UGGRH: Unsupervised Generative Completion and Graph-attention Refinement for Incomplete Cross-modal Hashing

Yunfei Chen[1], **Yuchen Zhang**[2] , **Hongyu Lin**[3] , **Peng Liu**[2] , **Zhan Yang**[2*]

[1]Information Support Force Engineering University, Wuhan, China
[2]Central South University, Changsha, China
[3]Xiamen University, Xiamen, China
yunfeichen_csu@126.com, Monica@csu.edu.cn, hongyulin@stu.xmu.edu.cn,
8208241309@csu.edu.cn, zyang22@csu.edu.cn

Abstract

The proliferation of multimodal data has underscored the imperative for efficiency in cross-modal retrieval. By mapping data into a unified Hamming space, Cross-modal hashing (CMH) significantly enhances both storage and computational efficiency, thereby establishing itself as a preferred favored paradigm for retrieval tasks. However most methods assume complete data, which rarely holds in practice. In incomplete settings, missing modalities (e.g., text or image) break semantic correspondence and make unsupervised hashing unreliable. Therefore, we propose Unsupervised Generative Completion and Graph-attention Refinement for Incomplete Cross-modal Hashing (UGGRH) method. Specifically, UGGRH leverages pretrained generative models for modality completion and employs a discriminator-guided similarity refinement module to fuse intra-modal relations with paired image-text matching signals, thereby constructing a robust cross-modal similarity matrix that yields a noise-tolerant similarity target for training. After that, it learns discriminative binary codes via hash-aware graph attention and cross-modal fusion, to refine modality embeddings and produce unified hash codes. Experiments on MIRFlickr-25K, NUS-WIDE, and MS COCO datasets demonstrate that UGGRH consistently achieves superior performance under high missing rates and modality imbalance, showing significant improvements in retrieval accuracy and robustness across challenging incomplete scenarios.

1 Introduction

Cross-modal retrieval searches semantically relevant instances in one modality using a query from another, such as image-to-text and text-to-image retrieval [10, 16, 17, 26]. Cross-modal hashing improves efficiency by mapping heterogeneous data into compact binary codes for Hamming ranking [5, 13, 24, 32]. Deep cross-modal hashing methods improve semantic alignment through end-to-end representation and hash-code learning, while recent supervised and weakly supervised approaches introduce adaptive constraints for imbalance and label noise [5, 24]. Target-label-free methods such as UDDH [37] and UCCH [8] reduce dependence on semantic labels through contrastive and dual-branch objectives. Yet most methods still assume fully paired training data and degrade when images or texts are missing, noisy, or only partially paired [23, 28, 31].

This work studies incomplete cross-modal hashing, where the training set contains a mixture of paired image-text samples and unpaired samples with a missing image or missing text [23, 25,

*Corresponding author.

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

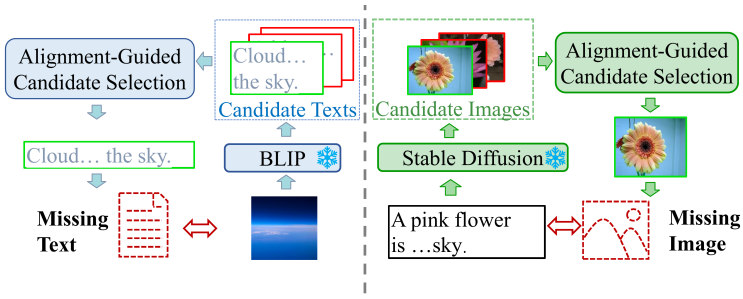


Figure 1: Offline modality completion with frozen generators and dual-encoder filtering. For each incomplete sample, the observed modality conditions candidate generation and also serves as the anchor for alignment-based re-ranking; only the selected candidate feature is cached for hash learning, rather than being treated as ground truth.

39]. The goal is to learn binary codes for both image-to-text and text-to-image retrieval using only observed target-dataset pairs and self-supervised signals from cross-modal matching and intra-modal structure. Existing incomplete methods such as PMCH [39] and CICH [23] can still suffer from inaccurate completion and unstable similarity estimation under high missing rates, motivating the candidate generation and alignment filtering strategy in Fig. 1 [4, 7, 29].

To address these challenges, this paper proposes Unsupervised Generative Completion and Graph-attention Refinement (UGGRH), a target-label-free framework for incomplete cross-modal hashing. Pretrained foundation models are frozen feature and completion modules, and all compared methods use the same fixed pretrained features for fair comparison. UGGRH completes missing modalities offline via candidate generation and frozen dual-encoder filtering, refines cross-modal similarity into a self-supervised target, and learns unified binary hash codes through hash-aware graph attention and cross-modal fusion. The main contributions are as follows:

(1) A modality completion strategy based on large-scale pretrained models is introduced to alleviate missing-modality issues, reduce semantic drift, and decouple expensive generation from hashing training.

(2) A semantics-enhanced cross-modal similarity refinement module is developed to produce a noise-tolerant similarity target, stabilizing unsupervised learning under incomplete and imperfectly completed data.

(3) A hash-aware graph attention network is designed to refine representations and learn unified, discriminative binary codes, improving retrieval robustness under modality imbalance.

Together, these designs couple semantic completion, affinity construction, and hash-space propagation for incomplete cross-modal hashing rather than simply concatenating standard modules.

2 Related Work

2.1 Complete Cross-Modal Hashing Methods

Complete methods assume fully observed modalities and learn cross-modal hashing representations under supervised or unsupervised settings. Early shallow methods rely on handcrafted descriptors and separate projection learning, limiting high-level semantic modeling [11, 18, 20, 22]. Recent deep supervised methods jointly optimize feature extraction and hash-code learning; representative works such as LtCMH [5] and AMSH [24] address long-tail distributions and weak supervision through adaptive fusion or boundary-aware constraints. Unsupervised complete methods reduce label dependency by capturing latent cross-modal structures, as in UDDH [37] and UCCH [8]. These methods, however, usually assume complete paired training data.

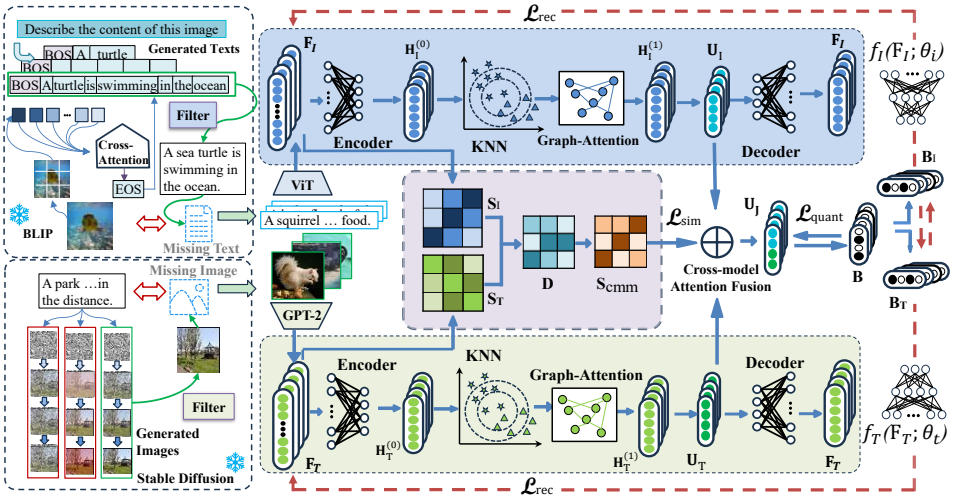


Figure 2: Overall framework of UGGRH. We first perform one-shot offline modality completion to obtain completed features, then refine cross-modal similarity to form a noise-tolerant target, and finally apply HashGAT to propagate information on modality-specific graphs and generate unified binary hash codes.

2.2 Incomplete Cross-Modal Hashing Methods

Incomplete cross-modal hashing targets missing or partially paired modalities while preserving semantic consistency and discriminative hash representations. Prototype-based methods such as PMCH [39] recover missing modalities by aggregating intra-modality prototypes, reconstruction-based alternatives use lightweight networks or graph autoencoders [28, 31], and contrastive methods such as CICH [23] and SICH [25] combine clustering or pseudo-label mining with hash learning. AGADH [38] uses mask-derived placeholders and a GAT encoder-decoder with structural alignment for modality reconstruction, then combines intra- and cross-modal similarities to construct an adjacency matrix and adaptively fuse modality features. These families address different aspects of incompleteness: prototype and reconstruction methods focus on imputing missing features, contrastive methods strengthen self-supervision with mined structure, and graph-based refinements propagate local neighborhood cues. However, most do not explicitly decouple candidate generation from hash training or use frozen cross-modal alignment to reject semantically drifted completions before similarity construction. Under severe incompleteness, asymmetric modality distributions, inaccurate recovery, and unstable alignment still weaken retrieval, motivating stronger completion with alignment-aware filtering, similarity refinement, and graph-based hashing.

3 Methodology

As illustrated in Fig. 2, UGGRH consists of three modules: pretrained-model-based modality completion, semantics-enhanced cross-modal similarity refinement, and hash-aware graph attention.

3.1 Notations and Problem Setup

Bold uppercase/lowercase symbols denote matrices/vectors, and $\|\cdot\|_F$, $\|\cdot\|_2$, and $\|\cdot\|_1$ denote Frobenius, ℓ_2, and ℓ_1 norms. Let $\mathcal{D} = \{(\mathbf{x}_I^i, \mathbf{x}_T^i)\}_{i=1}^N$ contain N image-text pairs, where $\mathcal{I}_{\text{miss-t}}$ and $\mathcal{I}_{\text{miss-i}}$ index samples with missing text and image, respectively. After completion and encoding, $\mathbf{F}_I, \mathbf{F}_T \in \mathbb{R}^{N\times d}$ store image/text features, and k is the hash length. The goal is to learn hashing functions that produce unified binary codes $\mathbf{B} \in \{-1, +1\}^{N\times k}$ while preserving cross-modal similarity.

3.2 Modality Completion via Large-Scale Pretrained Models

Vision-Driven Caption Generation. For samples $i \in \mathcal{I}_{\text{miss-t}}$, a pretrained vision–language captioning model (BLIP [17]) generates M_T candidate captions $\mathcal{C}_T^i$ from the available image $\mathbf{x}_I^i$ via stochastic sampling with fixed random seeds for reproducibility, and the caption that maximizes dual-encoder alignment is selected:

$$\hat{\mathbf{x}}_T^i = \arg\max_{\mathbf{x} \in \mathcal{C}_T^i} \cos\big(E_T(\mathbf{x}),\, E_I(\mathbf{x}_I^i)\big)\,. \tag{1}$$

The text feature is $\mathbf{f}_{T,i} = E_T(\mathbf{x}_T^i)$ if text is observed and $E_T(\hat{\mathbf{x}}_T^i)$ otherwise, where $E_I(\cdot)$ and $E_T(\cdot)$ are frozen CLIP image and text encoders [26].

Text-Guided Image Synthesis. For samples $i \in \mathcal{I}_{\text{miss-i}}$, a large-scale text-to-image diffusion model (Stable Diffusion, SD [27]) synthesizes M_I candidate images $\mathcal{C}_I^i$ conditioned on the available text $\mathbf{x}_T^i$. Candidates are generated via stochastic sampling with fixed random seeds for reproducibility. The candidate that maximizes dual-encoder alignment is then retained:

$$\hat{\mathbf{x}}_I^i = \arg\max_{\mathbf{x} \in \mathcal{C}_I^i} \cos\big(E_I(\mathbf{x}),\, E_T(\mathbf{x}_T^i)\big)\,. \tag{2}$$

The image feature is $\mathbf{f}_{I,i} = E_I(\mathbf{x}_I^i)$ if the image is observed and $E_I(\hat{\mathbf{x}}_I^i)$ otherwise.

Alignment Filtering and Efficiency. For each missing modality, a small candidate set is sampled, filtered for validity and safety, and re-ranked by frozen dual-encoder alignment. Only the top candidate is kept to mitigate semantic drift, and the completion is used only to obtain $\mathbf{f}_{m,i}$ rather than as ground truth. Generation is performed once offline; training and online retrieval use cached $\mathbf{F}_I, \mathbf{F}_T$ and do not invoke diffusion models. Key cost factors, including candidate counts, batching, and parallelization, are analyzed in the appendix.

3.3 Semantics-Enhanced Cross-Modal Similarity Refinement

Given completed features, row-wise ℓ_2 normalization gives $\tilde{\mathbf{f}}_{m,i} = \mathbf{f}_{m,i}/\|\mathbf{f}_{m,i}\|_2$ for $m \in \{I, T\}$. Let $\tilde{\mathbf{F}}_m$ stack normalized features; intra-modal similarities are $\mathbf{S}_m = \tilde{\mathbf{F}}_m \tilde{\mathbf{F}}_m^\top$, with affinity-scaled forms $\bar{\mathbf{S}}_m = (\mathbf{S}_m + \mathbf{1}\mathbf{1}^\top)/2$. To inject cross-modal cues, a lightweight image–text matching MLP g_θ is trained with binary cross-entropy by using observed pairs as positives and randomly mismatched pairs as negatives, without semantic labels or query/retrieval test pairs. Unlike direct CLIP cosine similarity, this scorer adapts the cross-modal matching signal to observed target-domain pairs. It is fixed during HashGAT training and forms $\mathbf{D}_{ij} = g_\theta([\tilde{\mathbf{f}}_{I,i}; \tilde{\mathbf{f}}_{T,j}]) \in [0,1]$. Since $\mathbf{S}_I, \mathbf{S}_T$ are cosine similarities and $\mathbf{D}$ is a normalized cross-modal affinity score, cosine scores are first mapped to a nonnegative affinity scale. The unified similarity is obtained by fusing intra-modal relations with discriminator-based cross-modal affinities: $\mathbf{S}_1 = \alpha\big(a_1 \bar{\mathbf{S}}_I + (1-a_1)\bar{\mathbf{S}}_T\big) + (1-\alpha)\mathbf{D}$. Here, $a_1, \alpha \in [0,1]$ are fusion weights. To suppress noisy connections and reduce graph complexity, row-wise top-K_s sparsification is applied:

$$(\mathbf{S}_1^{(K_s)})_{ij} = \begin{cases} (\mathbf{S}_1)_{ij}, & j \in \mathrm{TopK}(\mathbf{S}_1(i,:), K_s), \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $\mathrm{TopK}(\cdot, K_s)$ returns the K_s largest entries in a row. Row-wise sparsification may be asymmetric and is later symmetrized. Non-linear smoothing with a residual term is then performed, followed by symmetry enforcement:

$$\begin{aligned} \mathbf{S}_2 &= \lambda_s\big(2\tanh(\mathbf{S}_1^{(K_s)}) + \mathbf{I}\big) + (1-\lambda_s)\mathbf{S}_1^{(K_s)}, \\ \mathbf{S} &= a_2 \mathbf{S}_1^{(K_s)} + (1-a_2)\mathbf{S}_2,\ \ \mathbf{S}_{\text{cmm}} = \tfrac{1}{2}\big(\mathbf{S} + \mathbf{S}^\top\big), \end{aligned} \tag{4}$$

where $\lambda_s, a_1, a_2, \alpha \in [0,1]$. $\mathbf{S}_{\text{cmm}}$ is precomputed once and $\mathbf{S}_{\text{cmm}}[\mathcal{B}, \mathcal{B}]$ supervises mini-batch training.

3.4 Hash-Aware Graph Attention Network

Encoding. Hash-aware Graph Attention Network (HashGAT) is a hashing-oriented graph-attention network that refines completed features by message passing on two modality-specific graphs and

Algorithm 1 UGGRH Training Pipeline

Input: incomplete pairs $\mathcal{D}$; hash length k; neighbor size K; loss weights γ, η, β; epochs E; learning rate ξ
Output: Optimized parameters Θ^*
Initialize parameters Θ
Obtain completed features $\mathbf{F}_I, \mathbf{F}_T$ via $\mathrm{CompleteAndEncode}(\mathcal{D})$ (*offline*)
Compute the refined similarity target $\mathbf{S}_{\mathrm{cmm}}$ via $\mathrm{EnhanceSim}(\mathbf{F}_I, \mathbf{F}_T; g_\theta)$ (g_θ fixed)
for $e = 1, \ldots, E$ **do**
  Encode modalities to obtain $\mathbf{h}_I^{(0)}$ and $\mathbf{h}_T^{(0)}$
  Construct KNN graphs $\mathbf{A}_I$ and $\mathbf{A}_T$
  Compute hash-conditioned edge weights w_I and w_T
  Propagate on graphs to obtain $\mathbf{h}_I^{(1)}$ and $\mathbf{h}_T^{(1)}$
  Fuse modalities to obtain $\mathbf{u}$ and binarize $\mathbf{B} = \mathrm{sign}(\tanh(\mathbf{u}))$
  Compute $\mathcal{L} = \gamma\mathcal{L}_{\text{rec}} + \eta\mathcal{L}_{\text{sim}} + \beta\mathcal{L}_{\text{quant}}$
  Update Θ by gradient descent on $\mathcal{L}$ with step size ξ
end for
return Θ

outputs unified binary codes. Given completed features, modality embeddings are computed with learnable encoders and ℓ_2 normalization as $\mathbf{h}_{m,i}^{(0)} = \phi_m(\mathbf{f}_{m,i})/\|\phi_m(\mathbf{f}_{m,i})\|_2$ for $m \in \{I, T\}$.

Dynamic Graph Construction. A sparse feature-guided K-Nearest Neighbor algorithm (KNN) graph is built for each modality. For each modality, define $s_{ij}^m = \cos\left(\mathbf{h}_{m,i}^{(0)}, \mathbf{h}_{m,j}^{(0)}\right)$ and $\mathcal{N}_{m,i} = \mathrm{TopK}(s_{i:}^m, K)$. The adjacency entries are:

$$A_{m,ij} = \begin{cases} 1, & j \in \mathcal{N}_{m,i}, \\ 0, & \text{otherwise}, \end{cases} \qquad m \in \{I, T\}, \tag{5}$$

$\mathbf{A}_m$ is symmetrized and augmented with self-loops. Since $\mathbf{h}_m^{(0)}$ evolves during training, neighborhoods are updated dynamically.

Graph Attention Propagation. Because $\mathrm{sign}(\cdot)$ is non-differentiable, we inject hash-conditioned information into attention through a continuous hash proxy on each modality graph. For each node, a continuous hash proxy is predicted as $\mathbf{z}_{m,i} = \tanh(\mathrm{MLP}_e(\mathbf{h}_{m,i}^{(0)}))$, and the hash-conditioned edge weight is $w_{m,ij} = \exp(-\|\mathbf{z}_{m,i} - \mathbf{z}_{m,j}\|_2/\tau_h)$. $\log(w_{m,ij} + \epsilon)$ is injected into the attention logits:

$$\begin{aligned} e_{m,ij} = \mathrm{LeakyReLU}\Big(\mathbf{a}_m^\top [\mathbf{W}_m \mathbf{h}_{m,i}^{(0)}; \mathbf{W}_m \mathbf{h}_{m,j}^{(0)}]\Big) \\ + \log(w_{m,ij} + \epsilon)\,, \end{aligned} \tag{6}$$

where ϵ is a small constant for numerical stability. The normalized attention coefficient is $\alpha_{m,ij} = \mathrm{softmax}_{j \in \mathcal{N}_{m,i}}(e_{m,ij})$. Neighbor messages are aggregated as $\mathbf{h}_{m,i}^{(1)} = \sum_{j \in \mathcal{N}_{m,i}} \alpha_{m,ij} \mathbf{W}_m \mathbf{h}_{m,j}^{(0)}$, and experiments use M-head attention with residual connections and normalization.

Cross-Modal Fusion. Image-specific, text-specific, and joint logits are computed as $\mathbf{u}_{i,I} = \psi_I(\mathbf{h}_{I,i}^{(1)})$, $\mathbf{u}_{i,T} = \psi_T(\mathbf{h}_{T,i}^{(1)})$, and $\mathbf{u}_{i,J} = \psi_J([\mathbf{h}_{I,i}^{(1)}; \mathbf{h}_{T,i}^{(1)}])$, where the projection heads produce k-bit logits from modality-specific and fused embeddings. A softmax gate $\boldsymbol{\pi}_i = \mathrm{softmax}(\mathrm{MLP}([\mathbf{u}_{i,I}; \mathbf{u}_{i,T}; \mathbf{u}_{i,J}]))$ adaptively weighs the three sources and fuses them as $\mathbf{u}_i = \sum_{s \in \{I,T,J\}} \pi_{i,s} \mathbf{u}_{i,s}$. Continuous and binary codes are $\mathbf{H}_i = \tanh(\mathbf{u}_i)$ and $\mathbf{b}_i = \mathrm{sign}(\mathbf{H}_i)$. Stacking all samples yields $\mathbf{H} \in (-1, 1)^{N \times k}$ and $\mathbf{B} \in \{-1, +1\}^{N \times k}$.

Objective Function. For mini-batch $\mathcal{B}$, the target is $\mathbf{S}_\mathcal{B} = \mathbf{S}_{\mathrm{cmm}}[\mathcal{B}, \mathcal{B}]$ with $N_b = |\mathcal{B}|$. Let $\mathbf{H}_q$ ($q \in \{I, T, J\}$) denote branch codes and $\mathbf{H}$ the fused code. Two-layer decoders reconstruct modality features as $\hat{\mathbf{F}}_I = \mathrm{Dec}_I(\mathbf{H})$ and $\hat{\mathbf{F}}_T = \mathrm{Dec}_T(\mathbf{H})$. The reconstruction loss is defined as:

$$\mathcal{L}_{\text{rec}} = \|\hat{\mathbf{F}}_I - \mathbf{F}_I\|_F^2 + \|\hat{\mathbf{F}}_T - \mathbf{F}_T\|_F^2. \tag{7}$$

This term encourages the fused code to retain modality information from both image and text features. To preserve the refined neighborhood structure, the similarities induced by each branch code

are matched to $\mathbf{S}_{\mathcal{B}}$:

$$\mathcal{L}_{\text{sim}} = \sum_{q \in \{I,T,J\}} \|\mathbf{S}_{\mathcal{B}} - \text{sim}(\mathbf{H}_q)\|_F^2 , \tag{8}$$

where $\text{sim}(\mathbf{X}) = \bar{\mathbf{X}}\bar{\mathbf{X}}^\top$ with $\bar{\mathbf{X}}_i = \mathbf{X}_i / \|\mathbf{X}_i\|_2$. To reduce binarization error:

$$\mathcal{L}_{\text{quant}} = \frac{1}{N_b} \sum_{i=1}^{N_b} \| |\mathbf{H}_i| - \mathbf{1} \|_1 . \tag{9}$$

The quantization term pushes continuous codes toward ± 1, stabilizing the final sign operation for binary retrieval. The overall training objective is:

$$\mathcal{L} = \gamma \mathcal{L}_{\text{rec}} + \eta \mathcal{L}_{\text{sim}} + \beta \mathcal{L}_{\text{quant}}, \tag{10}$$

where $\gamma, \eta, \beta \geq 0$ are loss weights.

4 Experiments

Table 1: Partial Data Ratio (PDR) settings, reported as (complete, image-missing, text-missing).

Setting	Complete Samples	Image Missing	Text Missing
easy	50%	25%	25%
medium	30%	35%	35%
hard	10%	45%	45%

4.1 Experimental Settings

Datasets. Experiments are conducted on MIRFlickr-25K [9], MS COCO [19], and NUS-WIDE [3]. Each dataset is split into query, training, and retrieval sets. To emulate incomplete multimodal conditions, the Partial Data Ratio (PDR) protocol [39] randomly removes either image or text from part of the training set; the missing modality is then reconstructed offline, and all observed/reconstructed modalities are embedded by the same frozen CLIP dual encoder [26, 10]. Table 1 summarizes the symmetric PDR settings, where image-missing and text-missing ratios are equal.

Implementation. The implementation uses PyTorch 2.1.0, Python 3.10, Ubuntu 22.04, and CUDA 12.1 on a single NVIDIA RTX 4090 GPU. Training uses AdamW [21] with learning rate 10^{-3}, weight decay 10^{-2}, batch size 512, 60 epochs, mixed precision, 4-step gradient accumulation, max-norm clipping at 1.0, and plateau-based learning-rate decay.

Evaluation Metric. The tasks are image-to-text (I→T) and text-to-image (T→I). Mean Average Precision under Hamming ranking is the primary metric, with mAP@50 reported by default; Precision@K, Recall@K, and PR curves are also evaluated. Unless specified otherwise, $k \in \{16, 32, 64, 128\}$.

4.2 Comparison with Incomplete Hashing Methods

Baselines. Comparisons include up to 12 representative baselines, depending on dataset availability: shallow hashing methods DCH [34] and JIMFH [33]; deep hashing methods DCMH [12], SSAH [15], AGAH [6], DADH [1], and DCHMT [32]; real-valued incomplete retrieval methods MCCN [35] and PAN [36]; and recent incomplete hashing methods DSCMH [30], PMCH [39], and EGICH [2]. All methods use the same frozen CLIP features and identical splits. MCCN and PAN are implemented following the original papers; available implementations are used for the others.

mAP Analysis. Table 2 reports mAP@50 results with 32-bit codes under different PDR settings and shows that UGGRH achieves the best performance across all datasets and both retrieval directions. On MIRFlickr-25K, UGGRH improves the strongest baseline in the hard setting by 4.3 percentage points in mean mAP@50, with a larger gain on the more challenging T→I direction (0.901 compared with 0.822); on MS COCO, it attains mean mAP@50 values of 0.921, 0.923, and 0.926 in the hard, medium, and easy settings, respectively, and the hard-setting margin reaches 5.0 percentage points.

Table 2: mAP@50 comparisons under different levels of modality incompleteness (32-bit codes). The triples in the column headers denote (complete, image-missing, text-missing) as in Table 1. Δ is the absolute mAP gap between easy and hard in percentage points. All methods use unified features. Red, Blue, Green denote best, second-best, third-best.

Datasets	Methods	hard(10,45,45)%			med(30,35,35)%			easy(50,25,25)%			Δ (easy-hard)(%)		
		I→T	T→I	mean	I→T	T→I	mean	I→T	T→I	mean	I→T	T→I	mean
MIRFlickr	DCH	0.597	0.622	0.610	0.713	0.683	0.698	0.733	0.692	0.713	13.60	7.00	10.30
	JIMFH	0.560	0.589	0.575	0.589	0.598	0.594	0.599	0.600	0.600	3.90	1.10	2.50
	DCMH	0.890	0.817	0.854	0.891	0.826	0.859	0.894	0.829	0.862	0.40	1.20	0.80
	SSAH	0.895	0.822	0.859	0.895	0.832	0.864	0.896	0.845	0.871	0.10	2.30	1.20
	AGAH	0.617	0.626	0.622	0.841	0.811	0.826	0.876	0.830	0.853	25.90	20.40	23.15
	DADH	0.876	0.814	0.856	0.883	0.828	0.859	0.884	0.835	0.864	0.80	2.10	1.45
	PAN	0.896	0.807	0.852	0.901	0.806	0.854	0.903	0.811	0.857	0.70	0.40	0.55
	DCHMT	0.860	0.802	0.832	0.862	0.805	0.835	0.864	0.809	0.835	0.40	0.70	0.55
	DSCMH	0.873	0.792	0.833	0.881	0.801	0.841	0.889	0.809	0.849	1.60	1.70	1.60
	PMCH	0.885	0.806	0.846	0.891	0.814	0.853	0.896	0.823	0.860	1.10	1.70	1.40
	EGICH	0.892	0.821	0.857	0.898	0.829	0.864	0.904	0.836	0.870	1.20	1.50	1.30
	UGGRH	0.903	0.901	0.902	0.905	0.904	0.905	0.907	0.906	0.907	0.40	0.50	0.45
MSCOCO	DCH	0.527	0.628	0.578	0.535	0.643	0.589	0.539	0.645	0.592	1.20	1.70	1.45
	JIMFH	0.414	0.461	0.438	0.462	0.485	0.474	0.479	0.492	0.486	6.50	3.10	4.80
	DCMH	0.681	0.710	0.696	0.689	0.718	0.704	0.695	0.724	0.710	1.40	1.40	1.40
	SSAH	0.613	0.623	0.618	0.615	0.638	0.627	0.625	0.645	0.635	1.20	2.20	1.70
	AGAH	0.576	0.600	0.588	0.595	0.618	0.607	0.621	0.634	0.628	4.51	3.40	3.96
	DADH	0.686	0.670	0.678	0.689	0.688	0.689	0.692	0.689	0.691	0.60	1.90	1.25
	MCCN	0.672	0.617	0.645	0.673	0.621	0.647	0.704	0.636	0.670	3.20	1.90	2.55
	DCHMT	0.656	0.631	0.644	0.662	0.636	0.649	0.668	0.649	0.659	1.20	1.80	1.50
	DSCMH	0.846	0.838	0.842	0.854	0.846	0.850	0.861	0.853	0.857	1.50	1.50	1.50
	PMCH	0.861	0.853	0.857	0.869	0.861	0.865	0.876	0.868	0.872	1.50	1.50	1.50
	EGICH	0.875	0.867	0.871	0.882	0.874	0.878	0.888	0.881	0.885	1.30	1.40	1.35
	UGGRH	0.922	0.920	0.921	0.923	0.922	0.923	0.926	0.925	0.926	0.40	0.50	0.45
NUSWIDE	DCH	0.507	0.585	0.546	0.531	0.599	0.565	0.553	0.598	0.576	4.60	1.30	2.95
	JIMFH	0.132	0.205	0.169	0.154	0.207	0.181	0.154	0.213	0.184	2.20	0.80	1.50
	DCMH	0.605	0.521	0.563	0.614	0.541	0.578	0.622	0.545	0.584	1.70	2.40	2.05
	SSAH	0.514	0.496	0.505	0.563	0.505	0.534	0.579	0.555	0.567	6.50	5.90	6.20
	AGAH	0.304	0.356	0.330	0.470	0.413	0.442	0.616	0.574	0.595	31.20	21.80	26.50
	DADH	0.569	0.588	0.579	0.598	0.594	0.596	0.610	0.613	0.612	4.10	2.50	3.30
	MCCN	0.563	0.537	0.550	0.588	0.550	0.569	0.587	0.555	0.571	2.40	1.80	2.10
	DCHMT	0.625	0.542	0.584	0.649	0.553	0.601	0.671	0.560	0.616	4.60	1.80	3.20
	DSCMH	0.759	0.748	0.754	0.768	0.756	0.762	0.776	0.764	0.770	1.70	1.60	1.60
	PMCH	0.776	0.764	0.770	0.784	0.772	0.778	0.793	0.780	0.787	1.70	1.60	1.70
	EGICH	0.789	0.777	0.783	0.797	0.785	0.791	0.805	0.793	0.799	1.60	1.60	1.60
	UGGRH	0.846	0.843	0.844	0.847	0.845	0.846	0.849	0.848	0.849	0.30	0.50	0.40

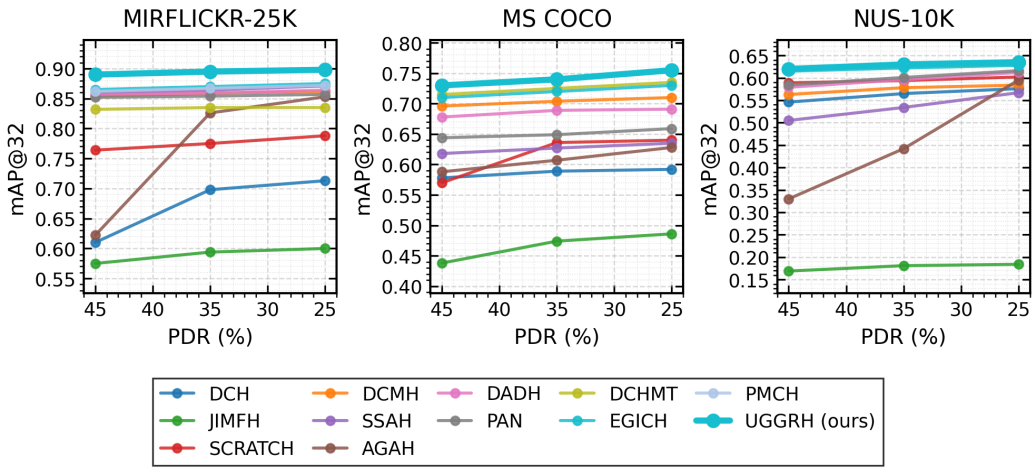


Figure 3: mAP under different incompleteness levels. The horizontal axis is the per-modality missing ratio defined in Table 1; all curves use 32-bit codes.

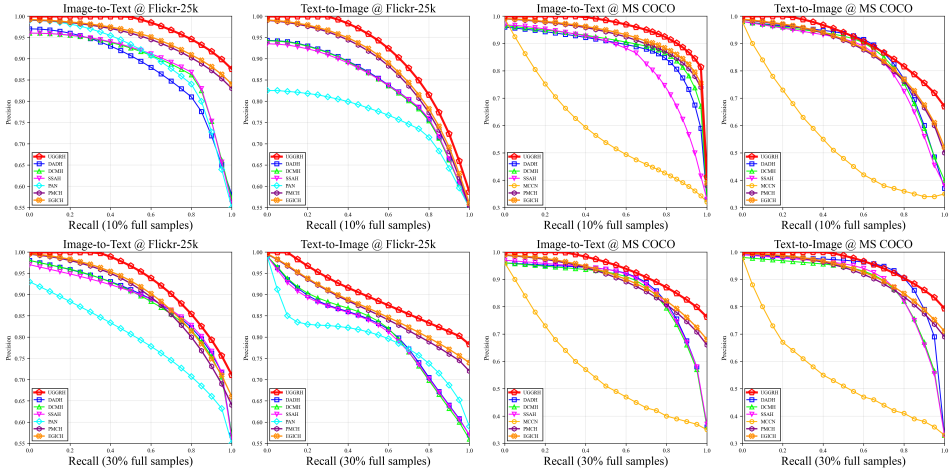


Figure 4: PR curves on MIRFlickr-25K and MS COCO under different incompleteness levels. UGGRH maintains higher precision across recall values, especially under severe incompleteness.

On NUS-WIDE, UGGRH also achieves the highest mean mAP@50 under all three PDR settings, showing that the advantage persists on a noisier multi-label benchmark.

Precision and Recall Analysis. The PR curves in Fig. 4 provide complementary evidence at the ranking level. Across both datasets and retrieval directions, UGGRH maintains higher precision at the same recall, with the separation being most pronounced under the hard configuration with 10% complete samples. The advantage is especially clear on MIRFlickr-25K for T→I retrieval at moderate to high recall and is also observed on MS COCO for both directions, matching the ordering and margins in Table 2. The improvement is not limited to the top of the ranked list, indicating more reliable retrieval under missing modalities beyond the highest ranked items.

4.3 Ablation Study

Variants on Modality Completion Method. Ablations are conducted on MIRFlickr-25K with 50% PDR applied to the training set, while query and retrieval sets remain complete. Four completion variants are compared: **w/o completion**; **+GCAE w/o UGGRH**, following GCIMH [28]; **+MLP w/o UGGRH**, following NCH [31]; and **+PMCH w/o UGGRH**, following PMCH [39].

Table 3: Ablations on MIRFlickr-25K under 50% PDR (mAP@50).

Component	Variant	I → T				T → I			
		16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits
Completion	w/o Completion	0.783	0.802	0.813	0.823	0.718	0.736	0.740	0.751
	+GCAE w/o UGGRH	0.806	0.815	0.827	0.829	0.744	0.759	0.764	0.768
	+MLP w/o UGGRH	0.801	0.817	0.827	0.828	0.744	0.760	0.769	0.772
	+PMCH w/o UGGRH	0.826	0.839	0.845	0.845	0.761	0.773	0.778	0.782
Discriminator	w/o Discriminator	0.880	0.894	0.906	0.912	0.881	0.895	0.907	0.911
HashGAT	w/o HAF	0.884	0.888	0.896	0.896	0.882	0.883	0.896	0.894
	w/o DGC	0.883	0.893	0.894	0.889	0.882	0.893	0.893	0.889
	Standard GAT	0.750	0.717	0.735	0.711	0.744	0.715	0.733	0.710
Generator	OFA + GLIDE	0.879	0.893	0.904	0.908	0.878	0.894	0.903	0.907
Full model	**Ours (UGGRH)**	**0.892**	**0.907**	**0.916**	**0.919**	**0.890**	**0.906**	**0.919**	**0.918**

Table 3 shows that UGGRH outperforms lightweight completion baselines across code lengths. The gains indicate that candidate generation followed by alignment-based filtering provides more useful completed features than direct recovery with GCAE, MLP, or prototype-based alternatives. Replacing BLIP and Stable Diffusion with OFA and GLIDE leads to a small performance drop, showing that the completion module remains effective with alternative pretrained generators while the default BLIP+SD configuration is stronger.

Effectiveness of Image-Text Matching Discriminator and HashGAT. In UGGRH, the image-text matching discriminator provides cross-modal semantic cues by predicting pairwise image-text matching scores. To evaluate its contribution, this discriminator signal is removed and the similarity target is recomputed using only intra-modal similarities, while keeping all other components unchanged.

The discriminator results in Table 3 show that the discriminator signal is important for capturing cross-modal semantic correspondence and constructing a reliable similarity target for HashGAT training. HashGAT is evaluated with four variants: HashGAT w/o HAF (removing hash-aware fusion), HashGAT w/o DGC (disabling dynamic graph construction), Standard GAT (removing both), and Full HashGAT. As shown in Table 3, Full HashGAT achieves the best mAP across code lengths for both I→T and T→I retrieval; ablating either HAF or DGC consistently degrades performance, and Standard GAT performs substantially worse.

4.4 Parameter Sensitivity Analysis

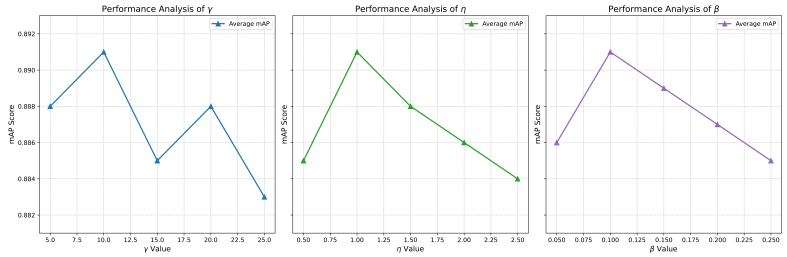


Figure 5: Sensitivity of γ, η, and β on MIRFlickr-25K under the easy setting (16-bit mean mAP@50).

Fig. 5 shows that mean mAP@50 changes by less than 0.01 under moderate sweeps of γ, η, and β; overly large γ slightly degrades performance, while $\gamma = 10$, $\eta = 1.0$, and $\beta = 0.15$ provide a balanced setting. The relatively flat curves indicate that the three loss terms do not require delicate tuning around the selected operating point. This stability is useful for incomplete retrieval settings, where missing-rate changes can otherwise amplify sensitivity to reconstruction or similarity weights. In practice, the reconstruction term mainly preserves modality-specific information, the similarity term controls cross-modal alignment, and the quantization term stabilizes binary-code learning. The mild sensitivity across these weights suggests that UGGRH is not dependent on a narrow hyperparameter configuration.

5 Conclusion

UGGRH addresses target-label-free cross-modal hashing with incomplete training modalities by combining offline completion, frozen dual-encoder filtering, similarity refinement, and HashGAT-based binary-code learning. Since generation is confined to preprocessing, online retrieval remains standard Hamming ranking, and experiments on three benchmarks show consistent gains under different missing rates. These results suggest that generative completion is most useful when paired with alignment filtering and noise-tolerant similarity construction, rather than used as a direct replacement for missing observations. The framework therefore preserves the efficiency of compact hash codes while improving robustness to incomplete training data. One limitation is that completion quality still depends on the pretrained generators and the frozen alignment model under domain shift. Future work will investigate uncertainty-aware filtering and lightweight adaptation while retaining the offline-completion design.

Acknowledgment

This work was supported by the Natural Science Foundation of Hunan Province (2025JJ40057).

References

[1] Bai, C., Zeng, C., Ma, Q., Zhang, J., and Chen, S. Deep adversarial discrete hashing for cross-modal retrieval. In *Proceedings of the 2020 on International Conference on Multimedia Retrieval (ICMR)*, pp. 525–531, 2020.

[2] Chen, J., Pu, R., Peng, D., Song, X., Chen, Y., and Sun, Y. External guidance incomplete cross-modal hashing. *IEEE Transactions on Image Processing*, 35:2596–2608, 2026.

[3] Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., and Zheng, Y. NUS-WIDE: A real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pp. 1–9, 2009.

[4] Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. volume 34, pp. 8780–8794, 2021.

[5] Gao, Z., Wang, J., Yu, G., Yan, Z., Domeniconi, C., and Zhang, J. Long-tail cross modal hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7642–7650, 2023.

[6] Gu, W., Gu, X., Gu, J., Li, B., Xiong, Z., and Wang, W. Adversary guided asymmetric hashing for cross-modal retrieval. In *Proceedings of the 2019 on international conference on multimedia retrieval*, pp. 159–167, 2019.

[7] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. volume 33, pp. 6840–6851, 2020.

[8] Hu, P., Zhu, H., Lin, J., Peng, D., Zhao, Y.-P., and Peng, X. Unsupervised contrastive cross-modal hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3877–3889, 2022.

[9] Huiskes, M. J. and Lew, M. S. The MIR Flickr retrieval evaluation. In *Proceedings of the ACM International Conference on Multimedia Information Retrieval*, pp. 39–43, 2008.

[10] Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

[11] Jiang, K., Wong, W. K., Fang, X., Li, J., Qin, J., and Xie, S. Random online hashing for cross-modal retrieval. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[12] Jiang, Q.-Y. and Li, W.-J. Deep cross-modal hashing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3232–3240, 2017.

[13] Jin, H., Zhang, Y., Shi, L., Zhang, S., Kou, F., Yang, J., Zhu, C., and Luo, J. An end-to-end graph attention network hashing for cross-modal retrieval. *Advances in Neural Information Processing Systems*, 37:2106–2126, 2024.

[14] Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. volume 35, pp. 26565–26577, 2022.

[15] Li, C., Deng, C., Li, N., Liu, W., Gao, X., and Tao, D. Self-supervised adversarial hashing networks for cross-modal retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4242–4251, 2018.

[16] Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. C. H. Align before fuse: Vision and language representation learning with momentum distillation. volume 34, pp. 9694–9705, 2021.

[17] Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

[18] Li, Y., Zhen, L., Sun, Y., Peng, D., Peng, X., and Hu, P. Deep evidential hashing for trustworthy cross-modal retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18566–18574, 2025.

[19] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755, 2014.

[20] Liu, K., Gong, Y., Cao, Y., Ren, Z., Peng, D., and Sun, Y. Dual semantic fusion hashing for multi-label cross-modal retrieval. In *International Joint Conferences on Artificial Intelligence Organization, IJCAI*, pp. 4569–4577, 2024.

[21] Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[22] Luo, H., Zhang, Z., and Luo, Y. Exploiting descriptive completeness prior for cross modal hashing with incomplete labels. volume 37, pp. 50826–50849, 2024.

[23] Luo, H., Zhang, Z., and Nie, L. Contrastive incomplete cross-modal hashing. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):5823–5834, 2024.

[24] Luo, K., Zhang, C., Li, H., Jia, X., and Chen, C. Adaptive marginalized semantic hashing for unpaired cross-modal retrieval. *IEEE Transactions on Multimedia*, 25:9082–9095, 2023.

[25] Peng, S., Yao, T., Li, Y., Wang, G., Wang, L., and Yan, Z. Self-supervised incomplete cross-modal hashing retrieval. *Expert Systems with Applications*, 262:125592, 2025.

[26] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

[27] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695. IEEE, 2022.

[28] Shen, X., Chen, Y., Pan, S., Liu, W., and Zheng, Y. Graph convolutional incomplete multi-modal hashing. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 7029–7037, 2023.

[29] Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, pp. 1–22. OpenReview.net, 2021.

[30] Sun, Y., Dai, J., Ren, Z., Chen, Y., Peng, D., and Hu, P. Dual self-paced cross-modal hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15184–15192, 2024.

[31] Tan, W., Zhu, L., Li, J., Zhang, Z., and Zhang, H. Partial multi-modal hashing via neighbor-aware completion learning. *IEEE Transactions on Multimedia*, 25:8499–8510, 2023.

[32] Tu, J., Liu, X., Lin, Z., Hong, R., and Wang, M. Differentiable cross-modal hashing via multi-modal transformers. In *Proceedings of the 30th ACM International Conference on Multimedia (MM)*, pp. 453–461, 2022.

[33] Wang, D., Wang, Q., He, L., Gao, X., and Tian, Y. Joint and individual matrix factorization hashing for large-scale cross-modal retrieval. *Pattern recognition*, 107:107479, 2020.

[34] Xu, X., Shen, F., Yang, Y., Shen, H. T., and Li, X. Learning discriminative binary codes for large-scale cross-modal retrieval. *IEEE Transactions on Image Processing*, 26(5):2494–2507, 2017.

[35] Zeng, Z., Sun, Y., and Mao, W. Mccn: Multimodal coordinated clustering network for large-scale cross-modal retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 5427–5435, 2021.

[36] Zeng, Z., Wang, S., Xu, N., and Mao, W. Pan: Prototype-based adaptive network for robust cross-modal retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1125–1134, 2021.

[37] Zhang, B. et al. Unsupervised dual deep hashing with semantic-index and content-code for cross-modal retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47 (1):387–399, 2025.

[38] Zhang, S., Wu, Y., Shi, L., Jin, H., Kou, F., Zhang, P., Xu, M., and Lv, P. Adaptive graph attention based discrete hashing for incomplete cross-modal retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28382–28390, 2026. doi: 10.1609/aaai.v40i33.40067.

[39] Zhu, S., Wu, D., Wu, C., Dai, P., and Li, B. Endogenous recovery via within-modality prototypes for incomplete multimodal hashing. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 2521–2529, 2025.

A Appendix

The appendix details the offline modality completion and candidate filtering pipeline (including prompt templates, validity checks, and CLIP-based selection), and summarizes the computational complexity of completion, training, and retrieval.

A.1 Modality Completion and Candidate Filtering

UGGRH performs one-shot offline completion on the training split under the Partial Data Ratio (PDR) protocol in Table 1. Let $\mathcal{I}_{\text{miss-t}}$ and $\mathcal{I}_{\text{miss-i}}$ denote the indices of training instances with missing text and missing images, respectively, as defined in Section 3.2. For each incomplete instance, a small candidate set is generated via stochastic sampling with different random seeds (the seed list is fixed for reproducibility), invalid or unsafe candidates are filtered, a single completion is selected by frozen CLIP dual-encoder alignment [26] to mitigate semantic drift, and the resulting completed CLIP features are cached for subsequent hash learning. Let $E_I(\cdot)$ and $E_T(\cdot)$ denote the frozen CLIP image and text encoders, respectively. Using $\mathrm{norm}(\mathbf{z}) = \frac{\mathbf{z}}{\|\mathbf{z}\|_2}$, the normalized embeddings are defined as $\mathbf{e}_I(\mathbf{x}) = \mathrm{norm}(E_I(\mathbf{x}))$ and $\mathbf{e}_T(\mathbf{x}) = \mathrm{norm}(E_T(\mathbf{x}))$. The cosine alignment score between two embeddings is $\cos(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b}$ when $\mathbf{a}$ and $\mathbf{b}$ are ℓ_2 normalized.

Vision-Driven Caption Generation. For $i \in \mathcal{I}_{\text{miss-t}}$, M_T candidate captions $\mathcal{C}_T^i = \{\mathbf{x}_T^{i,j}\}_{j=1}^{M_T}$ are sampled from BLIP [17] conditioned on the observed image $\mathbf{x}_I^i$ using a fixed seed list $\{s_j\}_{j=1}^{M_T}$.

Text-Guided Image Synthesis. For $i \in \mathcal{I}_{\text{miss-i}}$, M_I candidate images $\mathcal{C}_I^i = \{\mathbf{x}_I^{i,j}\}_{j=1}^{M_I}$ are sampled from Stable Diffusion [27] conditioned on the observed text $\mathbf{x}_T^i$ using a fixed seed list $\{s_j\}_{j=1}^{M_I}$.

Alignment Filtering and Efficiency. Concretely, $\mathcal{C}_T^i$ and $\mathcal{C}_I^i$ are obtained by running the corresponding generator M_T times for caption candidates and M_I times for image candidates with different seeds (either by explicitly passing `seed`=s_j when supported, or by setting the generator RNG state). Lightweight filtering is applied before alignment scoring to remove invalid or unsafe candidates. Formally, indicator functions $\mathbb{I}_T(\cdot)$ and $\mathbb{I}_I(\cdot)$ are defined, and only candidates that pass the checks are kept:

$$\tilde{\mathcal{C}}_T^i = \{\mathbf{x} \in \mathcal{C}_T^i : \mathbb{I}_T(\mathbf{x}) = 1\}, \qquad \tilde{\mathcal{C}}_I^i = \{\mathbf{x} \in \mathcal{C}_I^i : \mathbb{I}_I(\mathbf{x}) = 1\}. \tag{11}$$

For text candidates, $\mathbb{I}_T(\cdot)$ removes captions that fail hygiene checks, including empty strings, overly short captions (e.g., fewer than $L_{\min}$ tokens), excessive repetition (e.g., repeated n-grams beyond a threshold), URLs, and profane content. For image candidates, $\mathbb{I}_I(\cdot)$ removes samples that fail decoding or other validity checks (including NaN failures), are degenerate (e.g., near-constant outputs), or are flagged by model-provided safety filters. If filtering removes all candidates, additional candidates are resampled with new seeds until at least one valid candidate remains (rare in practice). After filtering, candidates are ranked by frozen CLIP alignment with the observed modality (the anchor). For missing text ($i \in \mathcal{I}_{\text{miss-t}}$), each candidate caption is scored by:

$$c_T^{i,j} = \cos\Big(\mathbf{e}_T(\mathbf{x}_T^{i,j}),\, \mathbf{e}_I(\mathbf{x}_I^i)\Big), \tag{12}$$

and select $\hat{\mathbf{x}}_T^i = \mathbf{x}_T^{i,j^*}$ where $j^* = \arg\max_j c_T^{i,j}$ over $\mathbf{x}_T^{i,j} \in \tilde{\mathcal{C}}_T^i$. Similarly, for missing image ($i \in \mathcal{I}_{\text{miss-i}}$):

$$c_I^{i,j} = \cos\Big(\mathbf{e}_I(\mathbf{x}_I^{i,j}),\, \mathbf{e}_T(\mathbf{x}_T^i)\Big), \tag{13}$$

and $\hat{\mathbf{x}}_I^i = \mathbf{x}_I^{i,j^*}$ is selected where $j^* = \arg\max_j c_I^{i,j}$ over $\mathbf{x}_I^{i,j} \in \tilde{\mathcal{C}}_I^i$. The selected completion is used only to obtain its feature embedding (rather than being treated as ground truth):

$$\mathbf{f}_{T,i} = \begin{cases} E_T(\mathbf{x}_T^i), & i \notin \mathcal{I}_{\text{miss-t}}, \\ E_T(\hat{\mathbf{x}}_T^i), & i \in \mathcal{I}_{\text{miss-t}}, \end{cases} \qquad \mathbf{f}_{I,i} = \begin{cases} E_I(\mathbf{x}_I^i), & i \notin \mathcal{I}_{\text{miss-i}}, \\ E_I(\hat{\mathbf{x}}_I^i), & i \in \mathcal{I}_{\text{miss-i}}. \end{cases} \tag{14}$$

Stacking $\{\mathbf{f}_{I,i}\}_{i=1}^N$ and $\{\mathbf{f}_{T,i}\}_{i=1}^N$ yields the completed feature matrices $\mathbf{F}_I, \mathbf{F}_T \in \mathbb{R}^{N \times d}$, which are cached and treated as fixed inputs in subsequent hash learning. This alignment rule can be viewed as selecting the completion that is closest to the observed modality in the shared CLIP space. Specifically, for unit-normalized embeddings $\mathbf{a}$ and $\mathbf{b}$, $\|\mathbf{a} - \mathbf{b}\|_2^2 = 2 - 2\cos(\mathbf{a}, \mathbf{b})$; therefore, maximizing

Algorithm 2 Vision-Driven Caption Generation and Feature Caching (Offline)

Input: $\mathbf{x}_I^i$, seed list $\{s_j\}_{j=1}^{M_T}$; **Output:** $\hat{\mathbf{x}}_T^i$, $\mathbf{f}_{T,i}$
Compute anchor image embedding $\mathbf{e}_I^i = \mathrm{norm}(E_I(\mathbf{x}_I^i))$
Set caption prompt prefix `prefix="a picture of "`
for $j = 1, \ldots, M_T$ **do**
  Sample caption $\mathbf{x}_T^{i,j}$ from BLIP conditioned on $\mathbf{x}_I^i$ with `prefix` using `seed` s_j
  Discard if it fails hygiene checks; otherwise compute $c^{i,j} = \cos\Big(\mathrm{norm}(E_T(\mathbf{x}_T^{i,j})),\ \mathbf{e}_I^i\Big)$
end for
$j^* = \arg\max_j c^{i,j}$; set $\hat{\mathbf{x}}_T^i = \mathbf{x}_T^{i,j^*}$ and cache $\mathbf{f}_{T,i} = E_T(\hat{\mathbf{x}}_T^i)$

Algorithm 3 Text-Guided Image Synthesis and Feature Caching (Offline)

Input: $\mathbf{x}_T^i$, seed list $\{s_j\}_{j=1}^{M_I}$; **Output:** $\hat{\mathbf{x}}_I^i$, $\mathbf{f}_{I,i}$
Set `steps`=30, `cfg_scale`=9.0, `width`=`height`=512, `sampler_name="Euler a"`
Set `suffix="photorealistic, high quality, bright lighting, realistic texture"`
Set `negative_prompt="lowres, blurry, bad anatomy, disfigured, poorly drawn"`
Set `prompt`=$\mathbf{x}_T^i$`+", "+suffix`
Compute anchor text embedding $\mathbf{e}_T^i = \mathrm{norm}(E_T(\mathbf{x}_T^i))$
for $j = 1, \ldots, M_I$ **do**
  Sample image $\mathbf{x}_I^{i,j}$ using `seed` s_j
  Discard if decoding checks fail or it is flagged by model safety filters; otherwise compute $c^{i,j} = \cos\Big(\mathrm{norm}(E_I(\mathbf{x}_I^{i,j})),\ \mathbf{e}_T^i\Big)$
end for
$j^* = \arg\max_j c^{i,j}$; set $\hat{\mathbf{x}}_I^i = \mathbf{x}_I^{i,j^*}$ and cache $\mathbf{f}_{I,i} = E_I(\hat{\mathbf{x}}_I^i)$

cosine similarity is equivalent to minimizing the squared Euclidean distance. This provides a simple and effective criterion for reducing semantic drift among stochastic generations. Algorithms 2 and 3 summarize the offline completion and feature caching procedures for vision-driven captioning and text-guided image synthesis, respectively.

In practice, BLIP captioning and Stable Diffusion sampling can be deployed locally or as inference services; the completion pipeline interacts with generators through a request-and-response interface, where an image payload is submitted and a caption string is returned, and a prompt together with generation parameters is submitted and an image is returned. Each request is executed with timeouts and retries, and per-batch logs and processed IDs are periodically saved for resuming from interruptions. All pretrained models are kept frozen throughout completion, and the seed lists are fixed to ensure reproducibility (e.g., by deterministically generating seeds from the sample ID and candidate index, or by using a fixed global seed table); training uses only the cached CLIP features $\{\mathbf{f}_{I,i}, \mathbf{f}_{T,i}\}$. The default implementation for text-guided image synthesis uses the `Euler a` sampler with 30 steps, CFG scale 9.0, and resolution 512×512, and augments the prompt with a short quality suffix and a negative prompt to avoid low-quality outputs. Vision-driven caption generation uses nucleus sampling with top-$p = 0.9$, maximum length 30, minimum length 10, and an empty prompt prefix (when applicable).

A.2 Complexity Analysis

The time and memory complexity of UGGRH is analyzed by following the main-text modules: Modality Completion via Large-Scale Pretrained Models, Semantics-Enhanced Cross-Modal Similarity Refinement, and Hash-Aware Graph Attention Network (HashGAT), followed by online encoding and Hamming retrieval. Let N be the number of training instances, d the CLIP feature dimension, k the hash length, B the mini-batch size, E the number of epochs, d_h the hidden dimension in HashGAT, H the number of attention heads, K the neighborhood size for dynamic graph construction, and K_s the rowwise sparsification size used in similarity refinement. Pretrained completion models and the discriminator g_θ are used only in preprocessing; training and retrieval operate on cached CLIP features, the fixed target $\mathbf{S}_{\text{cmm}}$, and learned binary codes.

Modality Completion via Large-Scale Pretrained Models (offline). Let N_I and N_T denote the numbers of instances with missing images and missing text, and let M_I and M_T be the numbers of candidates per instance in the two completion directions. The offline completion time is dominated

by candidate generation and CLIP scoring:

$$T_{\text{offline}} = \mathcal{O}(N_I M_I (C_{\text{SD}} + C_{\text{CLIP}}) + N_T M_T (C_{\text{BLIP}} + C_{\text{CLIP}})), \tag{15}$$

where C_{SD} and C_{BLIP} are the per candidate sampling costs for Stable Diffusion and BLIP, and C_{CLIP} is the per candidate cost of CLIP embedding and cosine similarity computation. This cost is incurred only during preprocessing.

Semantics-Enhanced Cross-Modal Similarity Refinement (offline). The refined similarity target $\mathbf{S}_{\text{cmm}}$ is precomputed once via (i) intra-modal similarities $\mathbf{S}_I = \tilde{\mathbf{F}}_I \tilde{\mathbf{F}}_I^\top$ and $\mathbf{S}_T = \tilde{\mathbf{F}}_T \tilde{\mathbf{F}}_T^\top$ followed by affinity scaling, (ii) discriminator affinities $\mathbf{D}$ with entries $\mathbf{D}_{ij} = g_\theta([\tilde{\mathbf{f}}_{I,i}; \tilde{\mathbf{f}}_{T,j}])$, and (iii) rowwise Top-$K_s$ sparsification and residual smoothing. Let C_g denote the per pair forward cost of g_θ; the dominant preprocessing time for building $\mathbf{S}_{\text{cmm}}$ is:

$$T_{\text{sim}} = \mathcal{O}\big(N^2 d + N^2 C_g + N^2 \log K_s + N K_s\big), \tag{16}$$

where $N^2 d$ comes from computing $\mathbf{S}_I$ and $\mathbf{S}_T$, $N^2 C_g$ from evaluating g_θ over all pairs, $N^2 \log K_s$ from rowwise Top-K_s selection (sorting-based upper bound), and $N K_s$ from operating on the sparsified graph and symmetrization. In practice, these terms are computed in blocks, and only the Top-K_s entries per row are kept. The space required for cached CLIP features is:

$$S_{\text{cache}} = \mathcal{O}(Nd), \tag{17}$$

and storing $\mathbf{S}_{\text{cmm}}$ as a sparse matrix costs:

$$S_{\text{target}} = \mathcal{O}(NK_s). \tag{18}$$

Hash-Aware Graph Attention Network (HashGAT) Training. During training, only the fixed target submatrix $\mathbf{S}_{\text{cmm}}[\mathcal{B}, \mathcal{B}]$ is indexed for each mini-batch $\mathcal{B}$; the dominant computation is HashGAT and the losses. For a mini-batch of size B, constructing modality-specific graphs requires pairwise similarities and Top-K selection, costing $\mathcal{O}(B^2 d_h + B^2 \log K)$. Graph attention propagation over approximately BK edges costs $\mathcal{O}(BKHd_h)$, where H is the number of attention heads, and the hash-conditioned edge weights add $\mathcal{O}(BKk)$. Computing the similarity-preserving loss involves forming $\text{sim}(\mathbf{H}_q) = \bar{\mathbf{H}}_q \bar{\mathbf{H}}_q^\top$ for $q \in \{I, T, J\}$, which costs $\mathcal{O}(B^2 k)$, while the encoder and decoder multilayer perceptrons scale linearly in B. Therefore, the dominant time per mini-batch can be summarized as:

$$T_{\text{batch}} = \mathcal{O}\big(B^2 d_h + B^2 \log K + BKHd_h + BKk + B^2 k\big), \tag{19}$$

and the overall training time scales as:

$$T_{\text{train}} = \mathcal{O}\left(E \cdot \frac{N}{B} \cdot \big(B^2 d_h + B^2 \log K + BKHd_h + BKk + B^2 k\big)\right). \tag{20}$$

Online encoding and retrieval. Given Q queries and R database items, online retrieval encodes inputs into k bit codes and computes Hamming distances against the retrieval database. The Hamming distance computation scales as:

$$T_{\text{retrieval}} = \mathcal{O}(QRk). \tag{21}$$

In terms of space, caching CLIP features, storing the sparse similarity target, and storing learned codes require:

$$S_{\text{cache}} = \mathcal{O}(Nd), \qquad S_{\text{target}} = \mathcal{O}(NK_s), \qquad S_{\text{code}} = \mathcal{O}(Nk) \text{ bits}, \tag{22}$$

and within a mini batch sparse neighborhoods and the sparsified similarity target are stored with:

$$S_{\text{graph}} = \mathcal{O}(BK), \qquad S_{\text{batch-target}} = \mathcal{O}(BK_s). \tag{23}$$

Overall, UGGRH pushes the expensive steps into one-shot preprocessing: the dominant superlinear term in N comes from constructing $\mathbf{S}_{\text{cmm}}$ (and, when missing rates are high, from candidate-based completion), while HashGAT training scales linearly in N for fixed B and is driven by dynamic graph construction and message passing within each mini-batch. After preprocessing, online retrieval remains standard hashing in the Hamming space with time linear in database size and code length; choosing moderate K and sparse K_s controls both training cost and the storage footprint of $\mathbf{S}_{\text{cmm}}$ without changing the retrieval complexity.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and introduction state the incomplete cross-modal hashing setting, the three main components of UGGRH, and the empirical scope on MIRFlickr-25K, MS COCO, and NUS-WIDE.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes].

   Justification: The impact statement and appendix discuss limitations related to pretrained generative completion, possible bias or undesirable generations, and the offline preprocessing and similarity-construction cost.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A].

Justification: The paper is empirical and algorithmic, and it does not claim new theoretical results or provide theorems requiring proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes].

Justification: The experiments section reports datasets, PDR settings, metrics, implementation environment, optimizer, training schedule, batch size, hash lengths, and completion details; the appendix further describes completion, filtering, and complexity.

Guidelines:

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

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

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

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

   Answer: [Yes].

   Justification: The datasets used in this work are publicly available benchmark datasets. The source code, training scripts, and implementation details for the proposed method are provided in the supplementary material and will be publicly released upon publication.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: The experimental settings specify the benchmark datasets, incompleteness protocol, retrieval tasks, metrics, hash lengths, optimizer, learning rate, batch size, epochs, hardware, and completion pipeline.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No].

Justification: Following standard protocol in deep cross-modal hashing literature, the main experimental results report standard mAP scores on fixed public dataset splits, consistent with prior work. Due to the high computational cost of offline generative modality completion, multi-run statistical significance tests or error bars are not reported, but the comparison is made fairly under identical experimental conditions.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes].

   Justification: The experiments section reports the GPU, GPU memory, CPU, RAM, software versions, and main training configuration, and the appendix analyzes the computational complexity of completion, training, and retrieval.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes].

   Justification: The work uses standard public image-text retrieval benchmarks and pretrained models, and the impact statement discusses privacy, misuse, bias, and completion-related risks.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes].

    Justification: The impact statement describes potential benefits for multimedia search and organization and discusses possible misuse, privacy-invasive retrieval, bias, and generated-content risks.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A].

    Justification: The paper does not release a new high-risk pretrained model or scraped dataset; it uses existing pretrained generators only for offline feature completion and describes validity and safety filtering.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All three datasets are correctly cited in the references: MIRFlickr-25K, NUS-WIDE, and MS COCO. These are standard public research benchmarks, and their usage adheres to their respective research licenses.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A].

    Justification: The paper does not introduce or release a new dataset, benchmark, pretrained model, or other standalone asset.

    Guidelines:

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

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

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

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A].

    Justification: The paper does not involve human-subject research or participant studies.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A].

    Justification: The core method uses pretrained vision-language encoders, captioning, and diffusion models for completion, but it does not use LLMs as a non-standard core research component.

    Guidelines:

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