PEOCH: Online Cross-Modal Hashing with Semi-Supervised Streaming Data Driving Prototype Evolution

Xiao Kang [1], Xingbo Liu [2 *], Shuo Pan [2], Xuening Zhang [3], Xiushan Nie[2, 4, 5], Yilong Yin [1]

[1]Shandong University, Jinan, China
[2]Shandong Jianzhu University, Jinan, China
[3] Harbin Institute of Technology, Shenzhen, China
[4] Shandong Yunhai Guochuang Cloud Computing Equipment Industry Innovation Co., Ltd, Jinan, China
[5] Institute of Applied AI Engineering and Technology, SDJZU, Jinan, China
{sckx, sclxb}@mail.sdu.edu.cn, panshuo@sdjzu.edu.cn, yukizhang0527@outlook.com, niexsh@hotmail.com, ylyin@sdu.edu.cn

Abstract

The exponential growth of streaming multi-modal data presents critical challenges for cross-modal retrieval: distribution shifts, modality gap, and scarce labels. Semi-supervised online cross-modal hashing has gained increasing interest due to its ability to encode complex streaming data and update hash functions simultaneously. Nevertheless, existing methods can hardly generate high-quality unsupervised hash codes, which fundamentally limits diversity and flexibility during the retrieval process. To this end, we propose a novel method named Prototype Evolution Online Cross-modal Hashing (PEOCH). By driving prototype evolution with semi-supervised streaming data, precise and stable hash codes are generated for both labeled and unlabeled data. Specifically, two prototype updates with stability guarantee are conducted: labeled samples push semantic knowledge into the supervised prototypes, while unlabeled samples perform clustering to generate unsupervised prototypes. Simultaneously, a co-optimization mechanism is designed to ensure the prototypes continuously evolve and preserve the consistency of the entire streaming data. Besides, an elasticity regularizer integrates discriminability and smoothness constraints, improving the reliability of prototypes. Extensive experiments on three benchmark datasets demonstrate that PEOCH outperforms state-of-the-art methods, achieving an average improvement of 6.7% in mAP@all across various retrieval tasks.

Introduction

Given the continuous accumulation and explosive growth of multimedia data, cross-modal hashing has attracted significant research attention across different domains, including computer vision, recommendation systems, retrieval augmented generation for LLMs, and *etc* (Yang et al. 2025; Salemi and Zamani 2024). Existing methods are predominantly designed under a batch-based paradigm *i.e.*, requiring complete data collection prior to training, which cannot adapt to dynamic retrieval in streaming scenarios (Sun et al. 2023; Sun, Peng, and Ren 2024). To address this limitation, online cross-modal hashing has emerged (Ma et al. 2022). It generates hash codes incrementally for continuously evolving data streams while dynamically updating hash functions to capture shifting data distributions (Liang et al. 2024).

Existing online cross-modal hashing methods can be categorized into supervised and unsupervised methods. Supervised methods (Wang, Luo, and Xu 2020; Zhan et al. 2021; Kang et al. 2023b) leverage human-annotated labels to preserve fine-grained semantic similarity. For instance, DOCH (Zhan et al. 2021) alternately refines continuous semantic labels and similarity graphs to tighten the alignment between modalities. DOCMH (Kang et al. 2024b) learns discriminative label embeddings and assigns adaptive bit-wise weights to improve retrieval precision. Unsupervised counterparts (Cui et al. 2024; Kang et al. 2024a) instead mine geometric structure through feature decomposition and reconstruction. OMGH (Liu et al. 2022), for example, employs matrix trifactorization to disentangle modality-specific factors from high-dimensional features, generating hash codes that respect the underlying manifold. Despite recent advancements, existing paradigms unrealistically assume all samples arrive either fully labeled or completely unlabeled. In practice, real-world retrieval datasets contain vast amounts of unlabeled data interspersed with sparse labeled samples (Ren et al. 2022; Yang, Zhao, and Zhao 2025), making accurate and flexible semi-supervised retrieval critically important.

Semi-supervised online cross-modal hashing remains largely unexplored because of three intertwined challenges: distribution drift, modality heterogeneity, and scarce labels. To date, only SSOCH (Kang et al. 2025) attempts a solution. It introduces an online tri-consistent loss that jointly enforces pseudo-label regularization, discriminative embedding, and fine-grained similarity. Nevertheless, SSOCH, as well as the batch-based semi-supervised hashing methods, generate hash codes solely for the labeled samples during the training process, relegating unlabeled data to auxiliary regularizers. In the static batch paradigm, this separation is safe because the retrieval codebase is produced once after training with fixed hash functions. However, in streaming scenarios, the retrieval codebase is continuously accumulating, while hash functions are dynamically updating. Applying varying hash functions

*Corresponding author

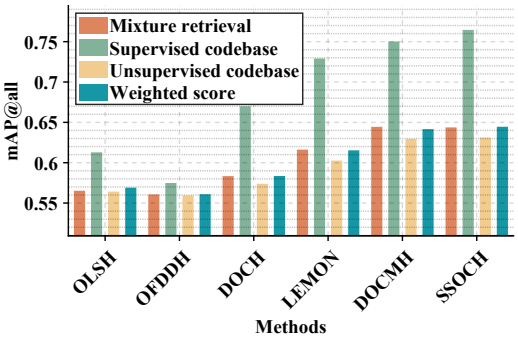


Figure 1: The average performance of two retrieval tasks on different hashing methods, with three codebases and weighted scores on the MIRFLICKR dataset.

struggles to learn high-quality and consistent unlabelled hash codes, leading to unsupervised code incon- consistency problem. Figure 1 illustrates this phenomenon: When evaluated in a retrieval codebase comprising 10% labeled and 90% unlabeled data, existing hashing methods (including SSOCH and supervised methods adapted to semi-supervised settings) exhibit markedly lower accuracy in the unsupervised portion. Moreover, both the weighted score (0.1 × supervised + 0.9 × unsupervised) and mixture retrieval performance are limited by the unsupervised codebase quality. It confirms that the overall effectiveness of semi-supervised retrieval is ultimately dictated by the unsupervised codebase.

To this end, we propose PEOCH, an online cross-modal hashing method in which semi-supervised streaming data drives prototype evolution. It continuously distills class prototypes from the labeled data and refines them with the unlabeled data, alleviating the problem of cumulative unlabeled code inconsistency. Specifically, in the supervised component, we devise a semantics-driven prototype generation module that injects discriminative label information into the prototypes and simultaneously generates the corresponding hash codes. In the unsupervised component, we present a clustering-based prototype embedding module. It employs binary clustering to propagate the learned prototypes into the optimization of unlabeled codes. Through continuous co-optimization, the prototypes are iteratively fine-tuned along with the streaming data, thereby boosting overall retrieval performance. Additionally, we design an elasticity prototype regularization to enhance reliability, constraining evolution from the discriminative initial point while permitting adaptation to streaming data distribution shifts. Rigorous theoretical analysis is provided to guarantee the stability of the streaming prototypes. The main contribution of this study can be concluded as follows:

- We propose a semi-supervised online cross-modal hashing method that generates hash codes for entire instances during the training process. To our knowledge, this may be the first attempt to tackle the unsupervised code inconsistency problem.
- A semi-supervised prototype evolution scheme is designed. It enables knowledge transfer from labeled to unlabeled codes and keeps continuous prototype evolution among streaming data via iterative co-optimization, strengthening the effectiveness of the entire hash codes.
- We introduce an elasticity regularizer that enforces both discriminability and reconstructability on the prototypes, accompanied by theoretical guarantees of stability under the streaming data scenario.
- Extensive analysis and experiments conducted on three widely used datasets verify the superiority of PEOCH.

The Proposed Method

In this section, we firstly introduce notations, then elaborate PEOCH from three aspects, *i.e.*, model formulation, optimization process, and time complexity analysis.

Notations and Problem Statement

Consider a training set $\mathbf{X}$ comprising labeled multi-modality instance pairs. Let $\mathbf{X}_m^{(t)} = \left[\mathbf{X}_{s,m}^{(t)} \in \mathbb{R}^{d_m \times n_{s,t}}, \mathbf{X}_{u,m}^{(t)} \in \mathbb{R}^{d_m \times n_{u,t}}\right]$ be the labeled and unlabeled samples of modality m arriving at chunk t with labels $\mathbf{Y}^{(t)} \in \mathbb{R}^{c \times n_{s,t}}$; cumulative data before t are denoted by $\mathbf{X}_{s,m}^{(t-1)} \in \mathbb{R}^{d_m \times N_{s,t}}$ and $\mathbf{X}_{u,m}^{(t-1)} \in \mathbb{R}^{d_m \times N_{u,t}}$. PEOCH aims to output compact hash codes $\mathbf{B}^{(t)} \in \{-1, +1\}^{r \times n_t}$ via streaming prototype evolution, where $n_t = n_{s,t} + n_{u,t}$.

Model Formulation

As illustrated in Figure 2, PEOCH builds a semi-supervised prototype evolution scheme, where prototypes are learned to guide the generation of hash codes and hashing functions. Besides, an elasticity prototype regularizer is designed for safeguarding the reliability of prototypes.

Semi-supervised Prototype Evolution Scheme Learning class prototypes to guide hash code generation has demonstrated effectiveness in unsupervised online hashing methods (Cao et al. 2025; Yan et al. 2022). In such settings, prototypes are typically derived through matrix factorization or pseudo-label strategies. However, the absence of category priors often results in inferior performance due to two fundamental challenges: determining the optimal number of classes and ensuring prototype reliability. Addressing these limitations typically requires semantic-level supervision. Fortunately, our semi-supervised online scenario provides proportionally labeled samples within each data chunk.

Consequently, we propose a semi-supervised prototype-evolution scheme with two interacting modules: 1) Semantics-driven Prototype Generation extracts semantic knowledge from labeled data as class prototypes, and simultaneously learns supervised hash codes. 2) Clustering-based Prototype Embedding employs the current prototypes to supervise the hash-code learning of unlabeled data and refines the prototypes on the global distribution. These modules are trained within a Co-optimization Training Mechanism, promoting prototypes to evolve with streaming data.

Semantics-driven Prototype Generation: Existing supervised online hashing methods (Chen et al. 2024; Han et al. 2024) map multi-modal features into a shared Hamming

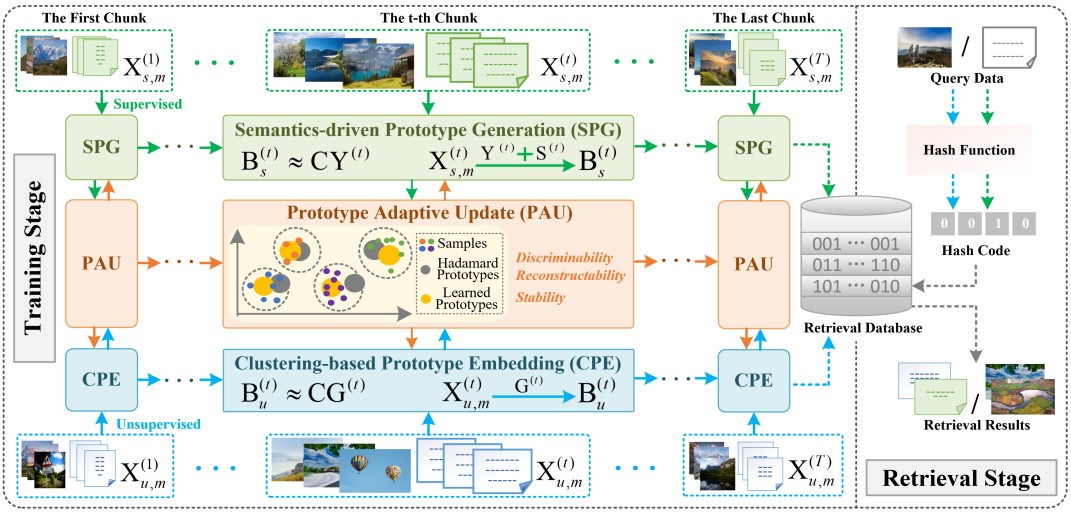


Figure 2: Framework of PEOCH, illustrated with toy data. Semi-supervised streaming chunks are processed through parallel modules: supervised data via *Semantics-driven Prototype Generation (SPG)* and unsupervised data via *Clustering-based Prototype Embedding (CPE)*. Semantic information is propagated across labeled/unlabeled data through *Prototype Adaptive Updating (PAU)*.

space and use label embedding to constrain sample distribution, achieving modality alignment and efficient retrieval. The strategy has proven effective in fully labeled settings. Yet, in the semi-supervised setting, it is hard to preserve the semantic knowledge contained in the labeled data and transfer it to the unlabeled codes, hindering the effectiveness of the unlabeled codes. To address this, we propose Semantics-driven Prototype Generation. It treats the labeled data from a clustering perspective: the label is taken as a cluster indicator, and class prototypes (*i.e.*, cluster centers) are learned by modeling the relationship between features and labels.

Specifically, we construct discriminative labels $\mathbf{Y}^{(t)}$ following (Kang et al. 2025) that offer superior semantic distinctiveness over one-hot labels. We establish a learnable projection $\mathbf{W}_m$ to bridge the relationship between class prototypes $\mathbf{C}$, labels $\mathbf{Y}^{(t)}$ and feature $\mathbf{X}_{s,m}^{(t)}$. Meanwhile, to smooth the solution, prevent over-fitting and improve the stability, we append a ℓ_2-norm regularization on $\mathbf{W}_m$. This process can be formulated as,

$$\min_{\mathbf{C},\mathbf{W}_m} \sum_{m=1}^{M} \left\| \mathbf{W}_m \mathbf{X}_{s,m}^{(t)} - \mathbf{C}\mathbf{Y}^{(t)} \right\|^2 + \gamma \left\| \mathbf{W}_m \right\|^2 . \tag{1}$$

Based on Eq. (6), we embed the optimization of hash codes $\mathbf{B}_s^{(t)}$ in to the training process, which can be rewritten as,

$$\begin{aligned} &\min_{\mathcal{J}} \sum_{m=1}^{M} (\left\| \mathbf{X}_{s,m}^{(t)} - \mathbf{W}_m^T \mathbf{B}_s^{(t)} \right\|^2 + \gamma \left\| \mathbf{W}_m \right\|^2) + \alpha \left\| \mathbf{B}_s^{(t)} - \mathbf{C}\mathbf{Y}^{(t)} \right\|^2 , \\ &\text{s.t. } \mathbf{B}_s^{(t)} \in \{-1, 1\}^{r \times n_{s,t}} , \end{aligned} \tag{2}$$

where $\mathcal{J} = \{\mathbf{B}_u^{(t)}, \mathbf{C}, \mathbf{W}_m\}$. Furthermore, to preserve the pairwise similarity, we employ the inner product to approximate the fine-grained asymmetric similarity matrix $\mathbf{S}_{oc} = 2\mathbf{U}^{(t-1)T}\mathbf{U}^{(t)} - 11^T$, $\mathbf{S}_{cc} = 2\mathbf{U}^{(t)T}\mathbf{U}^{(t)} - 11^T$, where $\mathbf{u}_j^{(t)} = \mathbf{y}_j^{(t)} / \|\mathbf{y}_j^{(t)}\|$. $\mathbf{y}_j^{(t)}$ is the discriminative label of j_{th} instances, $\mathbf{1}$ is an all-one column vector and $\mathbf{E}$ is an all-one matrix. To avoid the challenging binary quadratic problem (Yang 2013), we replace the binary hash codes $\mathbf{B}^{(t)}$ with real-valued representations $\mathbf{V}^{(t)}$ and learn an orthogonal projection $\mathbf{R}$ to map $\mathbf{V}^{(t)}$ into $\mathbf{B}^{(t)}$. Additionally, bit balance and uncorrelation constraints on $\mathbf{V}^{(t)}$ are implemented to promote the accuracy of hash codes. The process can be formulated as,

$$\begin{aligned} &\min_{\mathbf{B}_s^{(t)},\mathbf{R},\mathbf{V}^{(t)}} \left\| \mathbf{B}_s^{(t)} - \mathbf{R}\mathbf{V}^{(t)} \right\|^2 + \beta(\left\| \mathbf{V}^{(t)T}\mathbf{B}_s^{(t)} - r \cdot \mathbf{S}_{cc} \right\|^2 \\ &+ \left\| \mathbf{V}^{(t)T}\mathbf{B}_s^{(t-1)} - r \cdot \mathbf{S}_{oc} \right\|^2), \text{ s.t. } \mathbf{V}^{(t)T}\mathbf{V}^{(t)} = n_{s,t}\mathbf{I}, \\ &\mathbf{V}^{(t)T}\mathbf{1} = \mathbf{0}, \mathbf{R}\mathbf{R}^T = \mathbf{I}, \mathbf{B}_s^{(t)} \in \{-1, +1\}^{r \times n_{s,t}} . \end{aligned} \tag{3}$$

Clustering-based Prototype Embedding: Existing methods for unlabeled hash codes primarily fall into two categories: 1) Matrix decomposition-based methods focus on mining feature-hash code relationships. They typically suffer from distribution shifts in streaming data. 2) Prototype-based methods iteratively update class centers via clustering and generate hash codes accordingly. The absence of category priors leads to unstable prototypes. To this end, we propose a reliable prototype-guided scheme for generating hash codes. It enhances prototype stability by jointly optimizing $\mathbf{C}$ over labeled and unlabeled data, incorporating semantic information distilled from supervised data.

Specifically, we adopts binary clustering to correlate hash codes $\mathbf{B}_u^{(t)}$ with prototypes $\mathbf{C}$. Simultaneously, we integrate matrix decomposition to explore the relationships between $\mathbf{X}_{u,m}^{(t)}$ and $\mathbf{B}_u^{(t)}$. The unified process is formulated as,

$$\begin{aligned} &\min_{\mathcal{G}} \sum_{m=1}^{M} (\left\| \mathbf{X}_{u,m}^{(t)} - \mathbf{W}_m^T \mathbf{B}_u^{(t)} \right\|^2 + \gamma \left\| \mathbf{W}_m \right\|^2) + \alpha \left\| \mathbf{B}_u^{(t)} - \mathbf{C}\mathbf{G}^{(t)} \right\|^2 , \\ &\text{s.t. } \mathbf{B}_u^{(t)} \in \{-1, 1\}^{r \times n_{u,t}}, \mathbf{G}^{(t)} \in \{0, 1\}^{c \times n_{u,t}}, \sum_i \mathbf{g}_{is}^{(t)} = 1, \end{aligned} \tag{4}$$

where $\mathcal{G} = \{\mathbf{B}_u^{(t)}, \mathbf{C}, \mathbf{G}^{(t)}, \mathbf{W}_m\}$, $\mathbf{G}^{(t)}$ is the cluster indicator matrix for all data points.

Co-optimization Training Mechanism: The preceding sections realize prototype adaptation on labeled data and prototype-guided learning of unlabeled hash codes. To ensure prototypes remain valid across the entire dataset, we propose a co-optimization training mechanism. Specifically, we concatenate all hash codes as $\mathbf{B}^{(t)} = [\mathbf{B}_s^{(t)}, \mathbf{B}_u^{(t)}]$, and bridge $\mathbf{B}^{(t)}$ with the class prototypes $\mathbf{C}$ via the class-indicator matrices $\mathbf{Y}^{(t)}$ (labeled) and $\mathbf{G}^{(t)}$ (unlabeled). To let prototypes evolve with the data stream, we jointly optimize accumulated and newly arriving samples, formalized as,

$$\min_{\mathbf{C}} \left\|\mathbf{B}^{(t)} - \mathbf{C}[\mathbf{Y}^{(t)}, \mathbf{G}^{(t)}]\right\|^2 + \left\|\mathbf{B}^{(t-1)} - \mathbf{C}[\mathbf{Y}^{(t-1)}, \mathbf{G}^{(t-1)}]\right\|^2. \quad (5)$$

Elasticity Prototype Regularization The efficacy of our Semi-supervised Prototype Evolution Scheme hinges on prototype reliability, defined by two essential properties: 1) Discriminability: Class prototypes must maintain sufficient inter-class separation. 2) Reconstructability: Prototypes should preserve fine-grained information to enable instance reconstruction via class-indicator matrices. To mitigate the risk of model collapse during adversarial training (Ji et al. 2025), we propose a simple yet effective elasticity constraint. It initializes prototypes from a highly discriminative Hadamard matrix $\mathbf{H}$ (with mutually orthogonal rows/columns), constraining their evolution to start from this optimal point. Meanwhile, it allows prototypes to adapt to streaming data distribution shifts, ensuring feature space reconstruction. Besides, an ℓ_2 regularizer on $\mathbf{C}$ is added to stabilize prototype evolution process. This can be formulated as,

$$\begin{aligned}&\min_{\mathbf{C}} \left\|\mathbf{B}^{(t)} - \mathbf{C}[\mathbf{Y}^{(t)}, \mathbf{G}^{(t)}]\right\|^2 + \left\|\mathbf{B}^{(t-1)} - \mathbf{C}[\mathbf{Y}^{(t-1)}, \mathbf{G}^{(t-1)}]\right\|^2 \\ &+ \gamma(\|\mathbf{H} - \mathbf{C}\|^2 + \|\mathbf{C}\|^2).\end{aligned} \quad (6)$$

Theorem 1 *(Prototype Stability). Assume the feature space satisfies $\|\mathbf{x}_m\| \leq g$. Let the loss $F(\mathbf{C}) \in [0, G]$ be defined as in Eq. (6), and $\mathbf{C}_D$ and $\mathbf{C}_{D'}$ denote the prototypes obtained from datasets D and D' differing in one sample. Then,*

$$|F_D(\mathbf{C}) - F_{D'}(\mathbf{C})| \leq \frac{4g^2 G}{n_t \gamma}, \quad (7)$$

where $\gamma > 0$ is the regularization parameter in Eq. (6). Thus, the prototype update is $\mathcal{O}(1/n_t)$-stable; the influence of any single sample vanishes as the stream grows.

The Overall Hashing Framework Integrating Eq. (2), Eq. (3), Eq. (4) and Eq. (6), the final loss function for hash learning is formulated as,

$$\begin{aligned}&\min_{\mathcal{F}} \eta(\sum_{m=1}^{M} \xi(\left\|\mathbf{B}_u^{(t)} - \mathbf{W}_m \mathbf{X}_{u,m}^{(t)}\right\|^2 + \left\|\mathbf{B}_u^{(t-1)} - \mathbf{W}_m \mathbf{X}_{u,m}^{(t-1)}\right\|^2) \\ &+ (1-\xi)(\left\|\mathbf{B}_s^{(t)} - \mathbf{W}_m \mathbf{X}_{s,m}^{(t)}\right\|^2 + \left\|\mathbf{B}_s^{(t-1)} - \mathbf{W}_m \mathbf{X}_{s,m}^{(t-1)}\right\|^2) \\ &+ \gamma \|\mathbf{W}_m\|^2) + \beta(\left\|\mathbf{V}^{(t)T}\mathbf{B}_s^{(t)} - r\mathbf{S}_{cc}\right\|^2 + \left\|\mathbf{V}^{(t-1)T}\mathbf{B}_s^{(t)} - r\mathbf{S}_{oc}\right\|^2) \\ &+ \alpha(\left\|\mathbf{B}^{(t)} - \mathbf{C}[\mathbf{Y}^{(t)}, \mathbf{G}^{(t)}]\right\|^2 + \left\|\mathbf{B}^{(t-1)} - \mathbf{C}[\mathbf{Y}^{(t-1)}, \mathbf{G}^{(t-1)}]\right\|^2) \\ &+ \left\|\mathbf{B}_s^{(t)} - \mathbf{R}\mathbf{V}^{(t)}\right\|^2 + \gamma(\|\mathbf{H} - \mathbf{C}\|^2 + \|\mathbf{C}\|^2), \\ &\text{s.t. } \mathbf{B}_s^{(t)} \in \{-1, 1\}^{r \times n_{s,t}}, \mathbf{B}_u^{(t)} \in \{-1, 1\}^{r \times n_{u,t}}, \\ &\mathbf{V}^{(t)T}\mathbf{1} = \mathbf{0}, \mathbf{R}\mathbf{R}^T = \mathbf{I}, \mathbf{G}^{(t)} \in \{0, 1\}^{c \times n_{u,t}}, \sum_i \mathbf{g}_{is}^{(t)} = 1,\end{aligned} \quad (8)$$

where $\mathcal{F} = \{\mathbf{B}^{(t)}, \mathbf{C}, \mathbf{G}^{(t)}, \mathbf{R}, \mathbf{V}^{(t)}, \mathbf{W}_m\}$. α, β, ξ, η and γ are hyperparameters.

Optimization Process

Direct optimization of Eq. (8) is challenging because it is non-continuous and non-convex. To overcome this, we introduce an iterative framework with the following steps.

Optimizing $\mathbf{C}$ and $\mathbf{G}^{(t)}$: Removing the terms in Eq. (8) that are not related to $\mathbf{C}$ and $\mathbf{G}^{(t)}$, it leads to the subproblem,

$$\begin{aligned}&\min_{\mathbf{C}, \mathbf{G}^{(t)}} \gamma(\|\mathbf{H} - \mathbf{C}\|^2 + \|\mathbf{C}\|^2) + \alpha(\left\|\mathbf{B}^{(t)} - \mathbf{C}[\mathbf{Y}^{(t)}, \mathbf{G}^{(t)}]\right\|^2 \\ &+ \left\|\mathbf{B}^{(t-1)} - \mathbf{C}[\mathbf{Y}^{(t-1)}, \mathbf{G}^{(t-1)}]\right\|^2), \\ &\text{s.t. } \mathbf{G}^{(t)} \in \{0, 1\}^{c \times n_{u,t}}, \sum_i \mathbf{g}_{is}^{(t)} = 1.\end{aligned} \quad (9)$$

$\mathbf{C}$ step: With the other variables fixed, solving the class prototype $\mathbf{C}$ as a linear least squares problem (Lawson and Hanson 1995). Setting the derivative with respect to $\mathbf{C}$ to $\mathbf{0}$, it can be computed as,

$$\mathbf{C} = (\gamma\mathbf{H} + \alpha\mathbf{F}_1^{(t)})(2\gamma\mathbf{I} + \alpha\mathbf{F}_2^{(t)})^{-1}, \quad (10)$$

where

$$\begin{aligned}\mathbf{F}_1^{(t)} &= \mathbf{F}_1^{(t-1)} + \mathbf{B}^{(t)}[\mathbf{Y}^{(t)}, \mathbf{G}^{(t)}]^T, \\ \mathbf{F}_1^{(t-1)} &= \mathbf{B}^{(t-1)}[\mathbf{Y}^{(t-1)}, \mathbf{G}^{(t-1)}]^T, \\ \mathbf{F}_2^{(t)} &= \mathbf{F}_2^{(t-1)} + [\mathbf{Y}^{(t-1)}, \mathbf{G}^{(t)}][\mathbf{Y}^{(t)}, \mathbf{G}^{(t)}]^T, \\ \mathbf{F}_2^{(t-1)} &= [\mathbf{Y}^{(t-1)}, \mathbf{G}^{(t-1)}][\mathbf{Y}^{(t-1)}, \mathbf{G}^{(t-1)}]^T.\end{aligned} \quad (11)$$

$\mathbf{G}^{(t)}$ step: The indicator matrix $\mathbf{G}^{(t)}$ for clustering can be obtained by calculating the Hamming distance,

$$\mathbf{G}^{(t)}(i, j) = \begin{cases} 1, & j = \arg\min_s \mathbf{D}_H(\mathbf{b}_i, \mathbf{C}(s, :)), \\ 0, & otherwise. \end{cases} \quad (12)$$

Optimizing $\mathbf{V}^{(t)}$: With all variables but the real-value representation $\mathbf{V}^{(t)}$ fixed, Eq. (8) simplifies to,

$$\begin{aligned}&\min_{\mathbf{V}^{(t)}} \left\|\mathbf{B}_s^{(t)} - \mathbf{R}\mathbf{V}^{(t)}\right\|^2 + \beta\left\|\mathbf{V}^{(t)T}\mathbf{B}_s^{(t)} - r \cdot \mathbf{S}_{cc}\right\|^2, \\ &\text{s.t. } \mathbf{V}^{(t)T}\mathbf{V}^{(t)} = n_{s,t}\mathbf{I}, \mathbf{V}^{(t)T}\mathbf{1} = \mathbf{0}.\end{aligned} \quad (13)$$

Under the orthogonal constraint on $\mathbf{V}^{(t)}$, the problem in Eq. (13) is a classical orthogonal Procrustes problem (Xia et al. 2015). After algebraic manipulation, Eq. (13) can be reformulated as,

$$\begin{aligned}&\max_{\mathbf{V}^{(t)}} \operatorname{tr}(\mathbf{O}\mathbf{V}^{(t)T}), \\ &\text{s.t. } \mathbf{V}^{(t)T}\mathbf{V}^{(t)} = n_t\mathbf{I}, \mathbf{V}^{(t)T}\mathbf{1} = \mathbf{0}.\end{aligned} \quad (14)$$

Given $\mathbf{S}_{cc} = 2\mathbf{U}^{(t)T}\mathbf{U}^{(t)} - \mathbf{1}\mathbf{1}^T$, $\mathbf{O}$ can be defined as,

$$\mathbf{O} = \mathbf{R}^T\mathbf{B}_s^{(t)} + 2r\beta\mathbf{B}_s^{(t)}\mathbf{U}^{(t)T}\mathbf{U}^{(t)} - r\beta\mathbf{B}_s^{(t)}\mathbf{1}\mathbf{1}^T. \quad (15)$$

Defining $\mathbf{J} = \mathbf{I} - \frac{1}{n_t}\mathbf{1}\mathbf{1}^T$, Eq. (14) can be solved by conducting the Singular Value Decomposition (SVD) as follows,

$$\mathbf{O}\mathbf{J}\mathbf{O}^T = \begin{bmatrix} \mathbf{Q} & \hat{\mathbf{Q}} \end{bmatrix} \begin{bmatrix} \mathbf{\Omega} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Q} & \hat{\mathbf{Q}} \end{bmatrix}^T, \quad (16)$$

where $\mathbf{\Omega}$ is the diagonal matrix of positive eigenvalues, with $\mathbf{Q}$ and $\hat{\mathbf{Q}}$ denoting the eigenvectors. We execute the Gram-Schmidt process on $\hat{\mathbf{Q}}$ to obtain an orthogonal matrix $\overline{\mathbf{Q}}$.

We further represent $\mathbf{P} = \mathbf{J}\mathbf{O}^T\mathbf{Q}\mathbf{\Omega}^{-1/2}$ and yield a random orthogonal matrix $\overline{\mathbf{P}}$. The closed-form solution of $\mathbf{V}^{(t)}$ can be calculated as,

$$\mathbf{V}^{(t)} = \sqrt{n_t}\begin{bmatrix} \mathbf{Q} & \overline{\mathbf{Q}} \end{bmatrix}\begin{bmatrix} \mathbf{P} & \overline{\mathbf{P}} \end{bmatrix}^T. \tag{17}$$

Optimizing $\mathbf{W}_m$: Optimize the projections $\mathbf{W}_m$, with the other variables fixed. Analogous to the solution of $\mathbf{C}$, the closed-form solution of $\mathbf{W}_m$ becomes,

$$\mathbf{W}_m = (\xi\mathbf{F}_{3,m}^{(t)} + (1-\xi)\mathbf{F}_{5,m}^{(t)})(\xi\mathbf{F}_{4,m}^{(t)} + (1-\xi)\mathbf{F}_{6,m}^{(t)} + \gamma\mathbf{I})^{-1}, \tag{18}$$

where

$$\begin{aligned}
&\mathbf{F}_{3,m}^{(t)} = \mathbf{B}_u^{(t)}\mathbf{X}_{u,m}^{(t)T} + \mathbf{F}_{3,m}^{(t-1)}, \mathbf{F}_{3,m}^{(t-1)} = \mathbf{B}_u^{(t-1)}\mathbf{X}_{u,m}^{(t-1)},\\
&\mathbf{F}_{4,m}^{(t)} = \mathbf{X}_{u,m}^{(t)}\mathbf{X}_{u,m}^{(t)T} + \mathbf{F}_{4,m}^{(t-1)}, \mathbf{F}_{4,m}^{(t-1)} = \mathbf{X}_{u,m}^{(t-1)}\mathbf{X}_{u,m}^{(t-1)T},\\
&\mathbf{F}_{5,m}^{(t)} = \mathbf{B}_s^{(t)}\mathbf{X}_{s,m}^{(t)T} + \mathbf{F}_{5,m}^{(t-1)}, \mathbf{F}_{5,m}^{(t-1)} = \mathbf{B}_s^{(t-1)}\mathbf{X}_{s,m}^{(t-1)},\\
&\mathbf{F}_{6,m}^{(t)} = \mathbf{X}_{s,m}^{(t)}\mathbf{X}_{s,m}^{(t)T} + \mathbf{F}_{6,m}^{(t-1)}, \mathbf{F}_{6,m}^{(t-1)} = \mathbf{X}_{s,m}^{(t-1)}\mathbf{X}_{s,m}^{(t-1)T}.
\end{aligned} \tag{19}$$

Optimizing $\mathbf{B}_s^{(t)}$: Optimize the hash codes $\mathbf{B}_s^{(t)}$, holding the other variables fixed. This subproblem becomes,

$$\begin{aligned}
&\min_{\mathbf{B}_s^{(t)}} \left\|\mathbf{B}_s^{(t)} - \mathbf{R}\mathbf{V}^{(t)}\right\|^2 + \alpha\left\|\mathbf{B}_s^{(t)} - \mathbf{C}\mathbf{Y}^{(t)}\right\|^2\\
&+ \beta(\left\|\mathbf{V}^{(t)T}\mathbf{B}_s^{(t)} - r\mathbf{S}_{cc}\right\|^2 + \left\|\mathbf{V}^{(t-1)T}\mathbf{B}_s^{(t)} - r\mathbf{S}_{oc}\right\|^2)\\
&+ \eta\sum_{m=1}^{M}(1-\xi)\left\|\mathbf{B}_s^{(t)} - \mathbf{W}_m\mathbf{X}_{s,m}^{(t)}\right\|^2, \text{s.t. } \mathbf{B}_s^{(t)} \in \{-1,1\}^{r\times n_{s,t}}.
\end{aligned} \tag{20}$$

Considering $\left\|\mathbf{B}_s^{(t)}\right\|^2 = n_{s,t}\times r$, after several algebraic transformations, Eq. (20) can be rewritten as follows,

$$\begin{aligned}
&\max_{\mathbf{B}_s^{(t)}} \operatorname{tr}(\mathbf{B}_s^{(t)T}(\mathbf{R}\mathbf{V}^{(t)} + \alpha\mathbf{C}\mathbf{Y}^{(t)} + \beta r(\mathbf{V}^{(t)}\mathbf{S}_{cc} + \mathbf{V}^{(t-1)}\mathbf{S}_{oc}))\\
&+ \eta\sum_{m=1}^{M}(1-\xi)\mathbf{W}_m\mathbf{X}_{s,m}^{(t)}, \quad \text{s.t. } \mathbf{B}^{(t)} \in \{-1,+1\}^{n_{s,t}\times r}.
\end{aligned} \tag{21}$$

Given the definition of $\mathbf{S}_{cc}$ and $\mathbf{S}_{oc}$, the solution of $\mathbf{B}_s^{(t)}$ can be derived as,

$$\begin{aligned}
\mathbf{B}_s^{(t)} = \operatorname{sgn}(&\mathbf{R}\mathbf{V}^{(t)} + \alpha\mathbf{C}\mathbf{Y}^{(t)} + 2\beta r\mathbf{F}_5^{(t)}\mathbf{U}^{(t)}\\
&- \beta r\mathbf{F}_6^{(t)}\mathbf{1}^T + \eta\sum_{m=1}^{M}(1-\xi)\mathbf{W}_m\mathbf{X}_{s,m}^{(t)}),
\end{aligned} \tag{22}$$

where

$$\begin{aligned}
&\mathbf{F}_7^{(t)} = \mathbf{F}_7^{(t-1)} + \mathbf{V}^{(t)}\mathbf{U}^{(t)T},\ \mathbf{F}_7^{(t-1)} = \mathbf{V}^{(t-1)}\mathbf{U}^{(t-1)T},\\
&\mathbf{F}_8^{(t)} = \mathbf{F}_8^{(t-1)} + \mathbf{V}^{(t)}\mathbf{1},\ \mathbf{F}_8^{(t-1)} = \mathbf{V}^{(t-1)}\mathbf{1}.
\end{aligned} \tag{23}$$

Optimizing $\mathbf{B}_u^{(t)}$: Learn the hash codes $\mathbf{B}_u^{(t)}$, holding the other variables fixed. The optimization is rewritten as,

$$\begin{aligned}
&\min_{\mathbf{B}_u^{(t)}} \eta\sum_{m=1}^{M}\xi\left\|\mathbf{X}_{u,m}^{(t)} - \mathbf{W}_m\mathbf{B}_u^{(t)}\right\|^2 + \alpha\left\|\mathbf{B}_u^{(t)} - \mathbf{C}\mathbf{G}^{(t)}\right\|^2,\\
&\text{s.t. } \mathbf{B}_u^{(t)} \in \{-1,1\}^{r\times n_{u,t}}.
\end{aligned} \tag{24}$$

Analogous to the solution of $\mathbf{B}_s^{(t)}$, the solution of $\mathbf{B}_u^{(t)}$ can be derived as,

$$\mathbf{B}_u^{(t)} = \operatorname{sgn}(\eta\sum_{m=1}^{M}\xi\mathbf{W}_m^{(t)T}\mathbf{X}_{u,m}^{(t)} + \alpha\mathbf{C}\mathbf{G}^{(t)}). \tag{25}$$

Optimizing $\mathbf{R}$: Learn the orthogonal matrix $\mathbf{R}$ with other variables unchanged. The optimization problem in Eq. (8) can be rewritten as,

$$\max_{\mathbf{R}} \operatorname{tr}(\mathbf{V}^{(t)}\mathbf{B}^{(t)T}\mathbf{R}), \quad \text{s.t.} \quad \mathbf{R}\mathbf{R}^T = \mathbf{I}. \tag{26}$$

We compute the Singular Value Decomposition (SVD) to optimize $\mathbf{R}$, *i.e.*, $\mathbf{V}^{(t)}\mathbf{B}^{(t)T} = \mathbf{K}\sum\mathbf{O}^T$, where $\mathbf{K}$ is a $C\times C$ orthogonal matrix, $\sum$ is a $C\times r$ matrix and $\mathbf{O}$ is a $r\times r$ orthogonal matrix. The closed-form solution for $\mathbf{R}$ is

$$\mathbf{R} = \mathbf{O}\hat{\mathbf{K}}^T, \tag{27}$$

where $\hat{\mathbf{K}}$ contains first r columns of $\mathbf{K}$.

Time Complexity Analysis

The time complexities for optimizing C-Step, V-Step, and R-Step are $O(n_{u,t}rc + n_{u,t}c^2 + n_{s,t}rc)$, $O(n_{s,t}d_m r + n_{s,t}r^2)$, and $O(n_{s,t}rc + n_{s,t}r^2)$ respectively. The time complexities $O(n_{s,t}d_m r + n_{s,t}rc)$ and $O(n_{u,t}d_m r + n_{u,t}rc)$ correspond to solving the hash codes $\mathbf{B}_s^{(t)}, \mathbf{B}_u^{(t)}$. Besides, learning the projection $\mathbf{W}_m$ incurs time complexities of $O(n_{u,t}d_m^2 + n_{s,t}d_m^2 + n_{u,t}d_m r)$. As $n_{u,m}$ is usually much larger than r, and c, the overall training time complexity for training PEOCH simplifies to $T\cdot O(n_{u,t}d_m^2 + n_{u,s}d_m^2 + n_{u,t}d_m r)$, where T is the number of iterations. This complexity is linear to $n_{u,t}$, making PEOCH scalable for large-scale streaming datasets. large-scale streaming datasets.

Experiment

To validate the effectiveness, we evaluate PEOCH on three benchmark cross-modal retrieval datasets: IAPR TC-12 (Escalante et al. 2010), NUSWIDE (Chua et al. 2009), and MIRFLICKR (Huiskes and Lew 2008). This section details experimental settings, performance analysis, and ablation studies.

Experimental Settings

The proposed PEOCH is evaluated against 13 state-of-the-art online cross-modal hashing methods, including 1) Unsupervised method: OCMH (Xie, Shen, and Zhu 2016), OCMFH (Wang et al. 2020), OMGH (Liu et al. 2022), SPOCH (Kang et al. 2023a), DPOCH (Kang et al. 2024a), OHMFH (Kang et al. 2024c); 2) Supervised method: OLSH (Yao et al. 2019), LEMON (Wang, Luo, and Xu 2020), OFDDH (Liu, Wang, and Cheung 2022), DOCH (Zhan et al. 2021), ROHLSE (Li et al. 2023), DOCMH (Kang et al. 2024b); 3) Semi-Supervised method: SSOCH(Kang et al. 2025). To simulate semi-supervised settings, 10% instances per chunk are sampled for supervised baselines, and their hash codes of unlabeled data are obtained using the hashing function. All results report five-run averages.

In the implementation, we empirically set $\eta = 10^3$ and $\gamma = 10$. Moreover, we set $\alpha = 10^{-4}$, $\beta = 10^5$, $\xi = 10^{-4}$ through cross-validation and grid search. The number of iterations T is set to 10. The length of the Hadamard label c is set as 32 for the MIRFLICKR and NUSWIDE datasets, and 256 for the IAPR TC-12 dataset. All experiments are performed on a computer with an Intel(R) Core(TM) i9-10900K CPU@ 3.70GHz 64GB RAM.

Method	Publish	IAPR TC-12				NUSWIDE				MIRFLICKR			
		8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits
OCMH	AAAI/2016	0.3039	0.3025	0.3012	0.3019	0.3419	0.3416	0.3398	0.3397	0.5553	0.5565	0.5551	0.5557
OCMFH	SIGIR/2020	0.3007	0.3006	0.2989	0.2983	0.3676	0.3677	0.3741	0.3769	0.5641	0.5618	0.5605	0.5590
OMGH	TMM/2022	0.3047	0.3046	0.3046	0.3044	0.3406	0.3403	0.3406	0.3408	0.5562	0.5559	0.5556	0.5560
SPOCH	JCRD/2023	0.3383	0.3484	0.3634	0.3817	0.3496	0.3417	0.3419	0.3887	0.5657	0.5661	0.5781	0.5848
DPOCH	TOMM/2024	0.3026	0.3099	0.3100	0.3104	0.3402	0.3400	0.3405	0.3409	0.5611	0.5616	0.5630	0.5624
OHMFH	ICME/2024	0.3288	0.3393	0.3375	0.3291	0.3571	0.3577	0.3690	0.3608	0.5706	0.5691	0.5685	0.5678
OLSH	PR/2019	0.3203	0.3304	0.3369	0.3412	0.3523	0.3597	0.3595	0.3532	0.5830	0.5830	0.5698	0.5704
LEMON	MM/2020	0.3610	0.3711	0.3834	0.3939	0.5308	0.5372	0.5447	0.5496	0.6091	0.6144	0.6178	0.6213
OFDDH	TNNLS/2022	0.3057	0.3063	0.3071	0.3075	0.3925	0.4005	0.4134	0.4158	0.5602	0.5608	0.5609	0.5619
DOCH	PR/2022	0.3301	0.3293	0.3356	0.3402	0.5253	0.5538	0.5648	0.5703	0.5808	0.5860	0.5935	0.5968
ROHLSE	PR/2023	0.3143	0.3142	0.3167	0.3169	0.3421	0.3418	0.3434	0.3434	0.5616	0.5610	0.5613	0.5618
DOCMH	PR/2024	0.3817	0.3965	0.4096	0.4166	**0.5429**	0.5435	0.5490	0.5526	0.6361	0.6433	0.6451	0.6497
SSOCH	AAAI/2025	0.3884	0.3985	0.4064	0.4112	0.5414	**0.5451**	0.5403	0.5511	**0.6366**	**0.6380**	**0.6469**	0.6494
Ours	-	**0.3949**	**0.3999**	**0.4101**	**0.4185**	0.5232	0.5432	**0.5521**	**0.5575**	0.6277	0.6325	0.6446	**0.6495**
OCMH	AAAI/2016	0.3010	0.3031	0.3029	0.3007	0.3432	0.3438	0.3402	0.3399	0.5523	0.5552	0.5561	0.5555
OCMFH	SIGIR/2020	0.3377	0.3369	0.3331	0.3303	0.3932	0.4036	0.4168	0.4223	0.5604	0.5598	0.5594	0.5590
OMGH	TMM/2022	0.3047	0.3046	0.3046	0.3048	0.3410	0.3407	0.3406	0.3406	0.5565	0.5559	0.5564	0.5562
SPOCH	JCRD/2023	0.3468	0.3539	0.3781	0.4017	0.3525	0.3430	0.3457	0.3981	0.5666	0.5673	0.5776	0.5905
DPOCH	TOMM/2024	0.3032	0.3106	0.3124	0.3142	0.3405	0.3403	0.3410	0.3416	0.5602	0.5616	0.5625	0.5627
OHMFH	ICME/2024	0.3460	0.3557	0.3541	0.3481	0.3539	0.3559	0.3765	0.3727	0.5842	0.5846	0.5845	0.5764
OLSH	PR/2019	0.3090	0.3159	0.3232	0.3297	0.3502	0.3526	0.3511	0.3445	0.5675	0.5619	0.5607	0.5619
LEMON	MM/2020	0.3542	0.3632	0.3737	0.3832	0.5330	0.5411	0.5489	0.5539	0.6039	0.6081	0.6107	0.6134
OFDDH	TNNLS/2022	0.3043	0.3058	0.3058	0.3069	0.3657	0.3717	0.3882	0.3981	0.5597	0.5598	0.5605	0.5614
DOCH	PR/2022	0.3244	0.3267	0.3331	0.3365	0.5100	0.5386	0.5469	0.5488	0.5758	0.5806	0.5863	0.5876
ROHLSE	PR/2023	0.3101	0.3116	0.3142	0.3150	0.3424	0.3417	0.3436	0.3438	0.5607	0.5605	0.5610	0.5614
DOCMH	PR/2024	0.3822	0.3919	0.4013	0.4086	0.5369	0.5452	0.5535	0.5580	0.6319	0.6389	0.6426	0.6456
SSOCH	AAAI/2025	0.3899	0.3958	0.4062	0.4141	0.5276	0.5467	0.5593	0.5619	0.6331	0.6367	0.6406	0.6434
Ours	-	**0.4287**	**0.4505**	**0.4735**	**0.4913**	**0.6266**	**0.6523**	**0.6653**	**0.6722**	**0.6635**	**0.6720**	**0.6800**	**0.6878**

Table 1: Mixture codebase retrieval performance about mAP@all score on three benchmark datasets. The top panel is the performance for the Image2Text task, while the bottom panel is for the Text2Image task. The best mAP@all values of each case are shown in boldface.

We evaluate the proposed approach on two standard cross-modal retrieval tasks: 1) Image2Text: Retrieving texts using image queries; 2) Text2Image: Retrieving images using text queries. Performance is assessed on three retrieval codebases: unsupervised, supervised, and mixture. We adopt standard metrics: mean Average Precision (mAP@all), mAP@100, precision@100, and training time is reported to evaluate computational efficiency.

Performance Analyses

Accuracy: Table 1 presents the mAP@all scores for PEOCH and baselines across three benchmark datasets under the mixture codebase retrieval. PEOCH demonstrates significant improvements, achieving an average mAP@all gain of 6.7% over all tasks and bit lengths compared to the baselines. Particularly, in Text2Image tasks, it attains remarkable performance gains. A possible reason is the higher semantic density and structural clarity of the text modality, which enables more effective distillation of discriminative prototypes. For Image2Text tasks, PEOCH delivers competitive performance in most scenarios, although a slight decrease is observed under short-bit conditions. This minor decline can potentially be attributed to insufficient prototype discriminability in low-dimensional spaces on small-scale datasets.

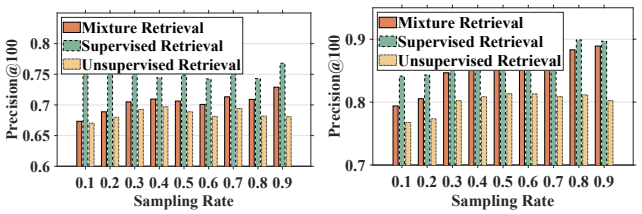
To further evaluate the performance of PEOCH, we conducted experiments with progressively increasing sampling rates. As depicted in Figure 3, the Precision@100 trajectories across three representative retrieval codebases on the MIRFLICKR dataset reveal a consistent stabilization pattern: all systems exhibit negligible performance fluctuations once the sampling rate attains 0.3, with further rate increases yielding only marginal improvements. This convergence phenomenon strongly indicates that a limited supervised subset suffices to effectively harness the potential of unlabeled data streams. These findings validate that sparse supervision can anchor model stability while maintaining retrieval efficacy.

Parameter Sensitivity: We conduct experiments to analyze the parameter sensitivity, including the parameters α and β. Figure 4 depicts the mAP@100 scores of the proposed PEOCH when these two parameters range from 10^{-7} to 10^{7}. The proposed PEOCH exhibits satisfactory stability and sensitivity as parameters change.

Training Time: Figure 5 indicates that PEOCH attains comparable or superior time costs relative to state-of-the-art baselines across varying data chunks. This demonstrates a key strength of our framework: it not only elevates retrieval accuracy but also maintains high computational efficiency, ensuring its practical utility for streaming data retrieval.

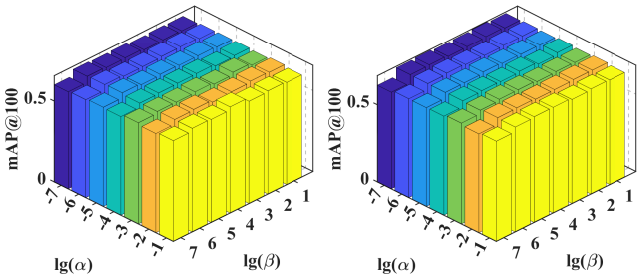
Method	IAPR TC-12				NUSWIDE				MIRFLICKR			
	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits
PEOCH-h	0.3907	0.3959	0.4097	0.4176	0.5244	0.5333	0.5489	0.5530	0.6297	0.6315	0.6320	0.6382
PEOCH-c	0.3550	0.3600	0.3721	0.3810	0.5033	0.5193	0.5349	0.5458	0.5957	0.5975	0.6119	0.6168
PEOCH-s	0.3113	0.3279	0.3329	0.3489	0.3438	0.3446	0.3484	0.3521	0.5563	0.5639	0.5679	0.5704
PEOCH	0.3949	0.3999	0.4101	0.4185	0.5232	0.5432	0.5521	0.5575	0.6277	0.6325	0.6446	0.6495
PEOCH-h	0.4151	0.4489	0.4618	0.4801	0.6112	0.6353	0.6563	0.6653	0.6523	0.6606	0.6709	0.6875
PEOCH-c	0.3810	0.3979	0.4199	0.4398	0.5950	0.6301	0.6498	0.6635	0.6178	0.6236	0.6399	0.6496
PEOCH-s	0.3137	0.3284	0.3372	0.3584	0.3462	0.3481	0.3549	0.3611	0.5561	0.5642	0.5658	0.5708
PEOCH	0.4287	0.4505	0.4735	0.4913	0.6266	0.6523	0.6653	0.6722	0.6635	0.6720	0.6800	0.6878

Table 2: Ablation study in terms of mAP@all score on three benchmark datasets. The top panel is the performance for the Image2Text task, while the bottom panel is for the Text2Image task.



(a) Image2Text (b) Text2Image

Figure 3: The Precision@100 score with three retrieval codebases on the MIRFLICKR dataset.



(a) Image2Text (b) Text2Image

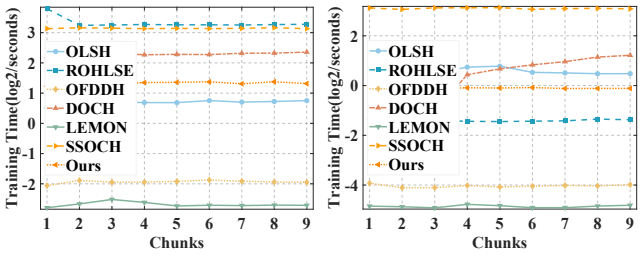
Figure 4: The parameter sensitivity with mAP@100 score about α and β on the MIRFLICKR dataset.

Ablation Study

To further validate the contribution of each key component, we design three variants. The comparative performance is summarized in Table 2.

PEOCH-h: To evaluate the impact of the Hadamard-based initialization, we create an ablated variant, PEOCH-h, by removing the corresponding regularization term in Eq. (8). The observed performance degradation in PEOCH-h confirms that the Hadamard constraint is instrumental in guiding effective prototype generation and updates.

PEOCH-c: To assess the effectiveness of the class prototype, we construct PEOCH-c, which discards the class prototype, thereby optimizing the unsupervised hash codes solely based on feature decomposition and reconstruction. Compared to PEOCH, this variant exhibits a substantial performance drop,



(a) IAPRTC-12 (b) NUSWIDE

Figure 5: The training time results vary with chunks based on IAPRTC-12 and NUSWIDE datasets.

verifying the critical role of the class prototype in unsupervised hash code learning.

PEOCH-s: The variant PEOCH-s removes the asymmetric similarity preservation, demonstrating a substantial performance deterioration. This indicates that enhancing the reliability and accuracy of hash codes corresponding to supervised samples through similarity preservation is crucial for ensuring the overall performance of the retrieval database.

Conclusion

This paper presented a novel semi-supervised online cross-modal hashing framework designed for dynamic retrieval scenarios, named PEOCH. Different from existing methods that suffer from unsupervised code inconsistency, PEOCH attempts to simultaneously generate hash codes for all instances (labeled/unlabeled) during the training process. Specifically, we propose a semi-supervised prototype evolution scheme, where class prototypes distilled from labeled data propagate semantic information to guide unlabeled hash code generation. Meanwhile, an iterative co-optimization is designed. It enables prototypes to evolve across data chunks, thus enhancing overall code reliability. Furthermore, an elasticity regularizer integrating discriminability and smoothness constraints provides provable stability under streaming scenarios. Theoretical analyses and extensive experiments demonstrate the superiority of PEOCH. Future work will extend the prototype evolution scheme to class-incremental scenarios for enhancing generalization capability.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (U23A20389, 62206160, 62176141, 62176139), Young Talent of Lifting Engineering for Science and Technology in Shandong (SDAST2024QTA020), Major Basic Research Project of Shandong Provincial Natural Science Foundation (ZR2024ZD03), Natural Science Foundation of Shandong Province (ZR2022QF082), Taishan Scholar Project of Shandong Province (tsqn202103088), Shandong Provincial Natural Science Foundation for Distinguished Young Scholars (ZR2021JQ26), China Postdoctoral Science Foundation (2025M771512), the Science and Technology Innovation Program for Distinguished Young Scholars of Shandong Province Higher Education Institutions (2024KJH084), and special funds for distinguished professors of Shandong Jianzhu University.

References

Cao, Y.; Chen, X.; Liu, Z.; Jia, W.; Meng, F.; and Gui, J. 2025. Deep Graph Online Hashing for Multi-Label Image Retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 1953–1961.

Chen, Y.; Fang, Y.; Zhang, Y.; Ma, C.; Hong, Y.; and King, I. 2024. Towards effective top-n hamming search via bipartite graph contrastive hashing. *IEEE Transactions on Knowledge and Data Engineering*.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. 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*, 48. ACM.

Cui, H.; Zhao, L.; Li, F.; Zhu, L.; Han, X.; and Li, J. 2024. Effective comparative prototype hashing for unsupervised domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 8329–8337.

Escalante, H. J.; Hernández, C. A.; González, J. A.; López-López, A.; Montes-y-Gómez, M.; Morales, E. F.; Sucar, L. E.; Pineda, L. V.; and Grubinger, M. 2010. The segmented and annotated IAPR TC-12 benchmark. *Comput. Vis. Image Underst.*, 114(4): 419–428.

Han, K.; Liu, Y.; Wei, R.; Zhou, K.; Xu, J.; and Long, K. 2024. Supervised hierarchical online hashing for cross-modal retrieval. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(4): 1–23.

Huiskes, M. J.; and Lew, M. S. 2008. The MIR flickr retrieval evaluation. In *Proceedings of the 1st ACM SIGMM International Conference on Multimedia Information Retrieval, 2008*, 39–43.

Ji, J.; Wang, K.; Qiu, T. A.; Chen, B.; Zhou, J.; Li, C.; Lou, H.; Dai, J.; Liu, Y.; and Yang, Y. 2025. Language Models Resist Alignment: Evidence From Data Compression. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 23411–23432. Vienna, Austria: Association for Computational Linguistics.

Kang, X.; Liu, X.; Lu, P.; Zhao, Z.; Nie, X.; Wang, S.; and Yin, Y. 2023a. Online Cross-Modal Hashing with Double Structure Preserving. *Journal of Computer Research and Development*, 1–13.

Kang, X.; Liu, X.; Xue, W.; Nie, X.; and Yin, Y. 2024a. Online Cross-modal Hashing With Dynamic Prototype. *ACM Trans. Multimedia Comput. Commun. Appl.*, 20(8).

Kang, X.; Liu, X.; Xue, W.; Zhang, X.; Nie, X.; and Yin, Y. 2024b. Discrete online cross-modal hashing with consistency preservation. *Pattern Recognition*, 110688.

Kang, X.; Liu, X.; Zhang, X.; Nie, X.; and Yin, Y. 2023b. Online Discriminative Cross-modal Hashing. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1.

Kang, X.; Liu, X.; Zhang, X.; Xue, W.; Nie, X.; Wang, S.; and Yin, Y. 2024c. Unsupervised Online Cross-modal Hashing With Multiple Association Exploitation. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. Niagara Falls, ON, Canada: IEEE. ISBN 979-8-3503-9015-5.

Kang, X.; Liu, X.; Zhang, X.; Xue, W.; Nie, X.; and Yin, Y. 2025. Semi-Supervised Online Cross-Modal Hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 17770–17778.

Lawson, C. L.; and Hanson, R. J. 1995. *Solving least squares problems*, volume 15 of *Classics in applied mathematics*. Siam.

Li, L.; Shu, Z.; Yu, Z.; and Wu, X.-J. 2023. Robust online hashing with label semantic enhancement for cross-modal retrieval. *Pattern Recognition*, 145: 109972.

Liang, X.; Yang, E.; Yang, Y.; and Deng, C. 2024. Multi-Relational Deep Hashing for Cross-Modal Search. *IEEE Transactions on Image Processing*.

Liu, X.; Wang, X.; and Cheung, Y.-M. 2022. FDDH: Fast Discriminative Discrete Hashing for Large-Scale Cross-Modal Retrieval. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11): 6306–6320.

Liu, X.; Yi, J.; Cheung, Y.-m.; Xu, X.; and Cui, Z. 2022. OMGH: Online Manifold-Guided Hashing for Flexible Cross-modal Retrieval. *IEEE Transactions on Multimedia*, 1–1.

Ma, Z.; Ju, W.; Luo, X.; Chen, C.; Hua, X.; and Lu, G. 2022. Improved Deep Unsupervised Hashing via Prototypical Learning. In Magalhães, J.; Bimbo, A. D.; Satoh, S.; Sebe, N.; Alameda-Pineda, X.; Jin, Q.; Oria, V.; and Toni, L., eds., *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, 659–667. ACM.

Ren, X.; Zheng, X.; Zhou, H.; Liu, W.; and Dong, X. 2022. Contrastive hashing with vision transformer for image retrieval. *Int. J. Intell. Syst.*, 37(12): 12192–12211.

Salemi, A.; and Zamani, H. 2024. Evaluating Retrieval Quality in Retrieval-Augmented Generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, 2395–2400. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704314.

Sun, Y.; Peng, D.; and Ren, Z. 2024. Discrete aggregation hashing for image set classification. *Expert Systems with Applications*, 237: 121615.

Sun, Y.; Wang, X.; Peng, D.; Ren, Z.; and Shen, X. 2023. Hierarchical Hashing Learning for Image Set Classification. *IEEE Transactions on Image Processing*, 32: 1732–1744.

Wang, D.; Wang, Q.; An, Y.; Gao, X.; and Tian, Y. 2020. Online Collective Matrix Factorization Hashing for Large-Scale Cross-Media Retrieval. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 1409–1418. ACM.

Wang, Y.; Luo, X.; and Xu, X. 2020. Label Embedding Online Hashing for Cross-Modal Retrieval. In *The 28th ACM International Conference on Multimedia, 2020*, 871–879. ACM.

Xia, Y.; He, K.; Kohli, P.; and Sun, J. 2015. Sparse projections for high-dimensional binary codes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3332–3339.

Xie, L.; Shen, J.; and Zhu, L. 2016. Online Cross-Modal Hashing for Web Image Retrieval. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence,*, 294–300. AAAI Press.

Yan, K.; Zhang, C.; Hou, J.; Wang, P.; Bouraoui, Z.; Jameel, S.; and Schockaert, S. 2022. Inferring Prototypes for Multi-Label Few-Shot Image Classification with Word Vector Guided Attention. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 2991–2999. AAAI Press.

Yang, L.; Zhao, Z.; and Zhao, H. 2025. Unimatch v2: Pushing the limit of semi-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yang, R. 2013. New results on some quadratic programming problems. *Dissertations & Theses - Gradworks*.

Yang, S.; Chen, Y.; Tian, Z.; Wang, C.; Li, J.; Yu, B.; and Jia, J. 2025. VisionZip: Longer is Better but Not Necessary in Vision Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19792–19802.

Yao, T.; Wang, G.; Yan, L.; Kong, X.; Su, Q.; Zhang, C.; and Tian, Q. 2019. Online latent semantic hashing for cross-media retrieval. *Pattern Recognit.*, 89: 1–11.

Zhan, Y. W.; Wang, Y.; Sun, Y.; Wu, X. M.; Luo, X.; and Xu, X. S. 2021. Discrete Online Cross-Modal Hashing. *Pattern Recognition*.