Efficient Self-Supervised Video Hashing with Selective State Spaces

Jinpeng Wang[1*], Niu Lian[2*], Jun Li[2*], Yuting Wang[1], Yan Feng[4], Bin Chen[2,3†], Yongbing Zhang[2], Shu-Tao Xia[1,3]

[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]Harbin Institute of Technology, Shenzhen
[3]Research Center of Artificial Intelligence, Peng Cheng Laboratory
[4]Meituan, Beijing
wjp20@mails.tsinghua.edu.cn, {220110904,220110924}@stu.hit.edu.cn, huangmozhi9527@gmail.com, fengyan14@meituan.com, chenbin2021@hit.edu.cn, ybzhang08@hit.edu.cn, xiast@sz.tsinghua.edu.cn

Abstract

Self-supervised video hashing (SSVH) is a practical task in video indexing and retrieval. Although Transformers are predominant in SSVH for their impressive temporal modeling capabilities, they often suffer from computational and memory inefficiencies. Drawing inspiration from Mamba, an advanced state-space model, we explore its potential in SSVH to achieve a better balance between efficacy and efficiency. We introduce S5VH, a Mamba-based video hashing model with an improved self-supervised learning paradigm. Specifically, we design bidirectional Mamba layers for both the encoder and decoder, which are effective and efficient in capturing temporal relationships thanks to the data-dependent selective scanning mechanism with linear complexity. In our learning strategy, we transform global semantics in the feature space into semantically consistent and discriminative hash centers, followed by a center alignment loss as a global learning signal. Our self-local-global (SLG) paradigm significantly improves learning efficiency, leading to faster and better convergence. Extensive experiments demonstrate S5VH's improvements over state-of-the-art methods, superior transferability, and scalable advantages in inference efficiency.

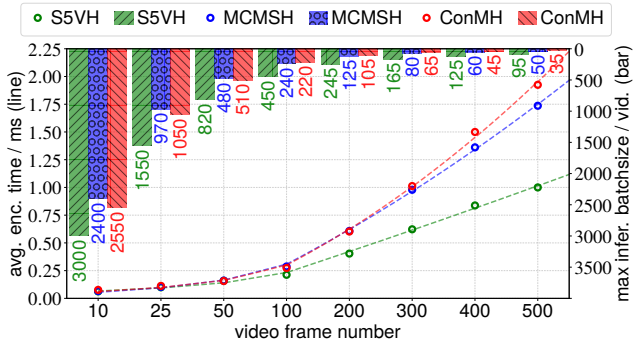
Code — https://github.com/gimpong/AAAI25-S5VH

Introduction

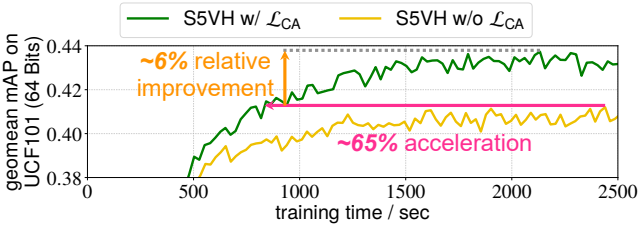
Content-based retrieval is a basic component in video search and recommendation. Hashing has been widely explored in this context to facilitate fast retrieval and reduce memory footprint (Gao et al. 2023; Sun et al. 2023b, 2024; Wang et al. 2024b). Video hashing has evolved from traditional methods with handcrafted features to advanced deep approaches with substantially improved retrieval performance, where self-supervised video hashing (SSVH) has gained increasing attention, given the ubiquity of large-scale unlabeled video data and evoked by the rapid progress of self-supervised learning in recent years (He et al. 2020).

In SSVH, temporal modeling is essential for video understanding and hash code learning. Early approaches (Song et al. 2018a; Li et al. 2019a) utilized nonlinear RNNs to process frame features, which suffered from gradient vanishing (or exploding) and struggled over long-range dependencies. In contrast, Transformer-based models (Li et al. 2021; Wang et al. 2023a) were shown to capture temporal semantics better, thanks to self-attentive interactions across all frames. Despite state-of-the-art retrieval performance, the quadratically growing complexity of Transformers in computation and memory to the frame number renders scalability limitations. Pursuing the optimal trade-off between efficacy and efficiency remains an under-explored problem.

Recent advances have sparked interest in state space models (SSMs) (Gu et al. 2021; Gupta, Gu, and Berant 2022),



(a) Our S5VH based on Mamba exhibits lower inference overheads on memory and computation. The efficiency advantages are scalable and more notable under larger frame numbers.



(b) The new global learning signal in the hash space effectively enhances training efficiency, showing faster and better convergence.

Figure 1: Highlights of this paper.

*The first three authors contributed equally to this work.
†Corresponding author.

where we find Mamba (Gu and Dao 2023), an improved variant of structured SSMs (S4) (Gu, Goel, and Ré 2021), can bring insightful solutions to the above problem. In detail, the data-dependent selective mechanism in Mamba ensures focusing on essential information while filtering out irrelevant noises regarding the input, which helps to capture temporal dynamics and understand long-range relations in videos. Moreover, Mamba's linear complexity regarding sequence length promises superior scalability for video processing, and its GPU-friendly implementation also aligns with SSVH's pursuit of efficient inference. These advantages motivate us to explore Mamba's potential for SSVH.

In this paper, we introduce **S5VH**[1], the first selective state-space approach for SSVH. For *network design*, we explore a **Mamba-based hashing network**, where each encoder or decoder layer comprises a bidirectional Mamba module for effective and efficient temporal modeling. For *self-supervised hash learning*, we develop a **self-local-global (SLG) paradigm** that excavates hierarchical learning signals to enhance training efficiency. Our motivation stems from the Mamba's serial nature that restricts training throughout (Liu et al. 2024), and we resort to a practical and hashing-oriented solution by maximizing sample efficiency.

State-of-the-art approaches (Wang et al. 2023a; Wei et al. 2023; Li, Tian, and Ng 2024) typically follow the *intra*- and *inter*-sample learning paradigm, where intra-sample (*i.e.*, *self*) signals refer to recovery tasks with various data augmentations for video understanding, while the inter-sample signals refer to contrastive tasks between videos to obtain discriminative hash codes. We note that inter-sample signals are subject to *local* signals reflected by individual samples, whose efficiency is limited by the (negative) sampling. We further integrate a *global* signal in the Hamming space to complement the learning paradigm. We start with clustering video feature space to summarize global semantic structure by cluster-level similarity. Then, we introduce a novel hash center generation algorithm to transform the global semantics into well-separated and semantically consistent hash centers. Based on them, we propose a center alignment loss to align the hash codes of each training sample to its associated hash center, supplementing global semantic guidance and enhancing sample efficiency.

We conduct extensive experiments on 4 datasets: ActivityNet, FCVID, UCF101, and HMDB51, demonstrating that S5VH outperforms state-of-the-art baselines under various setups and transfers better across datasets. Regarding inference efficiency, S5VH exhibits notable advantages, including lower memory overhead, which allows for larger batch-sizes, and faster computation. As shown in Figure 1(a), these advantages become more pronounced as the frame number increases. Additionally, we provide comprehensive ablations and analyses, focusing on network architecture and training strategy. The results verify Mamba's superiority in SSVH and confirm the necessity of the global signal in the SLG learning paradigm. In particular, Figure 1(b) shows that the proposed center alignment loss, which serves as the global signal, can guide the training process toward faster and better convergence.

To sum up, our paper makes the following contributions.

- We explore the first Mamba-based SSVH model, indicating a superior solution for both efficacy and efficiency.
- We design a hash center generation algorithm that computes semantically consistent and discriminative hash centers from the feature-space global semantics.
- We propose a center alignment loss as a global learning signal, contributing to a solid self-local-global (SLG) paradigm and improving training efficiency.

Related Works

Self-Supervised Video Hashing

Video hashing focuses on learning binary codes to enable fast, memory-efficient video retrieval. Self-supervised video hashing (SSVH) is particularly valuable for applications where labels are scarce. Early approaches often overlooked the temporal dynamics of videos, relying on traditional techniques like ITQ (Gong et al. 2012), SH (Weiss, Torralba, and Fergus 2008), and MFH (Song et al. 2011). VHDT (Ye et al. 2013) addressed this gap and showed notable improvement.

Recent methods focused on deep models, with progress in network design and learning strategy. While RNNs prevailed (Zhang et al. 2016; Li et al. 2017; Song et al. 2018a) initially, newer research (Li et al. 2021; Wang et al. 2023a; Li, Tian, and Ng 2024) has favored Transformers (Vaswani et al. 2017) for superior performance. Other remarkable contributions include the use of MLP-Mixer (Tolstikhin et al. 2021) in MCMSH (Hao et al. 2022) and EUVH (Duan et al. 2024), as well as the incorporation of graph networks (Veličković et al. 2018) in MAGRH (Zeng et al. 2022).

On learning strategy, existing methods can be classified into 4 categories: (**i**) *Self-recovery signals*, such as auto-regressive frame reconstruction (Zhang et al. 2016; Song et al. 2018a; Li et al. 2019a), separated reconstructions of appearance and temporal dynamics (Li et al. 2017), masked frame recovery (Li et al. 2021, 2022; Wang et al. 2023a), and temporal order prediction (Wei et al. 2023). (**ii**) *Inter-sample **local** signals*, including pairwise similarity preservation (Hao et al. 2017; Song et al. 2018a; Li et al. 2019b, 2021) and contrastive learning (Wang et al. 2023a; Duan et al. 2024). (**iii**) *Regularization signals*, including hashing regularization techniques like minimizing quantization error, bit decorrelation, and bit balance (Wu et al. 2017), as well as novel methods such as self-distillation (Li et al. 2022), temporal sensitivity regularization (Li, Tian, and Ng 2024), bit-wise distribution prediction (Li et al. 2019a; Wang et al. 2023b), and feature-space cluster alignment as an auxiliary task (Li et al. 2021; Hao et al. 2022). (**iv**) *Multi-modal signals*, where advanced methods incorporated extra modalities like motion (Zeng et al. 2022; Shen et al. 2023) or audio (Zhou et al. 2024), benefiting from cross-modal alignment.

Our work presents two key contributions: (**i**) In network design, we are the first to explore the novel Mamba architecture (Gu and Dao 2023) in SSVH, achieving an optimal balance between performance and efficiency. (**ii**) In hash learning strategy, we introduce a *global* signal through semantic hash center generation and a center alignment loss. Un-

[1] Self-Supervised Selective State-Space Video Hashing.

like previous methods (Li et al. 2021; Hao et al. 2022; Duan et al. 2024) that employed *feature-space* cluster alignment as auxiliary regularization, our *hash-space* center alignment is more direct and effective.

State Space Models

State space models (SSMs), originating from control theory (Kalman 1960), have emerged as a powerful framework for sequence modeling. Recent research has focused on linear SSMs to improve efficiency, yielding representative works like HiPPO (Gu et al. 2020) and LSSL (Gu et al. 2021). Based on this progress, the S4 model (Gu, Goel, and Ré 2021) set a milestone with successful performance across various sequence tasks. Several following works further optimized S4 to balance efficacy and efficiency by replacing the diagonal plus low-rank structure with a simpler diagonal matrix (Gu et al. 2022). Besides, many efforts have been devoted to hardware-efficient implementation. For example, the S5 model (Smith, Warrington, and Linderman 2022) incorporated a MIMO implementation and efficient parallel scanning techniques. H3 (Fu et al. 2022) addressed efficiency and performance gaps between SSMs and Transformers in language tasks, by proposing a fast FlashConv operator and a novel state passing algorithm.

Among the advances in SSMs, Mamba (Gu and Dao 2023) made significant strides with a novel data-dependent selective mechanism and hardware-efficient implementation. The past few months have seen emerging interest in various Mamba-base applications, including but not limited to vision (Liu et al. 2024; Zhu et al. 2024), multi-modality (Qiao et al. 2024; Zhao et al. 2024), and graph (Wang et al. 2024a). Inspired by these successes, we take the first exploration of Mamba's potential in video hashing.

Method

Problem Formulation and Overview of S5VH

Suppose there is an unlabeled video corpus, $\mathcal{C} = \{\boldsymbol{F}_i\}_{i=1}^{N}$, where $\boldsymbol{F}_i \in \mathbb{R}^{N_t \times D}$ is the frame feature collection of the i-th video, extracted by pre-trained 2D CNNs (Simonyan and Zisserman 2014; He et al. 2016). N_t and D denote frame number and feature dimension, respectively. Self-supervised video hashing (SSVH) aims to take $\boldsymbol{F}_i$ as input and generate a hash vector $\boldsymbol{b}_i \in \{-1, +1\}^K$, such that the Hamming distance can precisely reflect the semantic similarity. For this task, we propose **S5VH**, a Mamba-based network trained with an improved paradigm, as illustrated in Figure 2.

Mamba-based Video Hash Network

Preliminaries: State-Space Models (SSMs) and Mamba SSMs map input $x(t) \in \mathbb{R}^L$ to output $y(t) \in \mathbb{R}^L$ via the hidden state $h(t) \in \mathbb{R}^N$. Here, $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ defines the hidden state's evolution, while $\boldsymbol{B} \in \mathbb{R}^{N \times 1}$ and $\boldsymbol{C} \in \mathbb{R}^{1 \times N}$ represent the input and output mappings, respectively. We can express it by linear ordinary differential equations (ODEs):

$$\begin{aligned} \boldsymbol{h}'(t) &= \boldsymbol{A}\boldsymbol{h}(t) + \boldsymbol{B}\boldsymbol{x}(t), \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{h}(t). \end{aligned} \tag{1}$$

Modern SSMs approximate ODEs by discretizing $\boldsymbol{A}$ and $\boldsymbol{B}$ using a timescale Δ, through zero-order hold:

$$\bar{\boldsymbol{A}} = \exp(\Delta \boldsymbol{A}), \tag{2}$$

$$\bar{\boldsymbol{B}} = (\Delta \boldsymbol{A})^{-1}(\exp(\Delta \boldsymbol{A}) - \boldsymbol{I}) \cdot \Delta \boldsymbol{B}, \tag{3}$$

so that we obtain the discretized version of Equation (1):

$$\begin{aligned} \boldsymbol{h}_t &= \bar{\boldsymbol{A}}\boldsymbol{h}_{t-1} + \bar{\boldsymbol{B}}\boldsymbol{x}_t, \\ \boldsymbol{y}_t &= \boldsymbol{C}\boldsymbol{h}_t. \end{aligned} \tag{4}$$

Mamba (Gu and Dao 2023) introduced data dependence to Δ, $\boldsymbol{B}$, and $\boldsymbol{C}$, enabling input-aware selection for improved modeling. Despite limitations in parallelism due to recurrence, Mamba enhanced efficiency through structural reparameterization and parallel scanning algorithms.

Bidirectional Mamba Layers To effectively extract semantic information in videos, we build the temporal encoder $\mathcal{E}_t$ and decoder $\mathcal{D}_t$ with multi-layer bidirectional Mamba (Gu and Dao 2023) layers. As illustrated in Figure 2,each layer is composed of a *forward* Mamba block and a *backward* (*i.e.*, *reverse*) Mamba block, which is formulated by

$$\boldsymbol{S}_{\text{out}} = \overrightarrow{\text{Mamba}}(\boldsymbol{S}_{\text{in}}) + \overleftarrow{\text{Mamba}}(\boldsymbol{S}_{\text{in}}). \tag{5}$$

Here, $\rightarrow$ and $\leftarrow$ mark the forward and the reverse scans, respectively. $\boldsymbol{S}_{\text{in}}$ and $\boldsymbol{S}_{\text{out}}$ denote the input and the output hidden states, respectively. Each Mamba block adopts a gated structure with two branches, for example, the forward block:

$$\overrightarrow{\boldsymbol{S}}_{\text{out}} = \text{Linear}_3(\overrightarrow{\boldsymbol{S}}' \otimes \boldsymbol{S}''), \tag{6}$$

$$\overrightarrow{\boldsymbol{S}}' = \text{LN}_2(\overrightarrow{\text{SSM}}(\sigma(\text{Conv}(\text{Linear}_1(\text{LN}_1(\boldsymbol{S}_{\text{in}})))))), \tag{7}$$

$$\boldsymbol{S}'' = \sigma(\text{Linear}_2(\boldsymbol{S}_{\text{in}})), \tag{8}$$

where $\overrightarrow{\boldsymbol{S}}'$ and $\boldsymbol{S}''$ are hidden states of the main and the gating branches, respectively. $\otimes$ denotes the Hadamard product. LN denotes layer normalization. σ denotes the SiLU activation (Ramachandran, Zoph, and Le 2017). $\overrightarrow{\text{SSM}}$ denotes the forward selective scan module. Conv denotes the 1D convolution. Linear denotes learnable linear projection.

Hash Layer We design a hash layer upon encoder's output to transform visual embeddings into compact hash vectors. Given the encoded embeddings of the i-th video, $\boldsymbol{E}_i \in \mathbb{R}^{N_t \times D}$, we obtain K-dimensional soft hash vectors by

$$\boldsymbol{H}_i = \tanh(\text{Linear}(\boldsymbol{E}_i)) \in (-1, +1)^{N_t \times K}, \tag{9}$$

where $\tanh$ denotes the hyperbolic tangent function. To establish video-level hash codes, we aggregate the frame hash codes by mean pooling and the sign function, namely

$$\boldsymbol{b}_i = \text{sign}\left(\frac{1}{N_t}\sum_{j=1}^{N_t} \boldsymbol{H}_i[j]\right) \in \{-1, +1\}^K. \tag{10}$$

For end-to-end training, we pass the gradient through (Bengio, Léonard, and Courville 2013) the sign function.

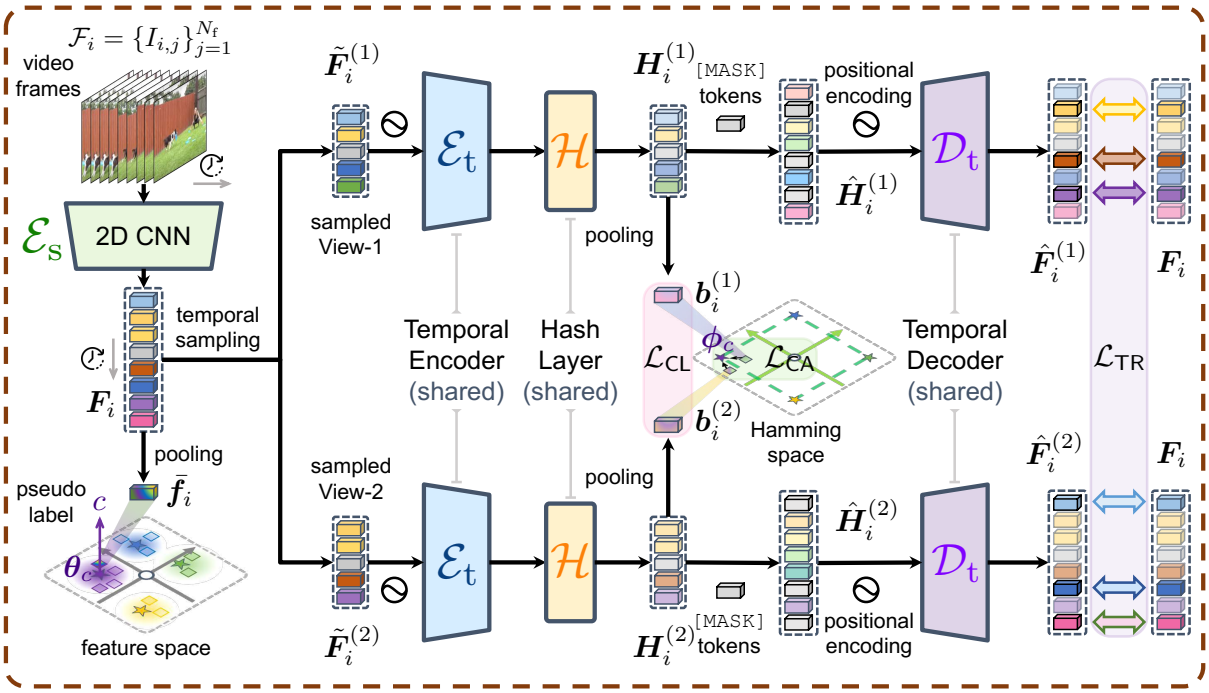


Figure 2: Overview of S5VH.

Semantic Hash Center Generation

Existing SSVH methods found it hard to use feature-space semantics as effectively as in supervised scenarios. They can only take inter-sample similarity as proxy or regularize feature space to assist hash learning, showing limited training efficiency. We break this dilemma with a more direct solution, exploring global categorical semantics for improved hash learning. For this purpose, we introduce how to (**i**) *extract* and (**ii**) *transform* such implicit information into the hash space. This process can be done before model training.

Global Semantic Structure Extraction Considering the large-scale clustering on video corpus, rather than all frame features, we use the temporally averaged features of videos to reduce the order of magnitude. By the k-means algorithm, we obtain N_c cluster centroids, $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1; \boldsymbol{\theta}_2; \cdots; \boldsymbol{\theta}_{N_c}]$, $\boldsymbol{\theta}_* \in \mathbb{R}^D$, which can be regarded as the compression of the corpus and encode the global semantic structure.

Optimization for Hash Center Generation We further transform $\boldsymbol{\Theta}$ into hash centers, which are expected to be (**i**) semantically consistent with the feature space, and (**ii**) well separated from each other to encourage discriminative hash codes. Let us translate these requirements into objectives:

$$\min_{\boldsymbol{\Phi}} \overbrace{\left\|\boldsymbol{\Phi}\boldsymbol{\Phi}^\top - K\boldsymbol{W}\right\|_{\mathrm{F}}^2}^{\text{semantic consistency}} + \overbrace{\frac{1}{2}\sum_{i,j}^{N_c} \boldsymbol{\phi}_i^\top \boldsymbol{\phi}_j}^{\text{separation}},$$

$$\text{s.t. } \boldsymbol{\Phi} = [\boldsymbol{\phi}_1; \boldsymbol{\phi}_2; \cdots; \boldsymbol{\phi}_{N_c}] \in \{-1, +1\}^{N_c \times K}. \quad (11)$$

$\boldsymbol{W}_{ij} = \cos(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ is the feature-space semantic similarity. $\boldsymbol{\Phi}$ denotes the desired hash center collection.

It is NP-hard to optimize Equation (11) for the binary constraint of hash centers. Fortunately, we take inspiration from Wu and Ghanem (2018) that $\boldsymbol{\Phi} \in \{-1, +1\}^{N_c \times K}$ is equivalent to $\boldsymbol{\Phi} \in [-1, +1]^{N_c \times K} \cap \{\boldsymbol{\Phi} \mid \|\boldsymbol{\Phi}\|_p^p = N_c K\}$ and adopt the ℓ_p-box ADMM algorithm to solve Equation (11).

Concretely, we first introduce two auxiliary variables $\boldsymbol{\Psi}_b$ and $\boldsymbol{\Psi}_p$ associated with the constrains $\mathcal{S}_b \equiv [-1, +1]^{N_c \times K}$ and $\mathcal{S}_p \equiv \{\boldsymbol{\Psi}_p \mid \|\boldsymbol{\Psi}_p\|_p^p = N_c K\}$, respectively. Then, we solve the following problem with $p = 2$ for simplicity:

$$\begin{aligned} \min_{\boldsymbol{\Phi}, \boldsymbol{\Psi}_*, \boldsymbol{\Upsilon}_*} & \left\|\boldsymbol{\Phi}\boldsymbol{\Phi}^\top - K\boldsymbol{W}\right\|_{\mathrm{F}}^2 + \frac{1}{2}\sum_{i,j}^{N_c} \boldsymbol{\phi}_i^\top \boldsymbol{\phi}_j \\ & + \delta_{\mathcal{S}_b}(\boldsymbol{\Psi}_b) + \delta_{\mathcal{S}_p}(\boldsymbol{\Psi}_p) \\ & + \frac{\mu_b}{2}\left\|\boldsymbol{\Phi} - \boldsymbol{\Psi}_b\right\|_{\mathrm{F}}^2 + \frac{\mu_p}{2}\left\|\boldsymbol{\Phi} - \boldsymbol{\Psi}_p\right\|_{\mathrm{F}}^2 \\ & + \mathrm{Tr}(\boldsymbol{\Upsilon}_b^\top(\boldsymbol{\Phi} - \boldsymbol{\Psi}_b)) + \mathrm{Tr}(\boldsymbol{\Upsilon}_p^\top(\boldsymbol{\Phi} - \boldsymbol{\Psi}_p)). \end{aligned} \quad (12)$$

$\delta_{\mathcal{S}}(\boldsymbol{\Psi})$ is an indicator that outputs 0 if $\boldsymbol{\Psi} \in \mathcal{S}$ else $+\infty$. $\boldsymbol{\Upsilon}_*$ and μ_* are the dual and penalty variables, respectively. Next, the optimization process follows Wu and Ghanem (2018).

Update $\boldsymbol{\Phi}$: We fix all variables except for $\boldsymbol{\Phi}$ at the $(k+1)$-th iteration. $\boldsymbol{\Phi}^{k+1}$ is updated by

$$\min_{\boldsymbol{\Phi}} \left\|\boldsymbol{\Phi}\boldsymbol{\Phi}^\top - K\boldsymbol{W}\right\|_{\mathrm{F}}^2 + \frac{1}{2}\sum_{i,j}^{N_c} \boldsymbol{\phi}_i^\top \boldsymbol{\phi}_j + \frac{\mu_b + \mu_p}{2}\left\|\boldsymbol{\Phi}\right\|_{\mathrm{F}}^2 + \mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{G}^\top), \quad (13)$$

where $\boldsymbol{G} = \boldsymbol{\Upsilon}_b^k + \boldsymbol{\Upsilon}_p^k - \mu_b\boldsymbol{\Psi}_b^k - \mu_p\boldsymbol{\Psi}_p^k$. The gradient can be calculated by the LBFGS-B method as

$$4(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top - K\boldsymbol{W})\boldsymbol{\Phi} + \mathbf{1}\mathbf{1}^\top\boldsymbol{\Phi} + (\mu_b + \mu_p)\boldsymbol{\Phi} + \boldsymbol{G}. \quad (14)$$

Update $\boldsymbol{\Psi}_*$: We fix $\boldsymbol{\Phi}^{k+1}$ and $\boldsymbol{\Upsilon}_b^k$, updating $\boldsymbol{\Psi}_b^{k+1}$ by

$$\min_{\boldsymbol{\Psi}_b} \delta_{\mathcal{S}_b}(\boldsymbol{\Psi}_b) + \frac{\mu_b}{2}\left\|\boldsymbol{\Psi}_b - \boldsymbol{\Phi}^{k+1}\right\|_{\mathrm{F}}^2 - \mathrm{Tr}(\boldsymbol{\Upsilon}_b^{k\top}\boldsymbol{\Psi}_b), \quad (15)$$

which can be easily solved with the proximal minimization method, yielding the closed-form solution:

$$\boldsymbol{\Psi}_b^{k+1} = \sqrt{N_c K} \times \frac{\boldsymbol{\Phi}^{k+1} + \boldsymbol{\Upsilon}_b^k/\mu_b}{\left\|\boldsymbol{\Phi}^{k+1} + \boldsymbol{\Upsilon}_b^k/\mu_b\right\|_{\mathrm{F}}}. \tag{16}$$

Updating $\boldsymbol{\Psi}_p^{k+1}$ follows the analogous procedure.

Update $\boldsymbol{\Upsilon}_*$: We update them by gradient ascent:

$$\boldsymbol{\Upsilon}_b^{k+1} = \boldsymbol{\Upsilon}_b^k + \eta\mu_b(\boldsymbol{\Phi}^{k+1} - \boldsymbol{\Psi}_b^{k+1}), \tag{17}$$

$$\boldsymbol{\Upsilon}_p^{k+1} = \boldsymbol{\Upsilon}_p^k + \eta\mu_p(\boldsymbol{\Phi}^{k+1} - \boldsymbol{\Psi}_p^{k+1}), \tag{18}$$

where η is the learning rate. The above optimization process alternates between each variable until convergence.

Self-Local-Global (SLG) Learning Paradigm

To enhance hash learning, we faithfully leverage hierarchical learning signals in different considerations, including (**i**) temporal reconstruction as *self-recovery* signal to capture relations in temporal dynamics, (**ii**) contrastive learning as *inter-sample local* signal for discriminative hash codes, and (**iii**) hash center alignment as *inter-sample global* signal to prompt faster, better convergence. They each play indispensable roles in efficient and effective learning.

Temporal Reconstruction Following Wang et al. (2023a), we reconstruct the masked frame features from the frame-level hash codes to maximize their semantic capacity. Specifically, we take the frame hash codes of the augmented view n of the i-th video, namely, $\boldsymbol{H}_i^{(n)}$, as a showcase. It has dropped the frames associated with indices $\mathcal{M}_i^{(n)}$ during data augmentation. We first insert the `[mask]` token (a learnable vector in $\mathbb{R}^K$) to $\boldsymbol{H}_i^{(n)}$ according to $\mathcal{M}_i^{(n)}$ and obtain the decoder input, $\hat{\boldsymbol{H}}_i^{(n)}$. Then we process it with the temporal decoder $\mathcal{D}_t$ and get decoded features by $\hat{\boldsymbol{F}}_i^{(n)} = \mathcal{D}_t(\hat{\boldsymbol{H}}_i^{(n)})$. Finally, we compute reconstruction loss for the masked frames as

$$\mathcal{L}_{\mathsf{TR}}^{(n)} = \frac{1}{|\mathcal{M}_i^{(n)}|} \sum_{m \in \mathcal{M}_i^{(n)}} \|\hat{\boldsymbol{F}}_i^{(n)}[m] - \boldsymbol{F}_i[m]\|_2^2. \tag{19}$$

Contrastive Learning We contrastively align video-level hash codes between views, with a temperature factor $\tau > 0$:

$$\mathcal{L}_{\mathsf{CL}} = -\log \frac{\exp(\cos(\boldsymbol{b}_i^{(1)}, \boldsymbol{b}_i^{(2)})/\tau)}{\sum_j \exp(\cos(\boldsymbol{b}_i^{(1)}, \boldsymbol{b}_j^{(2)})/\tau)} \cdot \frac{\exp(\cos(\boldsymbol{b}_i^{(1)}, \boldsymbol{b}_i^{(2)})/\tau)}{\sum_j \exp(\cos(\boldsymbol{b}_j^{(1)}, \boldsymbol{b}_i^{(2)})/\tau)}. \tag{20}$$

Hash Center Alignment We obtain the pseudo label c_i for the i-th video by clustering its temporally averaged features, then align the video hash codes $\boldsymbol{b}_i^{(n)}$ to the hash center of c_i:

$$\begin{aligned} \mathcal{L}_{\mathsf{CA}}^{(n)} &= -\log \frac{\exp(\cos(\boldsymbol{\phi}_{c_i}, \boldsymbol{b}_i^{(n)})/\tau)}{\sum_{c=1}^{N_c} \exp(\cos(\boldsymbol{\phi}_c, \boldsymbol{b}_i^{(n)})/\tau)} \\ &= -\log \frac{\exp(\boldsymbol{\phi}_{c_i}^\top \boldsymbol{b}_i^{(n)}/K\tau)}{\sum_{c=1}^{N_c} \exp(\boldsymbol{\phi}_c^\top \boldsymbol{b}_i^{(n)}/K\tau)}. \end{aligned} \tag{21}$$

Total Learning Objectives

$$\mathcal{L}_{\text{S5VH}} = \frac{1}{2}(\mathcal{L}_{\mathsf{TR}}^{(1)} + \mathcal{L}_{\mathsf{TR}}^{(2)}) + \alpha\mathcal{L}_{\mathsf{CL}} + \frac{\beta}{2}(\mathcal{L}_{\mathsf{CA}}^{(1)} + \mathcal{L}_{\mathsf{CA}}^{(2)}). \tag{22}$$

$\alpha, \beta > 0$ are hyperparameters to balance learning signals.

Experiments

Experimental Setup

Datasets We conduct experiments on 4 benchmark datasets. (**i**) **ActivityNet** (Caba Heilbron et al. 2015) contains 200 activity categories of recognition. We follow the standard setup as in Wang et al. (2023a), using 9,722 videos for training. We uniformly sample 1,000 videos across 200 categories in the validation set as queries, and the remaining 3,758 videos as the database. (**ii**) **FCVID** (Jiang et al. 2017) contains 91,223 videos across 239 categories. We follow Song et al. (2018b) to use 45,585 videos for training and 45,600 videos for the retrieval database and queries. (**iii**) **UCF101** (Soomro, Zamir, and Shah 2012) consists of 13,320 videos from 101 human actions. We use 9,537 videos for training and the database, and 3,783 videos from the test set as the query set. (**iv**) **HMDB51** (Kuehne et al. 2011) comprises 6,849 videos across 51 actions. We use 3,570 videos for both training and database and 1,530 videos from the test set are designated as the query set.

Metrics Following previous works (Hao et al. 2022; Wang et al. 2023a), we use mean Average Precision at top-N results (**mAP@N**) as the metric, namely

$$\textbf{mAP@}N = \frac{1}{Q}\sum_{q=1}^{Q} \text{AP@}N(q). \tag{23}$$

Here, Q is the number of queries in evaluation. AP@$N(q)$ means the Average Precision for the q-th query, defined by

$$\text{AP@}N = \frac{1}{|\text{Rel}(N)|}\sum_{n=1}^{N} P(n)\cdot r(n), \tag{24}$$

where $|\text{Rel}(N)|$ is the total amount of relevant items. $P(n)$ denotes the precision at the n-th position. $r(n)$ is the relevance of the n-th ranked item (0: irrelevant; 1: relevant). We set $N \in \{5, 20, 40, 60, 80, 100\}$ as showcases. We further compute the geometric mean of these showcases, denoted as **GmAP** for simplicity, to show the holistic performance:

$$\textbf{GmAP} = \sqrt{\sum_{N\in\{5,20,40,60,80,100\}} (\text{mAP@}N)^2}. \tag{25}$$

Additionally, we use **Precision-Recall (PR)** curves to illustrate the detailed performance.

Implementation Details

Frame Encoding For ActivityNet, we sample 30 frames per video and use ResNet-50 (He et al. 2016) to extract 2048-D features. Both CNNs are pre-trained on ImageNet. For UCF101, HMDB51, and FCVID, we uniformly sample 25 frames per video and use VGG-16 (Simonyan and Zisserman 2014) to extract 4096-D features.

Model Configurations Considering the bidirectional layers yield double cost, to keep a comparable model size to baselines (Wang et al. 2023a), we set the 6 layers for the encoder and 1 layer for the decoder. The latent dimensions of the encoder and decoder are set to 256 and 192, respectively.

SSVH	BTH	DKPH	MCMSH	ConMH	BerVAE	**S5VH**
0.0187	0.0210	0.0243	0.0229	0.0285	0.0335	**0.0378**

Table 1: Cross-dataset retrieval performance by GmAP. We train on UCF101 and test on HMDB51 using 16-bit models.

Training Configurations For the model training, we choose the AdamW optimizer with default parameters in Pytorch, and employ a cosine annealed learning rate scheduling from $5e^{-4}$ to $1e^{-5}$. The models are trained for up to 350 epochs with 5-patience early-stopping to prevent overfitting. The default hyperparameter configurations are as below: (**i**) We set the mask ratio $\rho = |\mathcal{M}|/N_t$ to 0.75 on the FCVID dataset and 0.5 on the rest of the datasets. (**ii**) The temperature factor τ in Equations (20) and (21) is set 0.5. (**iii**) The number of semantic centers N_c is set to 450 on FCVID and 100 on the other datasets.

Comparison with State-of-the-arts

Baselines We select 6 representative baselines for comparison: SSVH (Song et al. 2018a), BTH (Li et al. 2021), DKPH (Li et al. 2022), MCMSH (Hao et al. 2022), ConMH (Wang et al. 2023a) and BerVAE (Wang et al. 2023b). We have discussed them in the Related Works section.

Performance under Standard Protocols As illustrated in Figure 3, S5VH generally outperforms other methods across datasets and code lengths, demonstrating a superior efficacy. In particular, the improvements are more pronounced with lower-bit settings such as 16 bits, highlighting S5VH's advantages in scenarios where high top-ranked results and retrieval speed are crucial.

Cross-Dataset Transferability We evaluate the cross-dataset retrieval performance of different methods. Specifically, we train them on UCF101 and test them on HMDB51. Table 1 presents the holistic results of different methods with 16-bit hash codes in the cross-dataset scenario. S5VH continues to outperform existing models, showcasing its exceptional ability to generalize across diverse video data.

Inference Efficiency Inference efficiency is a crucial aspect of practical retrieval systems. Here we focus on inference time and memory overheads for producing video hash codes. We compare S5VH with 2 representative baselines, *i.e.*, the Transformer-based model ConMH, and the MCMSH based on MLP-Mixer (Tolstikhin et al. 2021). We perform *stress testing* with them in the same computational environment, taking 5 samples as a unit to probe the maximally affordable batchsizes and measuring the average inference time per sample. Since efficiency is independent of the retrieval performance, we directly stimulate tensors in various lengths as the testing input. The results are shown in Figure 1(b). It is clear that S5VH enjoys a larger inference throughput and faster processing speed. Moreover, S5VH exhibits ever-pronounced advantages with longer sequences.

Based on the measured values, we use the least squares estimation to fit the scaling laws of inference time T (in ms) *w.r.t.* input length L. The functions of the 3 models are given by $\hat{T}_{\text{ConMH}} = 2.4 \times 10^{-6}L^2 + 1.9 \times 10^{-3}L + 3.0 \times 10^{-2}$ ($R^2 \approx 0.999$); $\hat{T}_{\text{MCMSH}} = 1.3 \times 10^{-6}L^2 + 2.1 \times 10^{-3}L + 2.3\times10^{-2}$ ($R^2 \approx 0.999$); $\hat{T}_{\text{S5VH}} = 1.5\times10^{-3}L+3.4\times10^{-2}$ ($R^2 \approx 0.998$). We note that MLP-Mixer has linear complexity, but MCMSH focuses more on using its structure to boost performance, where the improved designs result in overall quadratic complexity. Different from ConMH and MCMSH, S5VH enjoys a preferable linear complexity.

ID	Method	UCF101			HMDB51		
		16bit	32bit	64bit	16bit	32bit	64bit
(0)	**S5VH**	**.357**	**.390**	**.440**	**.093**	**.130**	**.142**
(1)	$\rightarrow$ Only	.351	.375	.424	.079	.123	.136
(2)	$\leftarrow$ Only	.324	.337	.390	.070	.120	.133
(3)	$\mathcal{L}_{\mathsf{TR}}$ Only	.138	.210	.280	.045	.075	.081
(4)	w/o $\mathcal{L}_{\mathsf{CA}}$	.286	.342	.407	.071	.100	.118
(5)	w/o $\mathcal{L}_{\mathsf{CL}}$	.204	.217	.290	.070	.095	.102
(6)	w/ LSTM	.303	.351	.432	.084	.125	.137
(7)	w/ RetNet	.278	.355	.413	.079	.108	.134
(8)	w/ RWKV	.307	.340	.408	.074	.106	.135

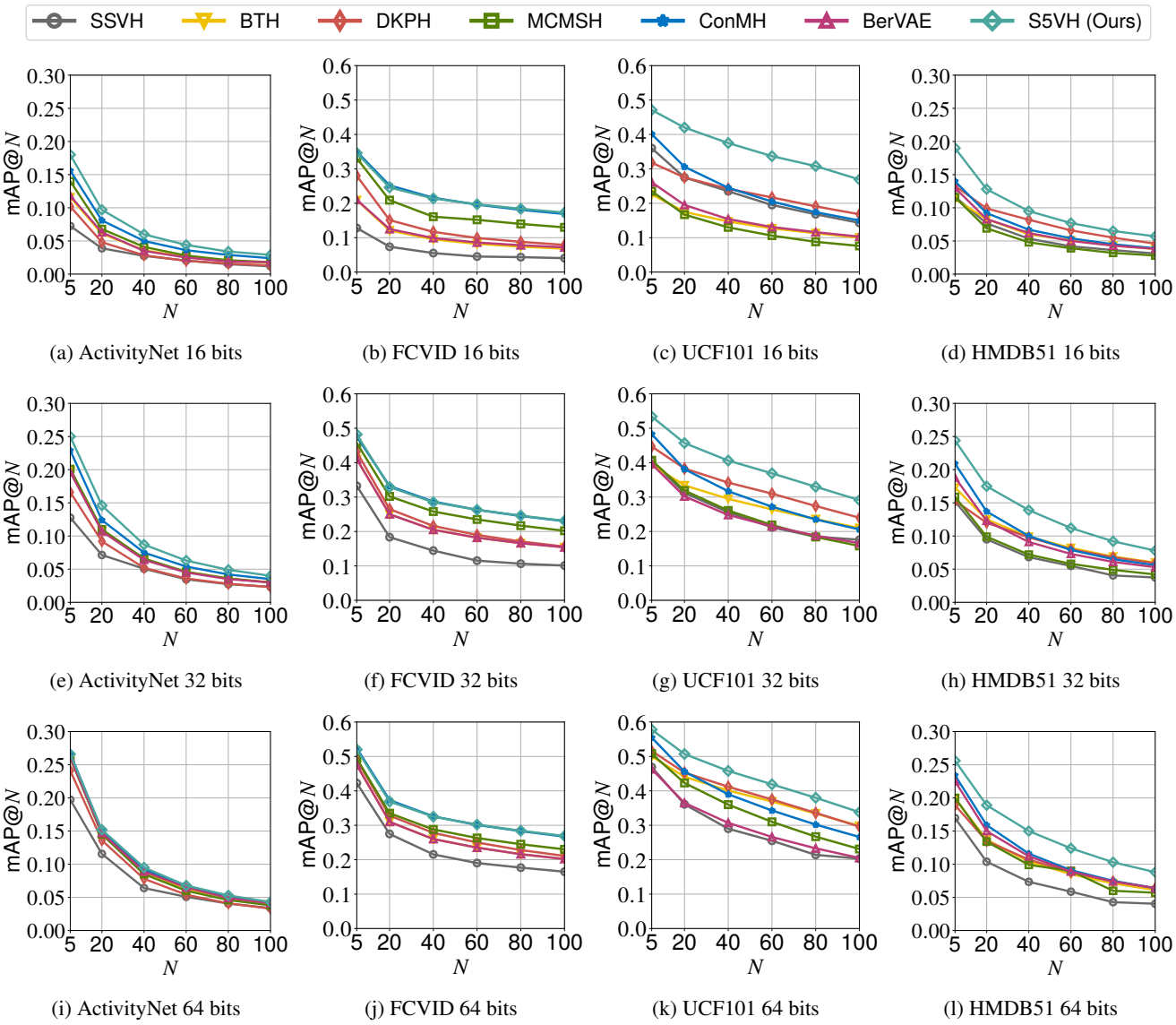
Table 2: Ablation study of S5VH. We use the GmAP metric.

Model Analyses

Effectiveness of Bidirectional Design We conduct an ablation study on the bidirectional design of S5VH and compare three block design strategies: (**i**) Forward Only, where the Mamba block processes the video sequence forward; (**ii**) Backward Only, where it processes the sequence backward; and (**iii**) Bidirectional (default), where stacked blocks process the sequence in both directions. As shown in Table 2, the bidirectional design improves retrieval performance by 1% to 6% compared to both Variant (1) and Variant (2). This advantage comes from its ability to capture temporal dependencies in both directions. By leveraging information from both past and future frames, the Bidirectional design creates a more comprehensive representation of the video sequence.

Effects of Different Loss Terms To analyse the effectiveness of three loss terms (*i.e.*, $\mathcal{L}_{\mathsf{TR}}$, $\mathcal{L}_{\mathsf{CL}}$ and $\mathcal{L}_{\mathsf{CA}}$) of S5VH, we construct several S5VH variants: (**i**) $\mathcal{L}_{\mathsf{TR}}$ Only: train the model with merely temporal reconstruction. (**ii**) w/o $\mathcal{L}_{\mathsf{CA}}$: S5VH removes the hash center alignment task. (**iii**) w/o $\mathcal{L}_{\mathsf{CL}}$: We train the model without contrastive learning. As shown in Table 2, the worst performance occurs when only $\mathcal{L}_{\mathsf{TR}}$ is used. Comparing Variant (4) with Variant (3), adding $\mathcal{L}_{\mathsf{CL}}$ significantly increases the GmAP, which can validate its necessity. Similarly, comparing Variant (5) with Variant (3) and Figure 1(c), incorporating $\mathcal{L}_{\mathsf{CA}}$ not only boosts GmAP accuracy by about 6% but also accelerates convergence by approximately 65%, underscoring the importance of $\mathcal{L}_{\mathsf{CA}}$.

Experiment with SSM Variants In addition to Mamba, other notable SSM architectures like RetNet (Sun et al. 2023a) and RWKV (Peng et al. 2023) have shown excellent performance in various tasks. We design S5VH Variants (6)-(8), equipped with LSTM (Hochreiter and Schmidhuber



(a) ActivityNet 16 bits (b) FCVID 16 bits (c) UCF101 16 bits (d) HMDB51 16 bits

(e) ActivityNet 32 bits (f) FCVID 32 bits (g) UCF101 32 bits (h) HMDB51 32 bits

(i) ActivityNet 64 bits (j) FCVID 64 bits (k) UCF101 64 bits (l) HMDB51 64 bits

Figure 3: Retrieval performance comparison by mAP@N.

1997) and the two architectures, to provide more insights for practical choices. To ensure a fair comparison, we only replace the Mamba layer with the corresponding bidirectional layers, while keeping all other factors the same.

According to Table 2, we validate that the default choice of Mamba is satisfactory to show the best performance. We notice that LSTM is still a strong option in SSVH, even though it has become less popular in recent years.

Conclusions

In this paper, we introduced S5VH, the first Mamba-based SSVH model with an enhanced learning paradigm. S5VH develop bidirectional Mamba layers to capture comprehensive temporal relations for hash learning. To improve training efficiency, we proposed a semantic hash center generation algorithm and a center alignment loss to extract and leverage the global learning signal. Experiments show S5VH's consistent improvements under various setups, transfers better, and superior inference efficiency. Our study suggests the strong potential of state-space models in video hashing, which we hope can inspire further research. *We have adjusted the size of the figures and tables as per the editorial requirements, which has led to the removal of some content. For the complete version, please refer to the arXiv version of the article with the same title.*

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China under grant 624B2088, 62171248, 62301189, and Shenzhen Science and Technology Program under Grant JCYJ20220818101012025, RCBS20221008093124061, GXWD20220811172936001.

References

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 961–970.

Duan, J.; Hao, Y.; Zhu, B.; Cheng, L.; Zhou, P.; and Wang, X. 2024. Efficient Unsupervised Video Hashing with Contextual Modeling and Structural Controlling. *IEEE Transactions on Multimedia*.

Fu, D. Y.; Dao, T.; Saab, K. K.; Thomas, A. W.; Rudra, A.; and Ré, C. 2022. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*.

Gao, K.; Bai, J.; Chen, B.; Wu, D.; and Xia, S.-T. 2023. Backdoor Attack on Hash-based Image Retrieval via Clean-label Data Poisoning. In *The 34th British Machine Vision Conference*, 172–173.

Gong, Y.; Lazebnik, S.; Gordo, A.; and Perronnin, F. 2012. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 35(12): 2916–2929.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Gu, A.; Dao, T.; Ermon, S.; Rudra, A.; and Ré, C. 2020. Hippo: Recurrent memory with optimal polynomial projections. *Advances in Neural Information Processing Systems*, 33.

Gu, A.; Goel, K.; Gupta, A.; and Ré, C. 2022. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35: 35971–35983.

Gu, A.; Goel, K.; and Ré, C. 2021. Efficiently Modeling Long Sequences with Structured State Spaces. In *International Conference on Learning Representations*.

Gu, A.; Johnson, I.; Goel, K.; Saab, K.; Dao, T.; Rudra, A.; and Ré, C. 2021. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34: 572–585.

Gupta, A.; Gu, A.; and Berant, J. 2022. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35: 22982–22994.

Hao, Y.; Duan, J.; Zhang, H.; Zhu, B.; Zhou, P.; and He, X. 2022. Unsupervised video hashing with multi-granularity contextualization and multi-structure preservation. In *Proceedings of the 30th ACM International Conference on Multimedia*, 3754–3763.

Hao, Y.; Mu, T.; Goulermas, J. Y.; Jiang, J.; Hong, R.; and Wang, M. 2017. Unsupervised t-distributed video hashing and its deep hashing extension. *IEEE Transactions on Image Processing*, 26(11): 5531–5544.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Jiang, Y.-G.; Wu, Z.; Wang, J.; Xue, X.; and Chang, S.-F. 2017. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(2): 352–364.

Kalman, R. E. 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1): 35–45.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T. A.; and Serre, T. 2011. HMDB: A large video database for human motion recognition. In Metaxas, D. N.; Quan, L.; Sanfeliu, A.; and Gool, L. V., eds., *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, 2556–2563. IEEE Computer Society.

Li, C.; Yang, Y.; Cao, J.; and Huang, Z. 2017. Jointly modeling static visual appearance and temporal pattern for unsupervised video hashing. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 9–17.

Li, P.; Xie, H.; Ge, J.; Zhang, L.; Min, S.; and Zhang, Y. 2022. Dual-stream knowledge-preserving hashing for unsupervised video retrieval. In *European Conference on Computer Vision*, 181–197. Springer.

Li, Q.; Tian, X.; and Ng, W. W. 2024. Self-supervised Temporal Sensitive Hashing for Video Retrieval. *IEEE Transactions on Multimedia*.

Li, S.; Chen, Z.; Li, X.; Lu, J.; and Zhou, J. 2019a. Unsupervised variational video hashing with 1D-CNN-LSTM networks. *IEEE Transactions on Multimedia*, 22(6): 1542–1554.

Li, S.; Chen, Z.; Lu, J.; Li, X.; and Zhou, J. 2019b. Neighborhood preserving hashing for scalable video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8212–8221.

Li, S.; Li, X.; Lu, J.; and Zhou, J. 2021. Self-supervised video hashing via bidirectional transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13549–13558.

Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; and Liu, Y. 2024. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*.

Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Biderman, S.; Cao, H.; Cheng, X.; Chung, M.; Derczynski, L.; et al. 2023. RWKV: Reinventing RNNs for the

Transformer Era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 14048–14077.

Qiao, Y.; Yu, Z.; Guo, L.; Chen, S.; Zhao, Z.; Sun, M.; Wu, Q.; and Liu, J. 2024. Vl-mamba: Exploring state space models for multimodal learning. *arXiv preprint arXiv:2403.13600*.

Ramachandran, P.; Zoph, B.; and Le, Q. V. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Shen, X.; Zhou, Y.; Yuan, Y.-H.; Yang, X.; Lan, L.; and Zheng, Y. 2023. Contrastive Transformer Hashing for Compact Video Representation. *IEEE Transactions on Image Processing*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, J. T.; Warrington, A.; and Linderman, S. W. 2022. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*.

Song, J.; Yang, Y.; Huang, Z.; Shen, H. T.; and Hong, R. 2011. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *Proceedings of the 19th ACM international conference on Multimedia*, 423–432.

Song, J.; Zhang, H.; Li, X.; Gao, L.; Wang, M.; and Hong, R. 2018a. Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE Transactions on Image Processing*, 27(7): 3210–3221.

Song, J.; Zhang, H.; Li, X.; Gao, L.; Wang, M.; and Hong, R. 2018b. Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE Transactions on Image Processing*, 27(7): 3210–3221.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR*, abs/1212.0402.

Sun, Y.; Dong, L.; Huang, S.; Ma, S.; Xia, Y.; Xue, J.; Wang, J.; and Wei, F. 2023a. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*.

Sun, Y.; Qin, Y.; Peng, D.; Ren, Z.; Yang, C.; and Hu, P. 2024. Dual Self-Paced Hashing for Image Retrieval. *IEEE Transactions on Multimedia*, 26: 9619–9629.

Sun, Y.; Ren, Z.; Hu, P.; Peng, D.; and Wang, X. 2023b. Hierarchical consensus hashing for cross-modal retrieval. *IEEE Transactions on Multimedia*, 26: 824–836.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34: 24261–24272.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, C.; Tsepa, O.; Ma, J.; and Wang, B. 2024a. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. *arXiv preprint arXiv:2402.00789*.

Wang, J.; Zeng, Z.; Chen, B.; Wang, Y.; Liao, D.; Li, G.; Wang, Y.; and Xia, S.-T. 2024b. Hugs Bring Double Benefits: Unsupervised Cross-Modal Hashing with Multi-granularity Aligned Transformers. *International Journal of Computer Vision*, 1–33.

Wang, Y.; Wang, J.; Chen, B.; Zeng, Z.; and Xia, S.-T. 2023a. Contrastive masked autoencoders for self-supervised video hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37(3), 2733–2741.

Wang, Y.; Zhou, M.; Sun, Y.; and Qian, X. 2023b. Uncertainty-aware unsupervised video hashing. In *International Conference on Artificial Intelligence and Statistics*, 6722–6740.

Wei, R.; Liu, Y.; Song, J.; Cui, H.; Xie, Y.; and Zhou, K. 2023. CHAIN: Exploring Global-Local Spatio-Temporal Information for Improved Self-Supervised Video Hashing. In *Proceedings of the 31st ACM International Conference on Multimedia*, 1677–1688.

Weiss, Y.; Torralba, A.; and Fergus, R. 2008. Spectral hashing. *Advances in neural information processing systems*, 21.

Wu, B.; and Ghanem, B. 2018. ℓ_p-Box ADMM: A Versatile Framework for Integer Programming. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1695–1708.

Wu, G.; Liu, L.; Guo, Y.; Ding, G.; Han, J.; Shen, J.; and Shao, L. 2017. Unsupervised deep video hashing with balanced rotation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 3076–3082.

Ye, G.; Liu, D.; Wang, J.; and Chang, S.-F. 2013. Large-scale video hashing via structure learning. In *Proceedings of the IEEE international conference on computer vision*, 2272–2279.

Zeng, Z.; Wang, J.; Chen, B.; Wang, Y.; and Xia, S.-T. 2022. Motion-Aware Graph Reasoning Hashing for Self-supervised Video Retrieval. In *33rd British Machine Vision Conference*, 82.

Zhang, H.; Wang, M.; Hong, R.; and Chua, T.-S. 2016. Play and rewind: Optimizing binary representations of videos by self-supervised temporal hashing. In *Proceedings of the 24th ACM international conference on Multimedia*, 781–790.

Zhao, H.; Zhang, M.; Zhao, W.; Ding, P.; Huang, S.; and Wang, D. 2024. Cobra: Extending mamba to multi-modal large language model for efficient inference. *arXiv preprint arXiv:2403.14520*.

Zhou, Y.; Sun, Z.; Liu, R.; Chen, Y.; and Zhang, D. 2024. AVHash: Joint Audio-Visual Hashing for Video Retrieval. In *Proceedings of the 32nd ACM international conference on Multimedia*.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.