluation and used unchanged across policies. Table 1 summarizes the costs and learned thresholds.

For $N = 4{,}000$ test queries, the realized mean system cost is

$$\overline{C} = \frac{NK + N_{\text{ver}}C_e + N_{\text{gen}}C_g + N_{\text{tx}}C_{\text{tx}} + N_{\text{rec}}C_{\text{rec}}}{N}.$$

4.2 Does EACache Reduce System Cost While Preserving User Satisfaction?

Figure 4 presents the main result of the paper on SNLI-VE/Flickr30K, while Table 2 shows that the same conclusion holds on COCO. Across all three evaluated entailment models, EACache achieves lower realized cost than every evaluated similarity-only threshold. On SNLI-VE/Flickr30K,

Table 2: Mean realized cost per incoming query across datasets. The similarity-only result is its lowest-cost evaluated threshold ($\tau = 0.20$ for both datasets). All costs include recovery after an unsatisfactory cache return.

Policy	SNLI-VE/Flickr30K	COCO
EACache–RoBERTa	9.737	7.693
EACache–DeBERTa	9.824	**7.679**
EACache–ViLT	**9.568**	7.805
Similarity-only ($\tau = 0.20$)	10.894	8.239
Always generate	14.020	14.020

Table 3: Cache-return performance on 4,000 queries. User acceptance is estimated by the satisfaction judge.

Policy	Valid hits	Valid-hit rate	Bad returns	Error rate	Return rate
RoBERTa two-threshold	2,238	55.95%	810	20.25%	76.20%
DeBERTa two-threshold	2,288	57.20%	889	22.23%	79.43%
ViLT two-threshold	2,117	52.93%	629	15.73%	68.65%
Similarity-only 0.20	2,486	62.15%	1,444	36.10%	98.25%
Similarity-only 0.25	1,470	36.75%	442	11.05%	47.80%
Similarity-only 0.27	585	14.63%	117	2.93%	17.55%
Always generate	0	0.00%	0	0.00%	0.00%

RoBERTa-, DeBERTa-, and ViLT-based EACache obtain mean costs of 9.737, 9.824, and 9.568 per incoming query, respectively, compared with 10.894 for the lowest-cost evaluated similarity-only policy and 14.020 for always generating. On COCO, the corresponding EACache costs are 7.693, 7.679, and 7.805, compared with 8.239 for the lowest-cost similarity-only policy and 14.020 for always generating. Importantly, these realized costs already include the recovery cost of failed first-round cache returns: when a returned cached image is unsatisfactory, the system generates and transmits a new image to the user. Thus, the reported improvements reflect end-to-end service cost rather than only first-round cache decisions.

The left panel of Figure 4 shows that the cost reduction on SNLI-VE/Flickr30K is not obtained by simply avoiding cache reuse. EACache still achieves substantial first-round successful reuse, with valid-hit rates of 52.93–57.20%, as detailed in Table 3. In contrast, similarity-only caching exhibits a clear trade-off. At a permissive threshold of 0.20, it attains a higher valid-hit rate of 62.15%, but 36.10% of all queries receive an unsatisfactory cached result. Raising the threshold to 0.25 or 0.27 reduces these bad returns to 11.05% and 2.93%, respectively, but also sharply reduces the valid-hit rate to 36.75% and 14.63%.

COCO exhibits the same trade-off. EACache maintains valid-hit rates of 68.10–68.42%, whereas similarity-only retrieval at threshold 0.20 achieves a higher valid-hit rate of 73.10% but returns an unsatisfactory image for 24.93% of queries. Increasing the threshold to 0.25 or 0.27 reduces the bad-return rate to 10.38% and 4.03%, but also lowers the valid-hit rate to 55.97% and 33.75%, respectively.

Together, these results show across both datasets that EACache reduces provider-side realized cost while preserving substantial successful cache reuse. It does so under a recovery-aware evaluation in which failed first-round cache returns are corrected by generation. The learned thresholds are nevertheless data-dependent: SNLI-VE/Flickr30K and COCO produce different thresholds because their similarity distributions and conditional success probabilities differ. The thresholds should therefore be re-estimated using data representative of each deployment rather than transferred unchanged across datasets.

Where does the cost reduction come from? To understand why EACache achieves lower realized cost, Figure 5 reorganizes the same outcomes according to three mutually exclusive ways of serving a query on SNLI-VE/Flickr30K dataset. For a total of N incoming queries, $N_1 + N_2 + N_3 = N$. N_1 denotes queries for which no cached result is returned and a new image is generated; N_2 denotes cached results that satisfy the user; and N_3 denotes cached results that fail to satisfy the user and therefore require recovery, regeneration, and another transmission.

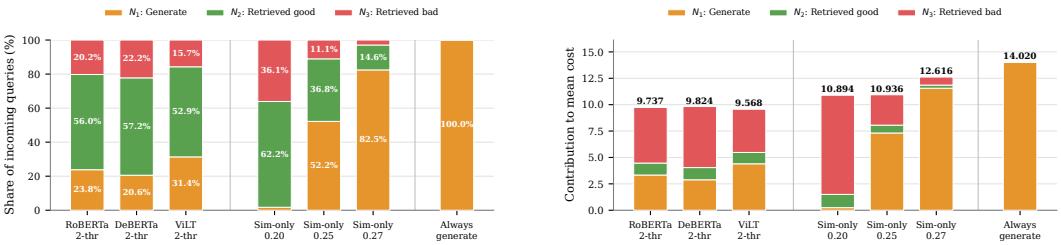

Figure 5: Query-outcome and cost decomposition over 4,000 incoming queries. **Left:** proportion of queries assigned to each outcome. **Right:** contribution of each outcome to the realized mean cost. E-verification balances the generation burden of conservative similarity thresholds against the recovery burden of permissive thresholds.

Table 4: Model precision in the middle similarity region. Direct high-similarity returns are excluded.

Model	Pairs sent to e-verification	Model accepted	User accepted	User rejected	Precision
RoBERTa	1,191	239	209	30	87.45%
DeBERTa	955	132	123	9	93.18%
ViLT	3,326	2,119	1,590	529	75.04%

The similarity-only policies reveal two different sources of system cost. At the permissive threshold of 0.20, only a small fraction of queries require immediate generation, but 36.10% of queries fall into the costly N_3 outcome. As a result, recovery from incorrect cache returns contributes substantially to its mean cost of 10.894. Increasing the similarity threshold suppresses N_3, but shifts many queries into N_1: at threshold 0.25, 52.20% of queries require generation, while at threshold 0.27 this increases to 82.45%. Consequently, their mean costs remain 10.936 and 12.616, respectively.

EACache balances these two failure modes. Its two-threshold policy avoids sending every low-confidence candidate to the user, reducing the recovery burden associated with permissive similarity thresholds, while e-verification allows intermediate candidates to be reused rather than immediately regenerated. The resulting policies therefore avoid both excessive recovery cost and excessive generation cost, which explains their lower overall realized cost.

Which entailment model should be used? No entailment model dominates across all settings. On SNLI-VE/Flickr30K, Table 4 shows that DeBERTa has the highest intermediate-region precision at 93.18%, followed by RoBERTa at 87.45% and ViLT at 75.04%. Nevertheless, ViLT-based EACache achieves the lowest realized cost of 9.568. On COCO, the ordering changes: DeBERTa achieves the lowest cost of 7.679, followed by RoBERTa at 7.693 and ViLT at 7.805, as shown in Table 2.

This variation occurs because model choice jointly affects acceptance frequency, accepted-result reliability, verification cost, and avoided generation. A precise but conservative verifier may trigger unnecessary generation, while a permissive verifier may reduce generation but cause more costly bad returns.

EACache captures this trade-off through $R_{V,m}(s)$, which combines the model-specific acceptance probability $e_m(s)$, reliability $r_m(s)$, verification cost $C_{e,m}$, and downstream generation and recovery costs. Verifier selection should therefore be based on end-to-end cost using representative data, rather than standalone precision or a single verifier ranking transferred across datasets.

4.3 Why Does E-Verification Recover Cache Hits Missed by Similarity?

The previous subsection evaluates EACache at the system level and shows that selective e-verification reduces realized cost by balancing generation and recovery. We next isolate the semantic mechanism behind this improvement: whether entailment models can recover valid image–query pairs that are rejected by similarity filtering.

Data construction. We separately construct controlled validation and test sets from SNLI-VE. For each split, we retain 1,000 Flickr30K images and select one complete premise group per image, containing one human-written premise and its entailment, neutral, and contradiction hypotheses. We repeat the label-independent selection using ten deterministic seeds while keeping the image set fixed.

Table 5: Test performance at three similarity thresholds selected independently on the validation split for each seed. Results are mean ± standard deviation over ten seeds. The entailment verifiers are applied to candidates rejected by similarity threshold. "Accepted" is the total number of pairs accepted by the resulting pipeline, while TP denotes correctly accepted gold-entailment pairs. Bold indicates the highest displayed mean within each similarity threshold; ties after rounding are both highlighted.

similarity threshold	Method	Threshold τ	Accepted	TP	Accuracy	Precision	Recall	F1
Best F1	similarity only	0.1850 ± 0.0031	1294.5 ± 48.5	881.8 ± 23.0	0.7337 ± 0.0074	0.6815 ± 0.0105	0.8801 ± 0.0232	0.7679 ± 0.0068
	similarity + RoBERTa		1407.4 ± 29.5	$\mathbf{992.9 \pm 3.9}$	$\mathbf{0.7883 \pm 0.0131}$	$\mathbf{0.7057 \pm 0.0138}$	$\mathbf{0.9909 \pm 0.0027}$	$\mathbf{0.8243 \pm 0.0087}$
	similarity + DeBERTa		1401.0 ± 30.4	986.7 ± 5.4	0.7853 ± 0.0121	0.7045 ± 0.0132	0.9847 ± 0.0051	0.8213 ± 0.0078
	similarity + ViLT		$\mathbf{1426.5 \pm 31.3}$	981.5 ± 5.3	0.7673 ± 0.0133	0.6883 ± 0.0135	0.9795 ± 0.0049	0.8084 ± 0.0085
Best accuracy	similarity only	0.1973 ± 0.0033	1127.2 ± 46.3	805.9 ± 25.5	0.7414 ± 0.0085	0.7153 ± 0.0113	0.8043 ± 0.0264	0.7569 ± 0.0103
	similarity + RoBERTa		1313.3 ± 22.9	$\mathbf{989.3 \pm 3.7}$	$\mathbf{0.8317 \pm 0.0118}$	$\mathbf{0.7535 \pm 0.0139}$	$\mathbf{0.9873 \pm 0.0024}$	$\mathbf{0.8546 \pm 0.0088}$
	similarity + DeBERTa		1303.4 ± 26.0	979.0 ± 5.6	0.8264 ± 0.0107	0.7513 ± 0.0136	0.9771 ± 0.0058	0.8494 ± 0.0078
	similarity + ViLT		$\mathbf{1334.8 \pm 21.3}$	969.7 ± 4.5	0.8014 ± 0.0115	0.7267 ± 0.0127	0.9678 ± 0.0043	0.8300 ± 0.0084
Best precision (≥ 100 accepted)	similarity only	0.2784 ± 0.0080	134.3 ± 66.9	117.8 ± 54.4	0.5498 ± 0.0209	0.8845 ± 0.0272	0.1175 ± 0.0541	0.2034 ± 0.0734
	similarity + RoBERTa		958.4 ± 16.7	$\mathbf{934.6 \pm 6.4}$	$\mathbf{0.9544 \pm 0.0053}$	$\mathbf{0.9753 \pm 0.0123}$	$\mathbf{0.9327 \pm 0.0047}$	$\mathbf{0.9535 \pm 0.0050}$
	similarity + DeBERTa		937.9 ± 18.7	914.5 ± 9.1	0.9446 ± 0.0060	$\mathbf{0.9753 \pm 0.0132}$	0.9127 ± 0.0072	0.9429 ± 0.0059
	similarity + ViLT		$\mathbf{977.3 \pm 28.2}$	856.9 ± 16.8	0.8673 ± 0.0059	0.8770 ± 0.0112	0.8552 ± 0.0163	0.8658 ± 0.0066

Table 6: Controlled cache-hit performance at the validation-selected best-precision similarity threshold. Gold entailment denotes a valid cache-hit opportunity, while gold contradiction denotes an invalid candidate. Results are mean ± sample standard deviation across ten seed-specific test selections.

Method	Hit decision	Valid hit	False hit	Hit precision	Hit recall
similarity only	6.71 ± 3.34	5.89 ± 2.71	$\mathbf{0.82 \pm 0.64}$	88.45 ± 2.72	11.75 ± 5.41
similarity + RoBERTa	47.90 ± 0.80	$\mathbf{46.71 \pm 0.26}$	1.19 ± 0.63	$\mathbf{97.53 \pm 1.23}$	$\mathbf{93.27 \pm 0.47}$
similarity + DeBERTa	46.88 ± 0.89	45.71 ± 0.40	1.17 ± 0.66	$\mathbf{97.53 \pm 1.32}$	91.27 ± 0.72
similarity + ViLT	$\mathbf{48.85 \pm 1.41}$	42.83 ± 0.82	6.02 ± 0.71	87.70 ± 1.12	85.52 ± 1.63

All values are percentages. Bold indicates the best mean in each column; for false-hit rate, lower is better.

After removing neutral examples, each split contains approximately 2,000 image–hypothesis pairs, with entailment treated as the positive class.

For each seed, similarity thresholds are selected on the validation split and fixed for the corresponding test split. RoBERTa, DeBERTa and ViLT are then applied only to candidates rejected by similarity thresholds. This setup directly measures whether e-verification can recover entailment pairs that would otherwise be discarded by similarity-based filtering. Results are reported as the mean and standard deviation across the ten seeds.

E-verification recovers valid pairs rejected by similarity. Table 5 shows that e-verification consistently recovers substantially more valid pairs than similarity alone at the best-F1, best-accuracy, and best-precision operating points. Focusing on the high-precision setting, Table 6 shows that similarity alone returns only 6.71% of candidates, yielding a 5.89% valid-hit rate and 11.75% recall. E-verification increases the cache-return rate to 46.88–48.85% and the valid-hit rate to 42.83–46.71%. RoBERTa and DeBERTa provide the strongest reliability–recovery balance: both attain 97.53% hit precision, recall above 91%, and false-hit rates near 1.2%. Thus, the text-based entailment models improve both coverage and precision relative to similarity alone. ViLT makes the most hit decisions (48.85%) but is more permissive, achieving 87.70% precision and a 6.02% false-hit rate, consistent with its behavior in our earlier experiments. These results demonstrate the central mechanism of EACache: e-verification recovers many semantically valid cache hits discarded by conservative similarity filtering, with a model-dependent trade-off between recovery and reliability.

5 Conclusion

We presented EACache, an entailment-aware caching framework that combines similarity retrieval with selective e-verification through a cost-aware two-threshold policy. EACache reduces provider-side cost while preserving user satisfaction through recovery from failed cache returns, and recovers valid cache hits missed by conservative similarity filtering. Results also show that entailment-model selection should consider end-to-end system cost, rather than precision alone.

AI USE STATEMENT

Generative AI tools were used to assist with draft polishing and code preparation. GPT-5.2 was additionally used as a multimodal satisfaction judge in the experiments, as described in the experimental setup and appendix. All experimental results, numerical values, analyses, and conclusions were finished and verified by the authors.

ETHICS STATEMENT

The study uses an existing visual-entailment benchmark and does not involve new human-subject data collection.

REPRODUCIBILITY STATEMENT

The submission describes the benchmark composition, bin boundaries, probability definitions, cost equations, model roles, and cost settings used to obtain the reported results. The supplementary material will include the notebooks and scripts for pair construction, entailment and satisfaction annotation, probability fitting, threshold calculation, bootstrap analysis, and figure generation, together with non-sensitive intermediate tables.

REFERENCES

Fu Bang. GPTCache: An open-source semantic cache for LLM applications enabling faster answers and cost savings. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software*, pp. 212–218, 2023. doi: 10.18653/v1/2023.nlposs-1.24.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA-02: A visual representation for neon genesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 552–562, 2023.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

Amita Kamath, Cheng-Yu Hsieh, Kai-Wei Chang, and Ranjay Krishna. The hard positive truth about vision-language compositionality. In *ECCV*, 2024.

Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5583–5594. PMLR, 2021. URL https://proceedings.mlr.press/v139/kim21k.html.

Darina Koishigarina, Arnas Uselis, and Seong Joon Oh. CLIP behaves like a bag-of-words model cross-modally but not uni-modally. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DldwXCCP25.

Jiaxing Li, Chi Xu, Feng Wang, Isaac M von Riedemann, Cong Zhang, and Jiangchuan Liu. Scalm: Towards semantic caching for automated chat services with large language models. In *2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, pp. 1–10, 2024. doi: 10.1109/IWQoS61813.2024.10682957.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

OpenAI. GPT-5.2 model. https://developers.openai.com/api/docs/models/gpt-5.2, 2025. Accessed: 2026-09-25.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763, 2021.

Luis Gaspar Schroeder, Aditya Desai, Alejandro Cuadron, Kyle Chu, Shu Liu, Mark Zhao, Stephan Krusche, Alfons Kemper, Matei Zaharia, and Joseph E. Gonzalez. vCache: Verified semantic prompt caching. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2502.03771.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. In *International Conference on Learning Representations*, 2016.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL https://aclanthology.org/Q14-1006/.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KRLUvxh8uaX.

A APPENDIX

A.1 SATISFACTION-JUDGE PROMPT

To evaluate whether a retrieved cached image can satisfactorily serve an incoming user query, we use GPT-5.2 as a fixed multimodal satisfaction judge. The model receives the retrieved image together with the user query and produces a semantic alignment score from 0 to 100. In the main experiments, scores greater than 50 are treated as satisfactory cache returns. The exact system prompt is shown below.

System Prompt for Cache-Return Satisfaction Evaluation

Role.
You are an expert human evaluator trained to judge whether an image satisfies the meaning of a text query. Your goal is to provide a stable, human-like semantic judgment of whether the image can serve the user's request.

Instructions.

1. Read the text query carefully. It specifies the visual content requested by the user.
2. Examine the image and consider only what is clearly visible.
3. Compare the query with the visible content and decide whether the image:
   - **ENTAILS** the query: the requested content is fully supported by the image;
   - is **NEUTRAL**: the requested content is uncertain or only partially supported; or
   - **CONTRADICTS** the query: the image conflicts with or fails to support essential requested content.
4. Ignore irrelevant additional content in the image. Extra visual information does **not** reduce the match as long as the requested content is satisfied.
5. Be strict about visual evidence. Do not infer or assume content that is not visibly supported.
6. Provide a short explanation of the judgment.
7. Output a final alignment score from 0 to 100:
   - **0–20:** clear contradiction
   - **21–49:** mostly unsupported
   - **50:** neutral or uncertain
   - **51–79:** partially supported
   - **80–100:** strongly supported

Label Mapping.

Judgment	JSON label
ENTAILS	`"match"`
NEUTRAL	`"partial_match"` or `"uncertain"`
CONTRADICTS	`"not_match"`

Output Requirement.
Keep the output faithful to visible evidence only. Return **STRICT JSON only**.