Video Hashing via a Mamba-Transformer Network for Retrieval

Likai Yang[1,2,3], Nianqiao Li[1,2,3], Xiaoping Liang[1,2,3], Lv Chen[1,2,3], and Zhenjun Tang[1,2,3](✉)

[1] Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin 541004, China
tangzj230@163.com

[2] Guangxi Key Lab of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China

[3] University Engineering Research Center of Educational Intelligent Technology, Guangxi Normal University, Guilin 541004, China

Abstract. Self-Supervised Video Hashing (SSVH) has been widely applied to efficient video retrieval. Existing methods mostly use Siamese-like pairwise training to model inter-sample similarities, yet lack direct optimization for high-quality video content representation, limiting practical performance. To address this issue, we propose a novel SSVH method based on the Mamba-Transformer network (hereafter VHMT), which employs a teacher-student architecture. The teacher model extracts robust low-frequency semantic features using the Discrete Wavelet Transform (DWT) and captures temporal dependencies through a Vision Transformer (ViT)-based Temporal Attention Module (TAM). Concurrently, the introduced Temporal Reconstruction Enhancement (TRE) block further improves fine-grained reconstruction, thereby generating high-quality temporal modeling signals. The student model employs the Mamba architecture to model temporal relationships and achieves knowledge transfer and efficient single-path inference by approximating the teacher model's feature reconstruction and attention outputs. Furthermore, we introduce a joint loss function to enhance the student model's approximation of the teacher model's temporal modeling capability while optimizing the clustering structure of the generated hash codes. Experimental results demonstrate the VHMT outperforms several state-of-the-art methods in mean Average Precision.

Keywords: Video hashing · Video retrieval · Vision Transformer (ViT) · Mamba

1 Introduction

With the rise of video-sharing platforms, efficient video retrieval has become crucial. Video hashing addresses this by compressing high-dimensional data into

L. Yang and N. Li—Contributed equally to this work.

J. Lokoč et al. (Eds.): MMM 2026, LNCS 16413, pp. 104–118, 2026.
https://doi.org/10.1007/978-981-95-6957-1_8

compact binary codes, enabling fast similarity search with low storage cost [1–3]. Recent deep learning advances have boosted its performance, among which Self-Supervised Video Hashing (SSVH) stands out by generating pseudo-labels via pre-training, achieving high accuracy without costly manual annotations [4]. This makes SSVH more practical than supervised methods in real-world scenarios.

Most existing SSVH methods still rely on a pairwise training paradigm similar to Siamese networks [5], focusing primarily on modeling inter-sample similarity, but they lack direct optimization for high-quality video content representation, thereby limiting model performance in real-world tasks [6–8]. Moreover, these methods exhibit insufficient capability in temporal modeling and feature reconstruction [9–11]. To address these issues, we propose a novel SSVH method based on the Mamba-Transformer network (hereafter VHMT), with the following main contributions:

(1) A novel teacher-student network is proposed, in which the teacher model extracts robust low-frequency semantic features via Discrete Wavelet Transform (DWT) and captures temporal dependencies through a Temporal Attention Module (TAM) derived from the Vision Transformer (ViT). The student model employs a Mamba to model temporal relationships and achieves knowledge transfer and efficient single-path inference by approximating the teacher model's feature reconstruction and attention outputs.
(2) The proposed TAM employs a Multi-Head Attention (MHA) mechanism to model temporal dependencies across video segments, and introduces a novel Temporal Reconstruction Enhancement (TRE) module to improve the fine-grained reconstruction of temporal features.
(3) A novel loss function is proposed to jointly optimize temporal reconstruction, hash contrastive learning, and local structure preservation, thereby enhancing the student model's approximation of the teacher model's temporal modeling capabilities while simultaneously optimizing the clustering structure of the hash codes.

Experiments on three public datasets [12–14] show that the VHMT outperforms several state-of-the-art (SOTA) methods in video hashing retrieval under the mean Average Precision (mAP) metric. The rest of the paper is organized as follows: Sect. 2 reviews related work; Sect. 3 details the VHMT method; Sect. 4 reports results; and Sect. 5 concludes the paper.

2 Related Work

Recent advances in deep learning have propelled the application of SSVH in large-scale retrieval. Early methods, such as SSTH [15] and JTAE [16], primarily adopted Recurrent Neural Networks (RNNs) to extract temporal features from videos. Subsequently, several LSTM-based encoder-decoder framework methods were proposed [4], attempting to reconstruct original frame features from relaxed hash codes. Although the memory capacity of these models hinders modeling high-dimensional temporal information, these methods have contributed foundational training paradigms for early SSVH development.

Later, the introduction of Transformers has driven SSVH towards more sophisticated temporal modeling. BTH [9] was the first Transformer-based SSVH method, proposing a bidirectional encoder structure to fully exploit long-range bidirectional correlations among frames. DKPH [17] introduced a Gaussian-adaptive similarity graph into the Transformer model, decoupling temporal reconstruction from semantic retrieval, offering a new perspective for SSVH advancement. Following this, ConMH [6] designed a Siamese network architecture under a Masked Autoencoder (MAE)-inspired training paradigm, significantly improving retrieval accuracy through masked contrastive learning. Benefiting from this progress, subsequent Transformer-based methods built on Siamese networks have emerged, including TSVH [10], SPVH [11], and AutoSSVH [8].

Meanwhile, several methods based on the MLP-Mixer [18] have been developed, such as MCMSH [19] and EUVH [20], which leverage the simplicity of MLPs to build more efficient retrieval models. Additionally, MAGRH [21], based on Graph Neural Networks [22], offers a novel direction in relational reasoning. Recently, S5VH [7] introduced an encoder-decoder structure based on Mamba, which is both effective and efficient in capturing temporal relationships, providing new potential for SSVH.

Most existing SSVH methods still rely on a Siamese-like pairwise training paradigm that emphasizes inter-sample similarity, but they lack direct optimization for high-quality video content representation, thereby limiting their effectiveness in real-world tasks. Furthermore, these methods exhibit limited capability in temporal modeling and feature reconstruction. This study is dedicated to addressing these issues.

3 Proposed VHMT Method

3.1 Problem Definition and Overview

Consider an unlabeled video dataset containing N videos, represented as $V = \{V_i\}_{i=1}^{N}$, where each $V_i \in \mathbb{R}^{T \times D}$ denotes the frame-level features of the i-th video. These features are extracted using a pretrained CNN backbone (e.g., VGG or ResNet [23,24]), with T and D indicating the number of frames and the feature dimension, respectively. The proposed VHMT network, depicted in Fig. 1, takes V_i as input and outputs a compact binary hash code $b_i \in \{-1, +1\}^k$, ensuring that the Hamming distance between any two codes corresponds to the semantic similarity between the respective videos.

3.2 Video Hash Network

The VHMT adopts a teacher-student dual-path architecture [17] to jointly optimize temporal feature extraction and hash code learning. The teacher branch M^{T} employs a TAM to capture discriminative temporal patterns from full-frame

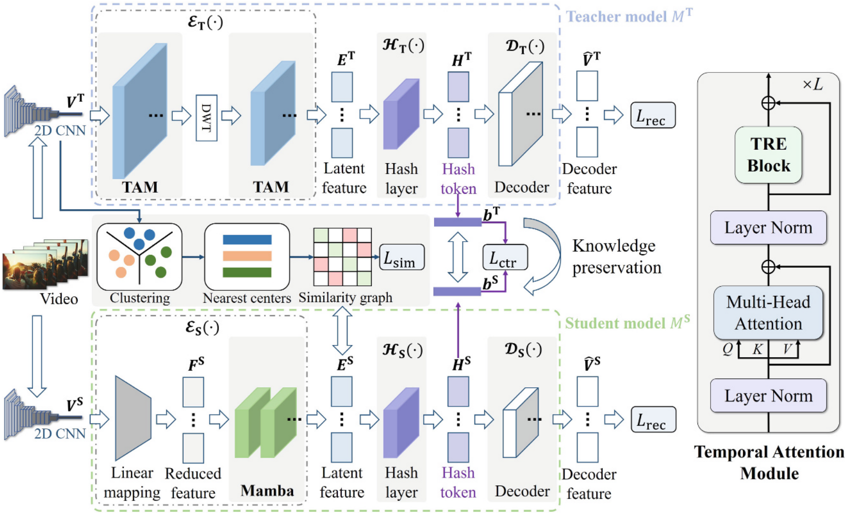


Fig. 1. Overview of the VHMT network.

features. The student branch M^{S} uses a lightweight Mamba structure [25] for efficient encoding. Between the teacher and student models, a parameter interaction mechanism is introduced to address the insufficient knowledge transfer commonly observed in conventional Siamese networks.

Preliminaries: State-Space Models and Mamba. State-space models (SSMs) process sequential data $\boldsymbol{x}(t) \in \mathbb{R}^d$ by maintaining a hidden state $\boldsymbol{h}(t) \in \mathbb{R}^u$, governed by

$$\dot{\boldsymbol{h}}(t) = \boldsymbol{A}\boldsymbol{h}(t) + \boldsymbol{B}\boldsymbol{x}(t), \quad \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{h}(t), \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{u\times u}$, $\boldsymbol{B} \in \mathbb{R}^{u\times 1}$, and $\boldsymbol{C} \in \mathbb{R}^{1\times u}$ are system matrices. Discretizing the continuous model using zero-order hold with step size Δ yields

$$\bar{\boldsymbol{A}} = e^{\Delta \boldsymbol{A}}, \quad \bar{\boldsymbol{B}} = \boldsymbol{A}^{-1}(e^{\Delta \boldsymbol{A}} - \boldsymbol{I})\boldsymbol{B}, \tag{2}$$

$$\boldsymbol{h}_t = \bar{\boldsymbol{A}}\boldsymbol{h}_{t-1} + \bar{\boldsymbol{B}}\boldsymbol{x}_t, \quad \boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{h}_t, \tag{3}$$

where $\boldsymbol{x}_t$, $\boldsymbol{h}_t$, and $\boldsymbol{y}_t$ are the input, hidden state, and output at discrete time step t. Mamba [25] enhances SSMs by making Δ, $\boldsymbol{B}$, and $\boldsymbol{C}$ input-adaptive, enabling selective feature propagation for better sequence modeling.

Preliminaries: 1D Discrete Wavelet Transform. The 1D DWT decomposes an input sequence $\boldsymbol{F}$ into low-frequency ($\boldsymbol{F}_{\mathrm{low}}$) and high-frequency ($\boldsymbol{F}_{\mathrm{high}}$) components, with Inverse Wavelet Transform (IWT) enabling full reconstruction:

$$[\boldsymbol{F}_{\mathrm{low}}, \boldsymbol{F}_{\mathrm{high}}] = \mathrm{DWT}(\boldsymbol{F}), \quad \boldsymbol{F} = \mathrm{IWT}(\boldsymbol{F}_{\mathrm{low}}, \boldsymbol{F}_{\mathrm{high}}), \tag{4}$$

where $\boldsymbol{F}_{\text{low}}$ captures stable global structural information of the video features, while $\boldsymbol{F}_{\text{high}}$ contains local details (e.g., textures, edges) [26]. In the teacher model, only $\boldsymbol{F}_{\text{low}}$ is used to promote stable temporal features.

Teacher Model. The teacher encoder $\mathcal{E}_{\text{T}}(\cdot)$ consists of alternating TAM and DWT layers:

$$\mathcal{E}_{\text{T}}(\boldsymbol{F}) = \text{TAM}^{\times E_n} \circ \text{DWT} \circ \text{TAM}^{\times E_{n-1}} \circ \cdots \circ \text{DWT} \circ \text{TAM}^{\times E_1}(\boldsymbol{F}), \tag{5}$$

where $\text{TAM}^{\times E_i}$ denotes a sequence of E_i TAM blocks, and $E_{\text{T}} = \{E_1, \ldots, E_n\}$ defines the per-layer depth. The decoder $\mathcal{D}_{\text{T}}(\cdot)$ reconstructs the input using a stack of TAM blocks:

$$\mathcal{D}_{\text{T}}(\boldsymbol{H}) = \text{TAM}^{\times D_{\text{T}}}(\boldsymbol{H}), \tag{6}$$

where D_{T} represents the total number of TAM blocks.

Student Model. The student model adopts a simplified architecture. Its encoder $\mathcal{E}_{\text{S}}(\cdot)$ and decoder $\mathcal{D}_{\text{S}}(\cdot)$ each consist of a single Mamba block stack:

$$\mathcal{E}_{\text{S}}(\boldsymbol{F}) = \text{Mamba}^{\times E_{\text{S}}}(\boldsymbol{F}), \tag{7}$$

$$\mathcal{D}_{\text{S}}(\boldsymbol{H}) = \text{Mamba}^{\times D_{\text{S}}}(\boldsymbol{H}), \tag{8}$$

where E_{S} and D_{S} denote the number of Mamba blocks in the encoder and decoder, respectively.

Hash Layer. To generate hash codes, the latent features $\boldsymbol{E}_i \in \mathbb{R}^{T\times d}$ are mapped to real-valued vectors $\tilde{\boldsymbol{H}}_i \in \mathbb{R}^{T\times k}$ via a fully connected layer FC$(\cdot)$, Layer Normalization LN$(\cdot)$, and the tanh$(\cdot)$ function:

$$\tilde{\boldsymbol{H}}_i = \tanh(\text{LN}(\text{FC}(\boldsymbol{E}_i))) \in (-1, 1), \quad \boldsymbol{H}_i = \text{sign}(\tilde{\boldsymbol{H}}_i). \tag{9}$$

The final binary code $\boldsymbol{b}_i \in \{-1, 1\}^{1\times k}$ is obtained by average pooling over the temporal dimension T and binarization using sign$(\cdot)$:

$$\tilde{\boldsymbol{b}}_i = \frac{1}{T}\sum_{j=1}^{T} \boldsymbol{H}_{i,j}, \quad \boldsymbol{b}_i = \text{sign}(\tilde{\boldsymbol{b}}_i). \tag{10}$$

3.3 Temporal Attention Module

The teacher model uses the TAM module to integrate low-frequency features extracted via DWT, generating temporal modeling signals:

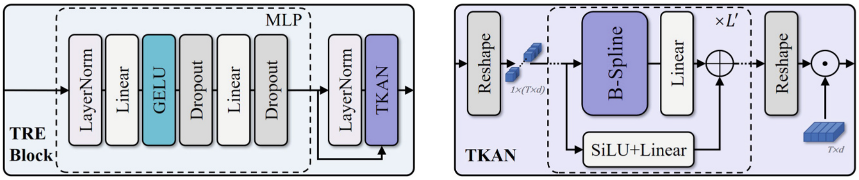
$$y = \text{MHA}(\text{LN}(x)) + \text{LN}(x), \quad z = \text{TRE}(\text{LN}(y)) + \text{LN}(y), \tag{11}$$

where x and z are the input and output features, respectively, MHA$(\cdot)$ is Multi-Head Attention from ViT, TRE$(\cdot)$ is the Temporal Reconstruction Enhancement block, and LN$(\cdot)$ is layer normalization.

Multi-head Attention. Following standard ViT blocks [27], we project input matrix $\boldsymbol{F}_i$ into query $\boldsymbol{Q}_i$, key $\boldsymbol{K}_i$, and value $\boldsymbol{V}_i$ matrices using learnable parameters $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$, and $\boldsymbol{W}^V$. The scaled dot-product attention is defined as

$$\boldsymbol{F}_i^{\text{update}} = \text{softmax}\left(\frac{\boldsymbol{F}_i\boldsymbol{W}^Q(\boldsymbol{F}_i\boldsymbol{W}^K)^\top}{\sqrt{d^K}}\right)\boldsymbol{F}_i\boldsymbol{W}^V, \tag{12}$$

where d^K is a scaling factor. After E_T MHA layers, inputs are mapped to d-dimensional latent features $\boldsymbol{H}_i \in \mathbb{R}^{T\times d}$, encoding both frame-specific content and temporal dependencies.



(a) TRE Block (b) TKAN

Fig. 2. Overview of the TRE block.

Temporal Reconstruction Enhancement Block. As shown in Fig. 2, the TRE block integrates a Temporal KAN (TKAN) [28] to overcome the fixed activation limitation in MLPs [19,20]. For input x_in, the transformation is

$$x_\text{MLP} = \text{MLP}(x_\text{in}) = \text{Drop}(\text{Linear}(\text{Drop}(\text{GELU}(\text{Linear}(\text{LN}(x_\text{in})))))), \tag{13}$$

$$x'_\text{MLP} = \text{Reshape}(\text{LN}(x_\text{MLP})), \tag{14}$$

$$x_\text{TKAN} = \text{Reshape}(\text{TKAN}(x'_\text{MLP})), \tag{15}$$

$$x_\text{out} = x_\text{TKAN} * x_\text{MLP}, \tag{16}$$

where x_MLP is the MLP$(\cdot)$ output, LN$(\cdot)$, GELU$(\cdot)$, Drop$(\cdot)$ denote Layer Normalization, GELU activation, and Dropout, Reshape$(\cdot)$ aligns dimensions for fusion, and $*$ is the Hadamard product. Let $\widetilde{x} = x'_\text{MLP}$, TKAN is defined as TKAN$(\widetilde{x}) = (\varPhi_{L'} \circ \cdots \circ \varPhi_1)\widetilde{x}$, with each $\varPhi_i$ being a KANLinear layer:

$$\varPhi(\widetilde{x}) = \omega \cdot \frac{\widetilde{x}}{1+e^{-\widetilde{x}}} + \omega' \cdot \sum_{p=1}^{N} c_p F_{p,q}(\widetilde{x}), \tag{17}$$

where ω, ω' are trainable weights, c are spline coefficients, and $F_{p,q}(\widetilde{x})$ are degree-q B-spline bases. The TRE block enables adaptive fusion for enhanced reconstruction.

3.4 Loss Function

To jointly optimize temporal reconstruction, hash contrastive learning, and local structure preservation, we propose a multi-component loss:

$$\mathcal{L}_{\text{VHMT}} = \alpha\mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{ctr}} + \gamma\mathcal{L}_{\text{sim}}, \tag{18}$$

where α, β, γ represent adjustable hyperparameters for task balancing.

Temporal Reconstruction. Inspired by [7,8], we perform temporal reconstruction via hash code decoding to maximize semantic capacity. For video sequences $\boldsymbol{V}_i \in \mathbb{R}^{T\times D}$, the reconstruction loss measures feature accuracy:

$$\mathcal{L}_{\text{rec}} = \frac{1}{NTD}\sum_{i=1}^{N}\sum_{j=1}^{T}\|\boldsymbol{V}_{i,j} - \hat{\boldsymbol{V}}_{i,j}\|_2^2, \tag{19}$$

where $\boldsymbol{V}_{i,j}$ is the j-th frame feature of video i, $\hat{\boldsymbol{V}}_{i,j}$ is its reconstruction, N is the batch size, T is the number of frames, and D is the feature dimension.

Hash Contrastive. To improve consistency between network views and mitigate sampling bias in teacher-student frameworks, we adopt a debiased InfoNCE loss [29]. For each video, correlated views $\boldsymbol{b}_i^{\text{S}}$ (student) and $\boldsymbol{b}_i^{\text{T}}$ (teacher) are treated as positives. The loss dynamically reweights negative samples:

$$\mathcal{L}_{\text{ctr}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\left(\mathcal{S}(\boldsymbol{b}_i^{\text{S}}, \boldsymbol{b}_i^{\text{T}})/\tau\right)}{\exp\left(\mathcal{S}(\boldsymbol{b}_i^{\text{S}}, \boldsymbol{b}_i^{\text{T}})/\tau\right) + (2N-2)\bar{f}_i}, \tag{20}$$

where $\bar{f}_i$ is a dynamic baseline:

$$\bar{f}_i = \max\left(\frac{\theta - \rho\exp\left(\mathcal{S}(\boldsymbol{b}_i^{\text{S}}, \boldsymbol{b}_i^{\text{T}})/\tau\right)}{1-\rho}, \exp(-1/\tau)\right), \tag{21}$$

and θ estimates the average negative similarity:

$$\theta = \frac{1}{2N-2}\sum_{\substack{w=1\\ w\neq i}}^{N}\left[\exp\left(\mathcal{S}(\boldsymbol{b}_i^{\text{S}}, \boldsymbol{b}_w^{\text{S}})/\tau\right) + \exp\left(\mathcal{S}(\boldsymbol{b}_i^{\text{S}}, \boldsymbol{b}_w^{\text{T}})/\tau\right)\right], \tag{22}$$

where N is the batch size, $\mathcal{S}(\cdot,\cdot)$ is cosine similarity, $\tau > 0$ controls temperature, $\rho \in [0, 1]$ estimates intra-video similarity, and θ modulates negative contributions. This formulation enhances hash code discriminability by reducing label noise in contrastive learning.

Local Structure Preservation. To preserve semantic similarity and discrimination, we construct a similarity matrix $\boldsymbol{S} \in \{-1, 0, 1\}^{N \times N}$ using K-means and PCA hybrid clustering [9]. Video-level representations $\bar{\boldsymbol{v}}_i$ are obtained by aggregating frame-level features V^{T}. A sparse affinity matrix $\boldsymbol{P} \in \mathbb{R}^{N \times M}$ is then computed based on proximity to cluster centroids $\{\boldsymbol{c}_{i,1}, \ldots, \boldsymbol{c}_{i,M}\}$:

$$\boldsymbol{P}_{i,m} = \begin{cases} \frac{\exp(-\|\bar{\boldsymbol{v}}_i - \boldsymbol{c}_{i,m}\|^2/\sigma)}{\sum_{l=1}^{M} \exp(-\|\bar{\boldsymbol{v}}_i - \boldsymbol{c}_{i,l}\|^2/\sigma)}, & \text{if } \boldsymbol{c}_{i,m} \text{ is the nearest centroid,} \\ 0, & \text{otherwise,} \end{cases} \tag{23}$$

where $\sigma > 0$ controls the kernel bandwidth. An affinity matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is then computed as

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{P}^{\top}, \tag{24}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{P}^{\top}\mathbf{1})$ is a diagonal matrix of cluster weights. $\boldsymbol{A}$ is thresholded at three levels $M_1 < M_2 < M_3$ to obtain binary matrices $\boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)}, \boldsymbol{B}^{(3)} \in \{1, -1\}^{N \times N}$. The refined similarity matrix $\boldsymbol{S}$ is defined as

$$\boldsymbol{S}_{i,m} = \begin{cases} 1, & \text{if } \boldsymbol{B}^{(1)}_{i,m} = 1, \\ -1, & \text{if } \boldsymbol{B}^{(2)}_{i,m} = -1 \text{ and } \boldsymbol{B}^{(3)}_{i,m} = 1, \\ 0, & \text{otherwise,} \end{cases} \tag{25}$$

encoding positive, negative, and neutral pairs. The local structure loss is

$$\mathcal{L}_{\mathrm{sim}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{N} \left(\frac{1}{d} e_i^{\top} e_m - \boldsymbol{S}_{i,m} \right)^2, \tag{26}$$

where d is the latent feature E^{S} dimension, e_i, e_m are pooled latent vectors, and the loss preserves neighborhood structure while reducing compression distortion.

4 Experiments

4.1 Datasets

The VHMT is evaluated on three standard large-scale video datasets: FCVID [12], ActivityNet [13], and YFCC [14], with their statistics detailed in Table 1.

Table 1. Datasets used in the experiments.

Dataset	Videos	Train	Test	Classes
FCVID [12]	91,223	45,585	45,600	239
ActivityNet [13]	20,000	9,722	4,758	203
YFCC [14]	800,000	409,788	101,256	80

4.2 Evaluation Metrics

Following [7,8], we use mAP@k as the main metric:

$$\text{mAP@}k = \frac{1}{N}\sum_{q=1}^{N} \text{AP@}k(q), \tag{27}$$

where N is the number of queries and AP@$k(q)$ is

$$\text{AP@}k(q) = \frac{1}{g_q}\sum_{r=1}^{k} P(r|q)\cdot\delta_r(q), \quad g_q = \min(k, R_q), \tag{28}$$

with R_q the total relevant items for query q, $P(r|q)$ the precision at rank r, and $\delta_r(q) = 1$ if the r-th result is relevant, else 0. For overall evaluation, we compute GmAP as the root-aggregate score over key k values:

$$\text{GmAP} = \sqrt{\sum_{k\in\mathcal{K}}(\text{mAP@}k)^2}, \quad \mathcal{K} = \{5, 20, 40, 60, 80, 100\}. \tag{29}$$

4.3 Implementation Details

Preprocessing. We adopt the standard preprocessing pipeline used in SSVH [6–11]. For FCVID and YFCC, 25 frames per video are sampled, and 4,096-D VGG-16 [30] features are used; for ActivityNet, 30 frames and 2,048-D ResNet50 [24] features are extracted. All backbones are pre-trained on ImageNet [31]. Features for FCVID and YFCC are sourced from [4], and those for ActivityNet from [19].

Model Settings. Following [7], the student encoder of VHMT has a depth of 6 and a hidden dimension of 256, while the decoder has a depth of 1 and a hidden dimension of 192. The preset state dimension u in Mamba is set to 16 [25]. TKAN employs 2 KANLinear layers (L'), 5 spline coefficients (c), and a recursion degree of $q = 3$ [28]. The Haar wavelet is used as the DWT basis.

Loss Settings. The temperature parameter τ in the contrastive loss is set to 0.5, and the class prior ρ is set to 0.1 [29]. The number of clusters M is fixed at 2000, with $M_1 = 3$, $M_2 = 4$, and $M_3 = 5$ [19]. Loss weights are set to $\alpha = 1$, $\beta = 1$, and $\gamma = 0.1$.

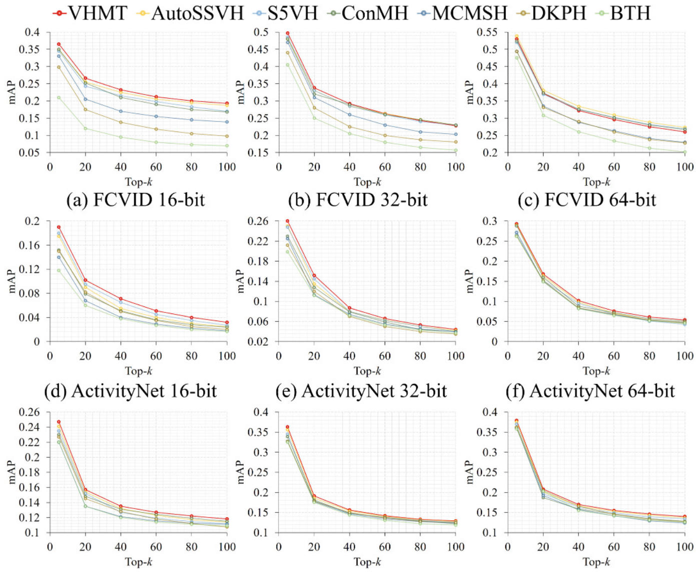
Training. Experiments are run on NVIDIA A800 GPUs with PyTorch 2.1.1 and CUDA 12.3. We use a batch size of 512 and train for 800 (FCVID), 500 (ActivityNet), and 40 (YFCC) epochs. The learning rate is initialized at 1×10^{-4}, reduced by 90% every 20 epochs, and clamped to a minimum of 1×10^{-6}. Optimization is performed using Adam [32].

Testing. Retrieval performance is evaluated on each dataset using mAP@k [7, 8]. The test sets consist of 45,600 (FCVID), 4,758 (ActivityNet), and 101,256 (YFCC) samples. The best and second-best results are highlighted in **bold** and *italic*, respectively.

4.4 Comparison with State-of-the-Arts

Baselines. We compare 6 influential open-source SSVH methods from the past 5 years: BTH [9], DKPH [17], MCMSH [19], ConMH [6], S5VH [7], and AutoSSVH [8].

Performance Under Standard Protocols. As illustrated in Fig. 3(a)–(i), the VHMT demonstrates competitive performance across three datasets and various hash lengths, with consistent improvements in mAP@k over the baselines. This performance improvement highlights the effectiveness of our method: (1) the proposed teacher-student network; (2) the TAM module; and (3) the novel loss function, which jointly contribute to enhanced retrieval accuracy.



(a) FCVID 16-bit (b) FCVID 32-bit (c) FCVID 64-bit

(d) ActivityNet 16-bit (e) ActivityNet 32-bit (f) ActivityNet 64-bit

(g) YFCC 16-bit (h) YFCC 32-bit (i) YFCC 64-bit

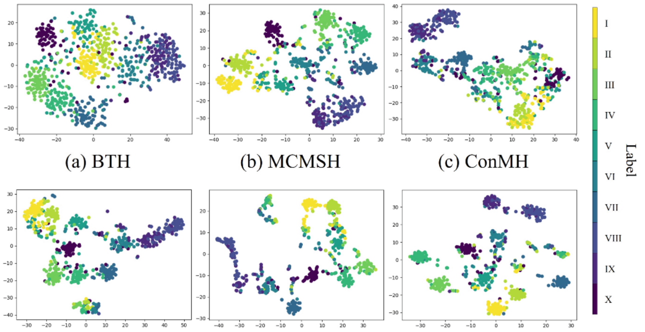
Fig. 3. VHMT vs. baselines: Performance (mAP@k) on three datasets with 16-bit, 32-bit, and 64-bit hash codes.

Cross-Dataset Transferability. As reported in Table 2, models are trained on FCVID and evaluated on YFCC. The VHMT shows a relatively smaller decline in mAP@20 compared to baselines, suggesting advantages in cross-dataset generalization.

Table 2. VHMT vs. baselines: Performance (mAP@20) on FCVID→YFCC with 64-bit hash codes and ↓drop (%).

BTH	DKPH	MCMSH	ConMH	S5VH	AutoSSVH	VHMT
0.191 ↓5.7	*0.199* ↓2.8	0.186 ↓3.2	0.188 ↓3.6	0.190 ↓3.6	0.194 ↓3.5	**0.201** ↓3.2

t-SNE Visualization. Figure 4(a)–(f) presents t-SNE visualizations [33] of hash codes on FCVID-small, constructed by sampling 10 classes with 80 videos each from FCVID following [6]. The VHMT achieves better class separation, more compact intra-class clustering, and a more uniform code distribution compared to baselines, thereby mitigating retrieval bias due to excessive false positives.



(a) BTH (b) MCMSH (c) ConMH

(d) S5VH (e) AutoSSVH (f) VHMT

Fig. 4. VHMT vs. baselines: t-SNE visualization of feature distributions on FCVID with 64-bit hash codes.

4.5 Ablation Study

Effectiveness of the VHMT Network. As shown in Table 3, deeper TAM modules (Exps. (I)–(V)) yield moderate GmAP gains, with encoding time increasing as the encoder size grows. We adopt the configuration of Exp. (III) for optimal efficiency. Using smoother wavelet bases (db4 in Exp. (VI), sym8 in Exp. (VII)) improves GmAP by 0.11% and 0.20%, while max pooling (Exp. (VIII)) and average pooling (Exp. (IX)) degrade performance by 4.66% and 3.23%, respectively, demonstrating the superiority of wavelets in extracting robust low-frequency semantics. Under the same setting as Exp. (III), extending the student model to a Siamese structure (Exp. (X)) results in a 4.27% lower GmAP, confirming the effectiveness of the teacher-student network.

Table 3. Performance (GmAP) and Encoding Time (ms) of the VHMT network scale on FCVID with 64-bit hash codes, where $E_{\mathrm{T}}/D_{\mathrm{T}}$ and $E_{\mathrm{S}}/D_{\mathrm{S}}$ denote the encoder/decoder sizes of the teacher (M^{T}) and student (M^{S}) models, respectively.

Exp.	Network	Technique	E_{T}	D_{T}	E_{S}	D_{S}	GmAP	Time (ms)
(I)	$M^{\mathrm{T}}+M^{\mathrm{S}}$	Haar	{4,2}	1	3	1	0.8461	1.62
(II)	$M^{\mathrm{T}}+M^{\mathrm{S}}$	Haar	{4,2,2}	1	6	1	0.8592	3.13
(III)	$M^{T}+M^{\mathrm{S}}$	Haar	{8,4,2}	1	6	1	0.8681	3.13
(IV)	$M^{\mathrm{T}}+M^{\mathrm{S}}$	Haar	{8,4,2}	2	12	2	0.8768	6.40
(V)	$M^{\mathrm{T}}+M^{\mathrm{S}}$	Haar	{16,8,4,2}	2	16	2	0.8839	8.64
(VI)	$M^{\mathrm{T}}+M^{\mathrm{S}}$	db4	{8,4,2}	1	6	1	0.8692	3.13
(VII)	$M^{\mathrm{T}}+M^{\mathrm{S}}$	sym8	{8,4,2}	1	6	1	0.8701	3.13
(VIII)	$M^{\mathrm{T}}+M^{\mathrm{S}}$	MaxPooling	{8,4,2}	1	6	1	0.8215	3.13
(IX)	$M^{\mathrm{T}}+M^{\mathrm{S}}$	AvePooling	{8,4,2}	1	6	1	0.8358	3.13
(X)	M^{S} Only	Haar	-	-	6	1	0.8254	3.13

Effectiveness of the TAM Module. As shown in Table 4, ablation studies evaluate the TAM module's components with other settings fixed. Combining MHA with MLP (Exp. (II)) improves performance over MHA alone (Exp. (I)), increasing mAP@5 from 0.504 to 0.519, while replacing MLP with KAN (Exp. (III)) degrades it, indicating KAN's limited fit for attention outputs. Substituting MHA with Mamba (Exp. (IV)) yields suboptimal results, suggesting the inductive bias of MHA is better aligned with the requirements of the teacher model. Integrating MHA with TRE (Exp. (V)) achieves the best mAP, validating their synergistic design.

Table 4. Performance (mAP@k) of TAM variants on FCVID with 64-bit hash codes.

Exp.	MHA	TRE	$k = 5$	$k = 20$	$k = 40$	$k = 60$	$k = 80$	$k = 100$
(I)	✓		0.504	0.353	0.306	0.281	0.266	0.247
(II)	✓	MLP	*0.519*	*0.366*	*0.318*	*0.292*	*0.274*	*0.258*
(III)	✓	KAN	0.500	0.350	0.303	0.279	0.262	0.245
(IV)	Mamba	✓	0.510	0.355	0.310	0.283	0.268	0.252
(V)	✓	✓	**0.530**	**0.372**	**0.322**	**0.296**	**0.275**	**0.260**

Effectiveness of the Loss Function. Table 5 shows that $\mathcal{L}_{\mathrm{rec}}+\mathcal{L}_{\mathrm{sim}}$ (Exp. (V)) improves mAP@5 from 0.497/0.381 to 0.507, $\mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{ctr}}$ (Exp. (VI)) raises it to 0.528, and the full loss $\mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{ctr}} + \mathcal{L}_{\mathrm{sim}}$ (Exp. (VII)) achieves the best performance across all mAP@k metrics, with mAP@5 reaching 0.530, confirming the complementarity of the loss terms.

Table 5. Performance (mAP@k) of loss variants on FCVID with 64-bit hash codes.

Exp.	Variant	$k = 5$	$k = 20$	$k = 40$	$k = 60$	$k = 80$	$k = 100$
(I)	$\mathcal{L}_{\text{rec}}$ Only	0.497	0.302	0.249	0.209	0.191	0.171
(II)	$\mathcal{L}_{\text{ctr}}$ Only	0.388	0.276	0.247	0.230	0.217	0.208
(III)	$\mathcal{L}_{\text{sim}}$ Only	0.381	0.255	0.214	0.193	0.181	0.170
(IV)	w/o $\mathcal{L}_{\text{rec}}$	0.406	0.291	0.259	0.242	0.227	0.219
(V)	w/o $\mathcal{L}_{\text{ctr}}$	0.507	0.310	0.244	0.212	0.194	0.171
(VI)	w/o $\mathcal{L}_{\text{sim}}$	*0.528*	*0.368*	*0.318*	*0.292*	*0.271*	*0.256*
(VII)	$\mathcal{L}_{\text{VHMT}}$	**0.530**	**0.372**	**0.322**	**0.296**	**0.275**	**0.260**

5 Conclusions

This paper has proposed a novel SSVH method called the VHMT based on a teacher-student architecture. The teacher model extracts robust low-frequency semantic features via DWT and captures temporal dependencies using a TAM, enhanced by a TRE block to improve fine-grained reconstruction and generate high-quality temporal modeling signals. The student model employs a Mamba architecture to model temporal relationships, achieving knowledge transfer and efficient single-path inference by approximating the teacher's feature reconstructions and attention outputs. A joint loss function enables the student to closely mimic the teacher's temporal modeling while simultaneously optimizing the clustering structure of the hash codes. Experiments have shown that the VHMT outperforms several SOTA methods in retrieval performance under the mAP metric.

Acknowledgements. This work is partially supported by the Guangxi Natural Science Foundation (2025GXNSFBA069370, 2025GXNSFBA069304), and the National Natural Science Foundation of China (62272111, 62302108).

References

1. Tang, Z., Chen, L., Zhang, X., Zhang, S.: Robust image hashing with tensor decomposition. IEEE Trans. Knowl. Data Eng. **31**(3), 549–560 (2019)
2. Liang, X., Tang, Z., Zhang, X., Yu, M., Zhang, X.: Robust hashing with local tangent space alignment for image copy detection. IEEE Trans. Dependable Secure Comput. **21**(4), 2448–2460 (2024)
3. Liang, X., Tang, Z., Zhang, X., Zhang, X., Yang, C.N.: Robust image hashing with weighted saliency map and laplacian eigenmaps. IEEE Trans. Inf. Forensics Secur. **20**, 665–676 (2025)
4. Song, J., Zhang, H., Li, X., Gao, L., Wang, M., Hong, R.: Self-supervised video hashing with hierarchical binary auto-encoder. IEEE Trans. Image Process. **27**(7), 3210–3221 (2018)

5. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a “siamese” time delay neural network. Adv. Neural Inf. Process. Syst. **6** (1993)
6. Wang, Y., Wang, J., Chen, B., Zeng, Z., Xia, S.T.: Contrastive masked autoencoders for self-supervised video hashing. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 3, pp. 2733–2741 (2023)
7. Wang, J., et al.: Efficient self-supervised video hashing with selective state spaces. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 7, pp. 7753–7761 (2025)
8. Lian, N., et al.: Autossvh: exploring automated frame sampling for efficient self-supervised video hashing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18881–18890 (2025)
9. Li, S., Li, X., Lu, J., Zhou, J.: Self-supervised video hashing via bidirectional transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13544–13553 (2021)
10. Li, Q., Tian, X., Ng, W.W.Y.: Self-supervised temporal sensitive hashing for video retrieval. IEEE Trans. Multimedia **26**, 9021–9035 (2024)
11. Du, L., Liang, X., Yang, L., Tang, Z.: Structure-preserving video hashing via self-supervised transformer for retrieval. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1–5 (2025)
12. Jiang, Y.G., Wu, Z., Wang, J., Xue, X., Chang, S.F.: Exploiting feature and class relationships in video categorization with regularized deep neural networks. IEEE Trans. Pattern Anal. Mach. Intell. **40**(2), 352–364 (2018)
13. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: Activitynet: a large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–970 (2015)
14. Thomee, B., et al.: YFCC100M: the new data in multimedia research. Commun. ACM **59**(2), 64–73 (2016)
15. Zhang, H., Wang, M., Hong, R., Chua, T.S.: Play and rewind: optimizing binary representations of videos by self-supervised temporal hashing. In: Proceedings of the ACM International Conference on Multimedia, pp. 781–790 (2016)
16. Li, C., Yang, Y., Cao, J., Huang, Z.: Jointly modeling static visual appearance and temporal pattern for unsupervised video hashing. In: Proceedings of the ACM Conference on Information and Knowledge Management, pp. 9–17 (2017)
17. Li, P., Xie, H., Ge, J., Zhang, L., Min, S., Zhang, Y.: Dual-stream knowledge-preserving hashing for unsupervised video retrieval. In: Proceedings of the European Conference on Computer Vision, pp. 181–197 (2022)
18. Tolstikhin, I.O., et al.: MLP-mixer: an all-MLP architecture for vision. Adv. Neural. Inf. Process. Syst. **34**, 24261–24272 (2021)
19. Hao, Y., Duan, J., Zhang, H., Zhu, B., Zhou, P., He, X.: Unsupervised video hashing with multi-granularity contextualization and multi-structure preservation. In: Proceedings of the ACM International Conference on Multimedia, pp. 3754–3763 (2022)
20. Duan, J., Hao, Y., Zhu, B., Cheng, L., Zhou, P., Wang, X.: Efficient unsupervised video hashing with contextual modeling and structural controlling. IEEE Trans. Multimedia **26**, 7438–7450 (2024)
21. Zeng, Z., Wang, J., Chen, B., Wang, Y., Xia, S.T., Intelligence, P.C.: Motion-aware graph reasoning hashing for self-supervised video retrieval. In: Proceedings of the British Machine Vision Conference, p. 82 (2022)

22. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2017)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
25. Gu, A., Dao, T.: Mamba: linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
26. Zou, W., Gao, H., Yang, W., Liu, T.: Wave-mamba: wavelet state space model for ultra-high-definition low-light image enhancement. In: Proceedings of the ACM International Conference on Multimedia, pp. 1534–1543 (2024)
27. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
28. Liu, Z., et al.: Kan: Kolmogorov-Arnold networks. In: International Conference on Learning Representations (2025)
29. Chuang, C.Y., Robinson, J., Lin, Y.C., Torralba, A., Jegelka, S.: Debiased contrastive learning. Adv. Neural. Inf. Process. Syst. **33**, 8765–8775 (2020)
30. Tammina, S.: Transfer learning using VGG-16 with deep convolutional neural network for classifying images. Int. J. Sci. Res. Publ. **9**(10), 143–150 (2019)
31. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vision **115**, 211–252 (2015)
32. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (2015)
33. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(11) (2008)