This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2026.3720512

Information-Bottleneck Guided Hybrid Neural Architecture Search for Temporal Action Detection in Untrimmed Videos

Yi Tang, Mansen Chen, Lepeng Chen, Sen Wang, Min Liu, Yaonan Wang and Jun Liu

***Abstract*—Temporal Action Detection (TAD) in untrimmed videos requires effective spatial feature extraction for precise action classification and temporal feature modeling for accurate boundary localization. To achieve effective spatio-temporal feature integration, several works manually design rule-based (i.e., sequential or parallel) hybrid Mamba-Transformer networks for TAD. However, few studies explore diverse integration strategies and network topologies due to the inherent limitations of manual design. Therefore, we propose NAS-TAD, the first Neural Architecture Search framework for TAD, systematically exploring this untouched problem. Specifically, we develop a spatio-temporal NAS objective function based on information-bottleneck theory to quantify task-relevant spatio-temporal features, providing interpretable guidance for the network search and optimization process. Furthermore, we reformulate Transformer self-attention as a state-space model, thereby enabling seamless switching between Mamba and Transformer blocks in a unified weight-sharing search space. Consequently, comprehensive experiments on ActivityNet, THUMOS14, HACS and FineAction demonstrate the effectiveness of the searched hybrid architectures, providing new insights into temporal and spatial feature fusion for TAD. Code is available for reproduction at *https://anonymous.4open.science/r/nastad*.**

***Index Terms*—Temporal action detection, Neural architecture search, Information bottleneck theory, Video understanding.**

I. Introduction

TEMPORAL Action Detection (TAD) aims to identify and precisely localize action instances in untrimmed videos [1]–[6], underpinning applications such as video understanding [7], [8], highlight detection [9], and content retrieval [10], [11]. Unlike action recognition, TAD demands both accurate classification of action categories and precise localization of their temporal boundaries in untrimmed sequences. This dual challenge requires integrating robust spatial representations for action classification with effective temporal modeling for boundary detection [12], [13].

To tackle this spatio-temporal modeling challenge, recent advances in Mamba and Vision Transformers (ViTs) have shown significant promise in TAD [2], [14]–[17]. Specifically, ViTs leverage global self-attention to capture intricate spatial relationships, enhancing action classification accuracy [2], [14]. By contrast, Mamba's state-space model (SSM) efficiently processes long-range temporal dependencies with linear complexity, enabling scalable and precise action boundary localization in lengthy TAD video sequences compared to Transformers' quadratic complexity [15]. Thus, an optimal TAD framework should synergize ViTs' spatial modeling capabilities with Mamba's efficient temporal processing.

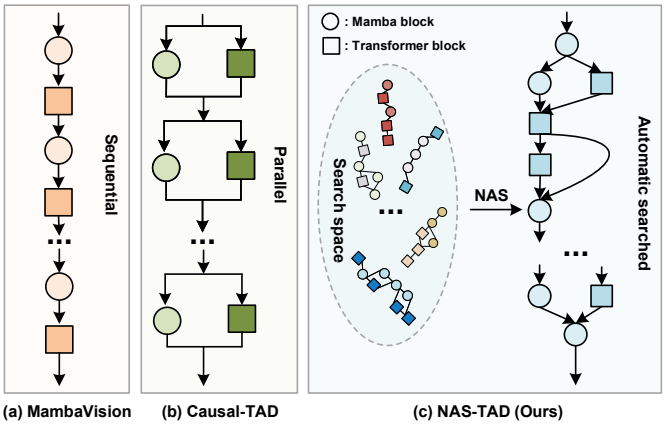

Fig. 1. Comparison between manually designed hybrid Mamba-Transformer architectures and our NAS-TAD framework. (a) MambaVision [19] employs a sequential arrangement of Mamba and Transformer blocks for efficient spatial and temporal feature extraction. (b) Causal-TAD [18] utilizes a parallel stacking of these blocks for complementary feature fusion. (c) Our NAS-TAD explores diverse architectural topologies, automatically discovering optimal hybrid networks through neural architecture search.

To this end, hybrid Mamba-Transformer architectures have been developed to combine these complementary strengths for TAD and computer vision tasks. For example, Causal-TAD integrates Transformer-based spatial modeling with temporal causality mechanisms to enhance both action classification and boundary localization in TAD [18]. Similarly, MambaVision [19] and MAP [20] combine Mamba's efficient sequence modeling with Transformer's self-attention in sequence, improving long-range spatial and temporal feature extraction for computer vision tasks.

Although recent advances have expanded strengths of these hybrid designs, most current architectures still rely on manually crafted sequential or parallel configurations, limiting exploration of broader integration strategies. As illustrated in Fig. 1(a) and (b), MambaVision [19] adopts a straightforward sequential alternation of Mamba and Transformer blocks to leverage their complementary strengths, whereas Causal-TAD employs a parallel stacking approach of Mamba and Transformer blocks to construct the hybrid architecture. However, fixed hand-crafted hybrid architectures impose a predefined interaction pattern between temporal operators, which can restrict temporal information propagation and integration.

In sequential designs, the later operator operates only on the representation produced by the earlier one, making temporal

Yi Tang, Mansen Chen, Lepeng Chen, Sen Wang, Jun Liu are with the Department of Data and Systems Engineering, The University of Hong Kong, Pokfulam, Hong Kong, China. (*Corresponding author: Jun Liu*). (e-mail: djliu@hku.hk)

Min Liu and Yaonan Wang are with the School of Artificial Intelligence and Robotics, Hunan University, Changsha, 410082, Hunan Province, China.

cues suppressed at early stages difficult to recover. In parallel designs, different operators typically process the input independently and interact only through late fusion, which limits intermediate information exchange and the modeling of dependencies between local boundary evidence and global semantic context. Therefore, the rationale behind selecting a specific sequential or parallel arrangement remains an open question. Few studies have explored the fundamental problem: **what is the optimal strategy to integrate Transformer and Mamba architectures to effectively fuse spatial and temporal features for TAD?**

Neural Architecture Search (NAS) offers a promising pathway to answer this question by exploring multifarious network topologies [21]–[23]. Nevertheless, few NAS studies have explored Mamba-Transformer hybrid architectures specifically for TAD tasks, primarily due to two critical challenges. First, existing NAS evaluation metrics predominantly prioritize classification accuracy [24], overlooking the qualitative assessment of effective spatial and temporal feature extraction critical for TAD. Second, mainstream weight-sharing NAS methodologies require candidate blocks to share a common tensor format to facilitate parameter exchange, which is incompatible with the fundamental structural differences between SSMs and Transformers: SSMs rely on sequential state transitions with convolution-like operations, whereas Transformers process spatial tokens through multi-head self-attention and feed-forward layers. Consequently, to our knowledge, a tailored NAS framework for TAD has not been proposed in the previous literature.

In this work, we propose NAS-TAD, the first neural architecture search framework specifically for temporal action detection as illustrated in Fig. 1(c), which provides new insights to effectively explore diverse network topologies for spatio-temporal feature fusion. First, we introduce a novel spatio-temporal NAS objective function based on information-bottleneck theory, which quantitatively evaluates a network's ability to extract task-relevant spatio-temporal features by maximizing mutual information with action labels while minimizing irrelevant redundancy. Unlike the conventional black-box, accuracy-based objective functions, our criterion provides interpretable theoretical guidance during the search process, balancing representational power and feature compactness. Second, we develop a harmonization scheme that reformulates Transformer self-attention as a state-space model (akin to Mamba), thereby mapping both SSM and Transformer blocks into a unified weight-sharing search space. Finally, based on the proposed objective function and harmonization scheme, we propose a mutual information-guided evolutionary algorithm to optimize the network architecture search process, yielding hybrid network with efficient spatial and temporal feature fusion. Extensive experiments on four established benchmarks (ActivityNet [25], THUMOS14 [26], HACS [27] and FineAction [1]) demonstrate the effectiveness of the searched hybrid network architecture.

In summary, our contributions are as follows:

- To the best of our knowledge, it is the first neural architecture search framework for TAD that explores optimal strategies for integrating Transformer and Mamba hybrid architectures, establishing a new paradigm and perspective for the TAD task.
- We develop a spatio-temporal NAS objective function based on information-bottleneck theory and provide interpretable guidance for NAS, breaking the conventional accuracy-based black-box NAS framework. Guided by this function, the unified Mamba-Transformer hybrid search space and the evolutionary search algorithm are proposed to effectively explore diverse network topologies, achieving efficient automatic network design for TAD in untrimmed videos.
- Comprehensive experiments on ActivityNet, THUMOS14, HACS, and FineAction demonstrate that NAS-TAD achieves new state-of-the-art results, confirming the advantages of the searched hybrid architecture compared with the manually designed networks.

II. Related Works

A. Temporal Action Detection

Temporal action detection is the task of identifying action instances in untrimmed videos with precise temporal boundaries and categories [3], [25]–[27]. Early methods employed CNNs like TSI [28] and I3D [29] for spatio-temporal feature extraction [30], [31]. However, these methods were limited by their constrained receptive fields, rendering them inadequate for modeling long-range temporal dependencies. Consequently, Recurrent Neural Network (RNN) based approaches [32], [33] were developed to enhance sequential modeling capabilities, yet gradient instability impaired their efficacy. With ViTs emerging in video processing [34], [35], Transformer-based TAD methods [2], [36], [37] leverage self-attention mechanisms to capture global spatial and temporal relationships. One of the representative works, ActionFormer [38], utilizes a Transformer-based architecture with multi-scale temporal feature extraction to achieve precise action localization and classification. For model compression of TAD, Zeng *et al.* [39] introduce graph convolutional networks to model proposal-proposal relations for temporal action localization, providing a structured alternative to dense prediction and enhancing model efficiency. More recently, Chen *et al.* [40] propose a progressive block-drop strategy that compresses an already-trained TAD model by iteratively pruning redundant blocks, while OVG-HQ framework [41] is proposed to handle online video grounding with hybrid-modal queries under tight efficiency constraints. More recently, Mamba-based approaches [18], [42], [43], utilizing state-space models' linear-complexity sequence modeling, achieved efficient action recognition and boundary regression. Nevertheless, these approaches depend on manually crafted architectures based on human expertise, constraining the exploration of sophisticated and automated neural network designs, which this work seeks to address.

B. Neural Architecture Search

Neural Architecture Search aims to automate the design of neural network architectures for optimal performance. Early foundational works employed reinforcement learning

and evolutionary algorithms to automate architecture optimization [21], yielding effective designs but incurring substantial computational costs. Consequently, DARTS [22] formulates NAS as a continuous optimization problem, enabling gradient-based search and dramatically reducing computational overhead. Similarly, weight-sharing NAS [44] utilizes parameter sharing to enhance efficiency, further improving scalability. Nevertheless, these approaches primarily address image classification, making them inadequate for video processing tasks that necessitate robust spatial and temporal dependency modeling.

Recent efforts have extended NAS to video processing, optimizing architectures to address high-dimensional video data and complex temporal dynamics across diverse tasks [45], [46]. Specifically, EVSRNet [45] employs NAS to discover architectures tailored for video super-resolution, while Yang *et al.* [46] introduce an automated cross-modal architecture search for video moment retrieval. Furthermore, to reduce the heavy computational consumption in video processing, some works have focused on lightweight and efficient network design [23], [47]. However, due to the complexity of modeling precise temporal boundaries and action categories, NAS for TAD has not been explored in previous works.

C. Hybrid Mamba-Transformer Network

To integrate global spatial modeling with efficient temporal processing, hybrid Mamba-Transformer architectures have emerged as a promising research topic. Jamba [48] developed a novel hybrid Mamba-Transformer language model architecture, integrating Transformer and Mamba layers for enhanced language modeling efficiency. In the vision domain, MambaVision [19] and MAP [20] developed hybrid architectures for vision tasks, leveraging Mamba's temporal efficiency and Transformer's spatial modeling to capture long-range spatial dependencies. Dimba [49] extends hybrid Mamba-Transformer designs to text-to-image generation by alternating Transformer and Mamba layers with cross-attention conditioning. For video tasks, Causal-TAD [18] integrates Transformer-based spatial modeling with temporal causality mechanisms to enhance both action classification and boundary localization. Therefore, we propose a spatio-temporal-aware NAS method to systematically search network topologies, automatically discovering Mamba-Transformer hybrids optimized for TAD tasks.

III. Methodology

Our approach employs weight-sharing neural architecture search, sampling subnetworks from a super-network without retraining to enhance computational efficiency, as illustrated in Fig. 2. Specifically, we first establish the spatio-temporal aware objective function to provide principled guidance for architecture evaluation. Building upon this foundation, we then design a unified search space that harmonizes the distinct computational paradigms of Mamba and Transformer architectures. Finally, we develop an evolutionary optimization framework that leverages the proposed objective function to efficiently navigate the search space and identify optimal architectures. The following subsections detail each component.

A. Spatio-Temporal Aware Objective Function

1) Problem Revisiting: In the conventional NAS framework, given a search space $\mathcal{A}$ and datasets $\mathcal{D} = \{\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}\}$, the goal is to identify an optimal subnetwork $\alpha^* \in \mathcal{A}$ that achieves superior performance in temporal action detection. For a detection model $\mathcal{M}(\alpha, w)$ with parameters w, the conventional NAS optimization problem is formulated as:

$$\begin{aligned} \alpha^* &= \arg\max_{\alpha \in \mathcal{A}} \text{mAP}(\mathcal{M}(\alpha, w^*), \mathcal{D}_{\text{val}}), \\ \text{s.t.} \quad w^* &= \arg\min_{w} \mathcal{L}(\mathcal{M}(\alpha, w), \mathcal{D}_{\text{train}}), \end{aligned} \tag{1}$$

where $\mathcal{L}(\cdot)$ is the detection loss (e.g., cross-entropy combined with temporal localization loss), and $\text{mAP}(\cdot)$ denotes the mean Average Precision.

However, such black-box objective functions only indicate *which* network architectures achieve good performance without explaining *why* certain architectures are effective. This explanatory gap limits deeper insights into architectural design principles for temporal action detection.

2) Spatio-Temporal Aware NAS Objective Function: To address the interpretability limitations of conventional black-box evaluation functions, we turn to information bottleneck (IB) theory [50] as a principled framework for evaluating network architectures' feature extraction capabilities. Specifically, IB theory provides a theoretical foundation for optimizing feature representations by maximizing mutual information with target variables while simultaneously minimizing irrelevant information retention. This principle directly aligns with our objective of discovering architectures that extract discriminative spatio-temporal features essential for TAD tasks.

Consequently, it serves as an ideal theoretical framework to guide architecture search, enabling interpretable evaluation of each candidate network's ability to capture meaningful spatio-temporal dependencies. Based on this theoretical foundation, we propose the following objective function:

$$\alpha^* = \arg\max_{\alpha \in \mathcal{A}} \left[\mathcal{I}\left(Z\left(\alpha, w; X\right), Y\right) - \beta \mathcal{I}\left(X, Z\left(\alpha, w; X\right)\right) \right], \tag{2}$$

where $Z(\alpha, w; X)$ is the feature representation from architecture α given input X, and $\beta > 0$ balances prediction and compression. The terms $\mathcal{I}(Z(\alpha, w; X), Y)$ maximize task-relevant temporal and spatial information, while $\mathcal{I}(X, Z(\alpha, w; X))$ minimizes input redundancy, ensuring compact and TAD-specific representations.

However, to achieve effective architecture search, we must explicitly quantify the spatial and temporal feature extraction capabilities. Building upon the IB framework, we decompose the mutual information term $\mathcal{I}(Z(\alpha, w; X), Y)$ into temporal and spatial components to provide granular evaluation of each architecture's performance.

For temporal feature evaluation, we quantify the dependency between extracted features Z and temporal labels Y^t (i.e., action start and end timestamps). This temporal mutual information serves as a direct measure of the architecture's temporal boundary localization capability:

$$\mathcal{I}_{\text{temporal}} = \mathcal{I}\left(Z; Y^t\right) = \mathbb{E}_{p(Z,\ Y^t)}\left[\log \frac{p\left(Z,\ Y^t\right)}{p\left(Z\right) p\left(Y^t\right)}\right], \tag{3}$$

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2026.3720512

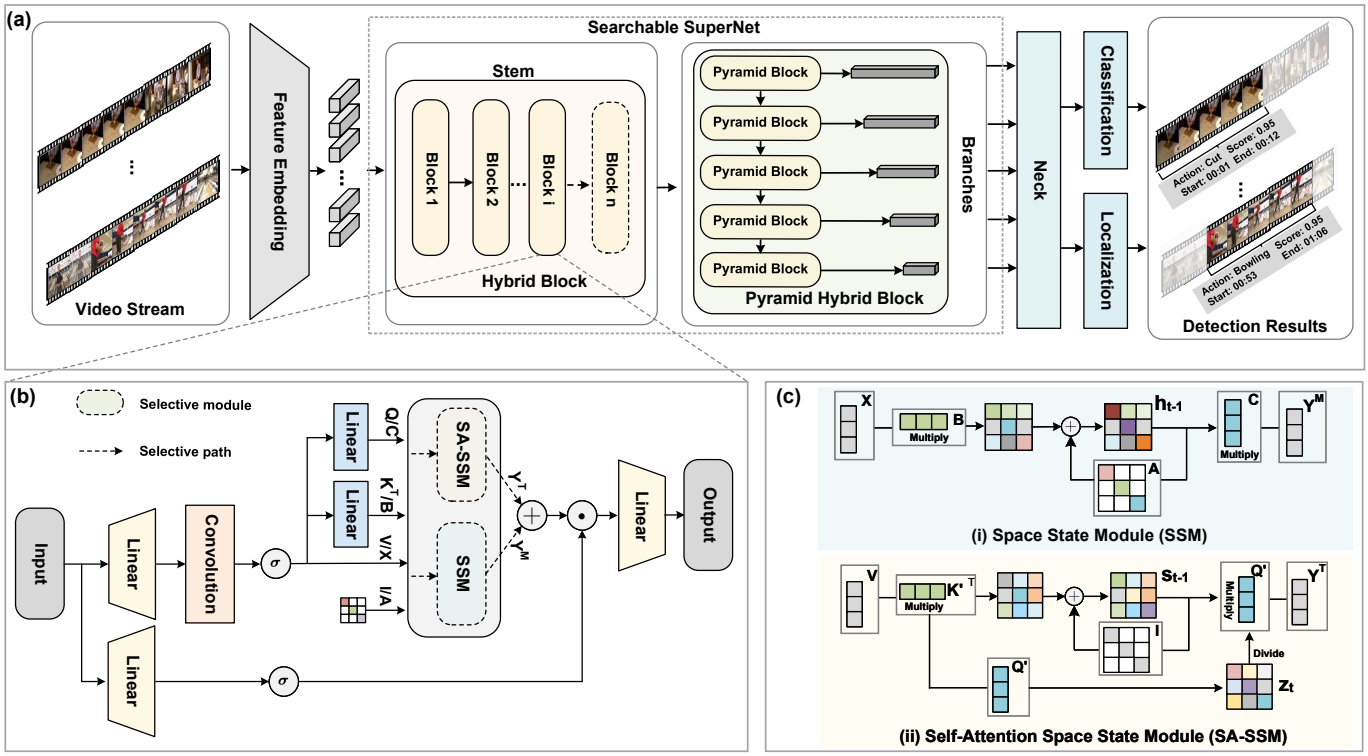

Fig. 2. (a) Searchable super-network architecture of the NAS-TAD framework. First, video streams are processed by a pretrained feature embedding module to generate feature patches. Next, these patches are fed into a stem of searchable Mamba-Transformer hybrid blocks for spatial and temporal feature fusion, with the number/layer of blocks searchable at the macro level (dashed lines). At the micro level, the specific module (i.e., SSM and SA-SSM) could also be searchable in each block. Subsequently, a searchable pyramid block, integrating Mamba-Transformer hybrid blocks akin to the stem and a convolutional downsampling mechanism, outputs multi-scale features for actions of varying temporal spans. Finally, these features are processed by a neck and classification/localization heads to produce TAD results. (b) Searchable Transformer-Mamba hybrid block, enabling seamless switching between Transformer, Mamba, or hybrid configurations via selective blocks and paths. (c) State-space modules: (i) Mamba's SSM for efficient sequence modeling, and (ii) Transformer's self-attention reformulated as an SSM for unified integration.

where $\mathcal{I}(\cdot)$ denotes the mutual information function, $p(Z, Y^t)$ represents the joint probability distribution of features Z and temporal labels Y^t, while $p(Z)$ and $p(Y^t)$ are their respective marginal distributions.

Analogously, for spatial feature evaluation, we measure the dependency between extracted features Z and spatial labels Y^s (i.e., action categories). This spatial mutual information quantifies the architecture's action classification performance:

$$\mathcal{I}_{\text{spatial}} = \mathcal{I}\left(Z; Y^s\right) = \mathbb{E}_{p(Z,\ Y^s)}\left[\log \frac{p\left(Z,\ Y^s\right)}{p\left(Z\right) p\left(Y^s\right)}\right], \quad (4)$$

where the notation follows the same convention as the temporal case, with Y^s representing spatial action category labels.

Consequently, our spatio-temporal aware objective function becomes:

$$\alpha^* = \arg\max_{\alpha \in \mathcal{A}} \left[\mathcal{I}_{\text{temporal}} + \mathcal{I}_{\text{spatial}} - \beta \mathcal{I}(X, Z(\alpha, w; X))\right], \quad (5)$$

where the first two terms maximize task-relevant spatial and temporal information extraction, while the third term ensures feature compactness by minimizing input redundancy.

B. Unified Mamba-Transformer Search Space

After establishing the spatio-temporal aware objective function as our guiding principle, the next critical challenge lies in constructing an effective search space that enables systematic exploration of network architectures. However, directly integrating Mamba and Transformer blocks presents a fundamental compatibility issue: these architectures exhibit inherently distinct computational paradigms that preclude straightforward parameter sharing in conventional NAS frameworks. To address this challenge, we propose a unified search space that harmonizes these disparate architectures through mathematical reformulation, enabling weight-sharing and efficient architecture exploration.

1) Mamba Review: Given input feature sequences $X = \{x_1, x_2, \ldots, x_n\}$, Mamba employs a structured state-space model (SSM) with a selective scanning mechanism to capture long-range temporal dependencies efficiently. For time step $t \in \{1, \ldots, n\}$, the core recurrence relation of the SSM is defined as:

$$\begin{cases} h_t = Ah_{t-1} + Bx_t \\ y_t = Ch_t \end{cases}. \quad (6)$$

Where h_t is the hidden state at time t, x_t is the input, y_t is the output, and A, B, C are learnable parameters governing state transitions, input and output mapping, respectively.

Nevertheless, to construct a weight-sharing hybrid block, we need to establish a consistent mathematical framework between the two architectures. This requires deriving an equivalent recursive formulation of the Transformer's attention

mechanism that mirrors Mamba's state transition dynamics, facilitating direct architectural comparison and search space design.

2) Transformer Reformulation: Given input feature sequences $X = \{x_1,\ x_2,\ \cdots,\ x_n\}$, the output for time step $t \in \{1, \ldots, n\}$ self-attention can be defined as:

$$Q_t = W_Q x_t,\ K_t = W_K x_t,\ V_t = W_V x_t, \tag{7}$$

$$y_t = \sum_{j=1}^{t} \frac{\exp\left(\frac{Q_t K_j^\top}{\sqrt{d}}\right)}{\sum_{j=1}^{t} \exp\left(\frac{Q_t K_j^\top}{\sqrt{d}}\right)} V_t, \tag{8}$$

where W_Q, W_K, W_V are the weight matrices of query, key and value, respectively.

Inspired by Linear-Attention [51], [52], we replace the exponential function with a non-negative kernel function $\phi(\cdot)$, yielding:

$$Q'_t = \phi\left(x_t W_Q\right),\ K'_t = \phi\left(x_t W_K\right),\ V_t = x_t W_V, \tag{9}$$

$$y_t = \sum_{j=1}^{t} \frac{Q'_t K_j^{'\top}}{\sum_{j=1}^{t} Q'_t K_j^{'\top}} V_t = \frac{Q'_t \sum_{j=1}^{t} K_j^{'\top} V_t}{\sum_{j=1}^{t} Q'_t K_j^{'\top}}. \tag{10}$$

For convenience, let $s_t = \sum_{j=1}^{t} K_j^{'\top} V_t$ and $z_t = \sum_{j=1}^{t} Q'_t K_j^{\top}$, then the recursive form of Transformer can be written as:

$$\begin{cases} s_t = 1 \cdot s_{t-1} + K_t^\top V_t \\ y_t = \frac{Q'_t}{z_t} s_t \end{cases}. \tag{11}$$

3) Unified Recursive Structure: Based on Eq. 11 and 6, Mamba's SSM and the reformulated Transformer self-attention exhibit analogous recursive structures, with parameter correspondences:

$$X \sim V, \quad A \sim I, \quad B \sim K'^\top, \quad C \sim \frac{Q'^\top}{z}, \tag{12}$$

where I denotes the identity matrix and $\sim$ denotes a mapping between Transformer and Mamba parameters. This analogy enables the Self-Attention State-Space Model (SA-SSM), as illustrated in Fig. 2(c), which reinterprets self-attention as a linear recurrence process compatible with Mamba's SSM.

For multi-head self-attention, the query, key, and value matrices are partitioned into h heads. Each head computes SA-SSM independently:

$$Q_t'^{(i)} = \phi(x_t W_Q^{(i)}), K_t'^{(i)} = \phi(x_t W_K^{(i)}), V_t^{(i)} = x_t W_V^{(i)}, \tag{13}$$

$$s_t^{(i)} = I s_{t-1}^{(i)} + (K_t'^{(i)})^\top V_t^{(i)}, y_t^{(i)} = \frac{Q_t'^{(i)} {s_t^{(i)}}^\top}{z_t^{(i)}}, \tag{14}$$

$$z_t^{(i)} = z_{t-1}^{(i)} + Q_t'^{(i)} (K_t'^{(i)})^\top, \tag{15}$$

where $y_t^{(i)}$ is the output of the i-th head and $W_Q^{(i)}, W_K^{(i)}, W_V^{(i)} \in \mathbb{R}^{d \times (d_k/h)}$ are the weight matrices for the query, key, and value of the i-th head, respectively. The head outputs are concatenated and projected via a linear layer $W_O \in \mathbb{R}^{d_k \times d}$:

$$y_t = \text{Concat}(y_t^{(1)}, y_t^{(2)} \ldots, y_t^{(h)}) W_O. \tag{16}$$

This output projection is jointly optimized with Mamba's output mapping matrix $C \in \mathbb{R}^{d \times d}$ to align feature representations, enabling seamless switching within the searchable hybrid Mamba-Transformer block.

Thus, we have achieved a unified reformulation of Transformer's self-attention and Mamba's SSM, constructing a searchable hybrid Mamba-Transformer block that facilitates weight-sharing and efficient architecture search.

C. Optimization

Based on the spatio-temporal aware objective function and the unified Mamba-Transformer search space, the optimization process aims to identify the most effective architecture for TAD tasks. Thus, we propose an optimization framework comprising two key components: mutual information estimation for quantifying architecture performance and an evolutionary algorithm for systematic architecture exploration. These components work synergistically to navigate the complex search space and identify architectures that maximize spatial and temporal feature extraction capabilities.

1) MI Estimation: Building upon the established objective function in Eq. 5, MI serves as a fundamental metric for assessing a network's feature extraction capabilities. However, computing MI directly poses significant challenges since the true joint and marginal distributions in Eq. 3 and 4 remain unknown. To address this intractability, we adopt a variational approach inspired by DVIB [53] and CLUB [54], introducing a learnable distribution $q_\phi(Y^t \mid Z)$ to approximate the unknown conditional distribution $p(Y^t \mid Z)$. Following this variational framework, the temporal MI estimation can be formulated as:

$$\begin{aligned} \mathcal{I}_{\text{temporal}} = \mathcal{I}(Z; Y^t) \approx \mathbb{E}_{p(Z,Y^t)} \left[\log q_\phi(Y^t \mid Z)\right] \\ - \mathbb{E}_{p(Z)p(Y^t)} \left[\log q_\phi(Y^t \mid Z)\right]. \end{aligned} \tag{17}$$

Hence, the problem is transformed into how to make q_ϕ as close as possible to the distribution p, which can be formulated using the KL divergence:

$$\begin{aligned} &\min_\phi \text{KL}(p(Y^t \mid Z) \parallel q_\phi(Y^t \mid Z)) \\ &= \min_\phi \left[\mathbb{E}_{p(Z,Y^t)} \left[\log p(Y^t \mid Z) - \log q_\phi(Y^t \mid Z)\right]\right]. \end{aligned} \tag{18}$$

Although p is unknown, the first term is independent of the parameters ϕ, so its expectation is a constant. Therefore, minimizing Eq. 18 is equivalent to maximizing $\mathbb{E}_{p(Z,Y^t)} \left[\log q_\phi(Y^t \mid Z)\right]$, which can be optimized using the maximum likelihood loss. Given N samples $\{(z_i, g_i^t)\}_{i=1}^N$ from the joint distribution $p(Z, Y^t)$, the optimal parameters are:

$$\phi^* = \arg\max_\phi \frac{1}{N} \sum_{i=1}^{N} \log q_\phi(g_i^t \mid z_i). \tag{19}$$

This is maximum likelihood estimation for the variational distribution, optimized jointly with the SuperNetwork training.

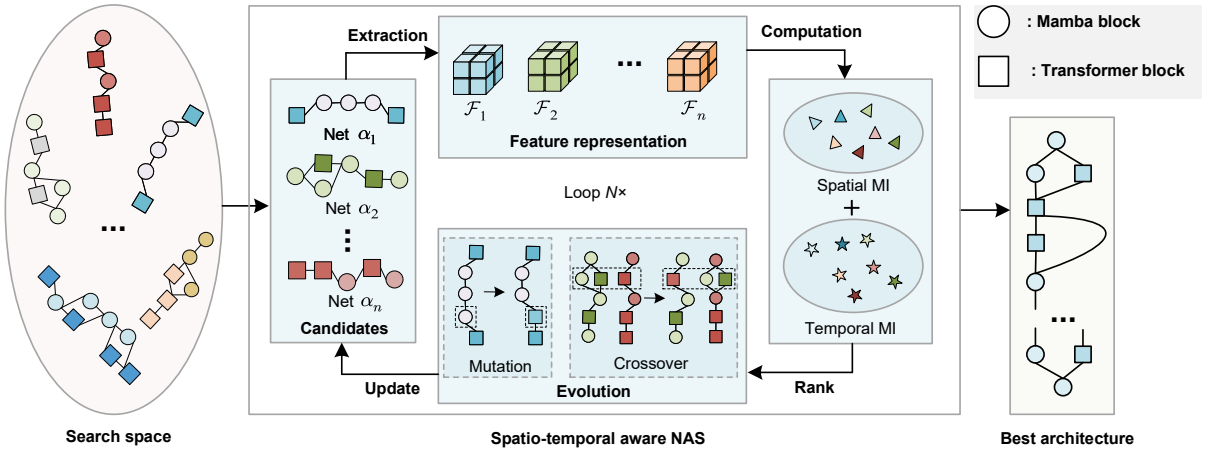

Fig. 3. Optimization framework of the proposed NAS-TAD. First, candidate subnetwork architectures are sampled from the unified Mamba-Transformer search space. Second, each subnetwork extracts distinct spatio-temporal features. Then, these subnetworks are evaluated and ranked by the spatio-temporal aware objective function. Next, the top-K networks undergo evolutionary updates via mutation and crossover. Subsequently, these evolutionary networks, along with newly sampled candidates, update the candidate pool for the next loop. Finally, the architecture with the maximum objective value is selected as the optimal architecture after iterative refinement.

Algorithm 1 Spatio-Temporal Aware NAS Optimization Algorithm

Input: TAD validation set $\mathcal{D}_{\text{val}}$; search space $\mathcal{A}$
Parameter: Population size $|\mathcal{P}|$; evolution generations E_m; computation budget η; mutation probability p_m; crossover probability p_c;
Output: Optimal network α_{best} ;

Initialize: $\alpha_{\text{best}} \leftarrow \emptyset, \mathcal{I}_{\max} \leftarrow 0, e \leftarrow 1$
$\mathcal{P} \leftarrow \text{sample}(\mathcal{A}, |\mathcal{P}|; \eta)$
while $e \leq E_m$ **do**
  for α_j in $\mathcal{P}$ **do**
    $\mathcal{I}_j \leftarrow \mathcal{I}(\alpha_j, \mathcal{D}_{\text{val}})$
    if $\mathcal{I}_j \geq \mathcal{I}_{\max}$ **then**
      $\alpha_{\text{best}}, \mathcal{I}_{\max} \leftarrow \alpha_j, \mathcal{I}_j$
    end if
  end for
  $\mathcal{P}_{\text{parent}} \leftarrow \text{top\_k}(\mathcal{P}, K)$
  $\mathcal{P}_{\text{child}} \leftarrow \text{crossover\_mutation}(\mathcal{P}_{\text{parent}}, p_c, p_m)$
  $\mathcal{P}' \leftarrow \mathcal{P} \cup \mathcal{P}_{\text{child}}$
  $\mathcal{P} \leftarrow \text{top\_k}(\mathcal{P}', |\mathcal{P}|)$
  $e \leftarrow e + 1$
end while
return α_{best}

Similar to temporal MI, the spatial MI evaluates the correlation between network-extracted features Z and spatial labels Y^s. Thus, the spatial MI estimation can be expressed as:

$$\begin{aligned}\mathcal{I}_{\text{spatial}} &= \mathbb{E}_{p(Z,\ Y^s)}\left[\log \frac{p\left(Z,\ Y^s\right)}{p\left(Z\right)p\left(Y^s\right)}\right] \\ &\approx \mathbb{E}_{p(Z,Y^s)}\left[\log q_\varphi(Y^s \mid Z)\right] \\ &\quad - \mathbb{E}_{p(Z)p(Y^s)}\left[\log q_\varphi(Y^s \mid Z)\right],\end{aligned} \tag{20}$$

where $q_\varphi(Y^s \mid Z)$ is a variational distribution approximating the unknown distribution $p(Y^s \mid Z)$. It can also be optimized using maximum likelihood estimation. Y^s denotes spatial labels representing action categories in TAD.

2) Evolutionary optimization Framework: Since the objective function's mutual information terms are estimated via sampling, they are non-differentiable, precluding gradient-based optimization. Instead, we adopt evolutionary algorithms to optimize the architecture search. The overall optimization framework is illustrated in Fig. 3. First, a population of candidate subnetworks is randomly sampled from the unified Mamba-Transformer search space. Subsequently, each subnetwork processes input video features to extract distinct spatial and temporal representations. Following this feature extraction, these subnetworks are then evaluated using the proposed spatio-temporal aware objective function, which quantifies their effectiveness in capturing task-relevant spatio-temporal features for TAD tasks. Based on the evaluation results, the top-K performing subnetworks are selected as parents for the next generation. To explore architectural diversity, crossover and mutation operations are applied to these parent architectures, thereby introducing diversity and exploring novel configurations. Accordingly, the newly generated candidates, along with the existing population, form the candidate pool for the next iteration. This evolutionary process continues iteratively for a predefined number of generations, progressively refining the population towards optimal architectures. Finally, upon completion of the search process, the subnetwork with the highest spatio-temporal objective function value is selected as the optimal architecture for deployment. The detailed procedure is presented in Algorithm 1.

IV. Experiments

A. Datasets and Implementation Details

Comprehensive evaluation of the proposed NAS-TAD framework is conducted through extensive experiments on four established temporal action detection benchmarks: ActivityNet [25], HACS [27], THUMOS14 [26], and FineAction [1]. Detailed dataset characterization is first provided, followed by implementation details and training configurations for repro-

ducible comparisons. Finally, evaluation metrics for temporal action detection performance assessment are described.

1) ActivityNet: ActivityNet comprises 20,000 untrimmed videos spanning 200 activity classes with over 648 hours of real-world footage [25]. The dataset provides temporal action annotations for 10,000 training videos, 5,000 validation videos, and 5,000 test videos. Its diverse content, ranging from sports to daily activities, presents challenges in action localization and classification, making it ideal for evaluating TAD methods.

2) HACS: HACS contains 49,408 untrimmed videos sourced from YouTube, encompassing 200 action classes with over 1.55 million clips and 139,000 temporally annotated segments for recognition and localization tasks [27]. The dataset is partitioned into 39,138 training videos and 10,270 validation videos. Its extensive scale and diversity make it a challenging benchmark for temporal action detection, particularly in complex scenarios with overlapping actions and varying temporal dynamics.

3) THUMOS14: The THUMOS14 dataset features 1,010 untrimmed videos from 20 sports action classes, with temporal annotations provided for 220 validation videos and 213 test videos, yielding over 3,200 action instances [26]. This collection is renowned for its challenging temporal boundaries and precise localization requirements in dynamic sports scenarios.

4) FineAction: FineAction consists of 4,000 videos annotated with 106 fine-grained action classes, encompassing 103,000 temporal instances that capture subtle variations in human activities [1]. The dataset is divided into 3,000 training, 500 validation, and 500 test videos, emphasizing distinctions in action manners, tools, and environments.

5) Implementation Details: Our method is primarily implemented using the MMAction2 [55] and OpenTAD [56] libraries in Python with PyTorch 2.0. All experiments are conducted on a workstation equipped with four NVIDIA RTX 3090 GPUs. To ensure a fair and objective comparison with established baselines in the temporal action detection domain, we maintain the training epochs and hyperparameter selections consistent with the default configurations of MMAction2 and OpenTAD. Regarding the architecture of the learnable distributions, both q_ϕ and q_φ are instantiated as lightweight two-layer MLPs with a hidden dimension of 256, GELU activation, and dropout of 0.1. Specifically, $q_\phi\left(Y^t \mid Z\right)$ outputs a 2-dimensional Gaussian over the normalized boundary pair (t_s, t_e) by predicting its mean and log-variance, whereas $q_\varphi\left(Y^s \mid Z\right)$ outputs class logits over the C action categories. Both distributions share the super-net's penultimate feature Z and are jointly optimized with the super-net via the maximum-likelihood loss in Eq. 19. For training, we employ the AdamW [57] optimizer across all datasets, with a cosine annealing learning rate scheduler to facilitate stable convergence.

Following the setting of previous evolutionary NAS algorithms [58]–[60], our search process utilizes a population size of 100 (i.e., the number of network architectures sampled per iteration) across 50 generations. In each iteration, the top 10 architectures are selected as parents and undergo crossover and mutation operations to generate 50 child-networks. To balance between exploration and exploitation, the mutation probability is set to 0.2, while the crossover probability is 0.4. To balance complexity and efficiency, the network depth is constrained between 10 and 20 layers, with a maximum parameter limit of 50M. After the search phase, the discovered optimal network is retrained from scratch on each benchmark using the same training configuration as ActionFormer [38] to ensure a strictly fair comparison.

6) Evaluation Metrics: The primary evaluation metric is mean Average Precision (mAP), computed at dataset-specific Intersection over Union (IoU) thresholds. For ActivityNet and HACS datasets, mAP is reported at IoU thresholds of 0.5, 0.75, 0.95, and averaged over [0.5: 0.05: 0.95], while the THUMOS14 datasets are evaluated at IoU thresholds of [0.3: 0.1: 0.7]. Our baseline adopts ActionFormer [38] as the detection framework. Specifically, we replace ActionFormer's Transformer encoder layers (the feature projection and fusion network) with our searched hybrid Mamba-Transformer architecture, while keeping the classification and localization heads unchanged to ensure fair comparison.

B. Comparison with State-of-the-art Methods

To evaluate the effectiveness of our NAS-TAD framework, comprehensive comparisons are conducted against state-of-the-art temporal action detection methods across four benchmarks: ActivityNet, HACS, THUMOS14, and FineAction. The experimental results validate the generalization capability and robustness of our automatically discovered architectures across diverse video domains, ranging from long-form activities to fine-grained actions.

1) ActivityNet: We compare our NAS-TAD with state-of-the-art TAD approaches, including representative convolution-based methods such as TriDet [65], Transformer-based methods like ActionFormer [38] and AdaTAD [74], Mamba-based methods such as ActionMamba [42], and hybrid architectures like Causal-TAD [18] and MambaVision [19], as summarized in Table I. Our NAS-TAD achieves 43.1% in terms of average mAP, surpassing the previous state-of-the-art ActionMamba and DiGIT by 1.1% in mAP@avg. NAS-TAD also achieves the best results at IoU thresholds of 0.5 and 0.75, while remaining competitive at the strict IoU threshold of 0.95. These results highlight the framework's enhanced precision in action localization within long untrimmed videos and its ability to capture fine-grained temporal boundaries.

From the perspective of design paradigms, with SlowFast-R50 embeddings, our NAS-TAD attains 39.5% in average mAP, exceeding TriDet's 36.8% by 2.7% and DyFADet's 38.5% by 1.0% in terms of mAP@avg., as the automated search mitigates human biases in architecture selection. When employing InternVideo2-6B, our hybrid NAS approach yields improvements of 1.1% in average mAP compared with ActionMamba and DiGIT. These empirical gains demonstrate that NAS-TAD's searched architectures effectively integrate diverse network components for superior spatial and temporal feature fusion. In addition, performance improvements on varied feature embeddings substantiate the framework's adaptability and robustness across different video representations.

From the aspect of model complexity, NAS-TAD delivers a favorable balance between accuracy and efficiency

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2026.3720512

TABLE I
PERFORMANCE AND COMPLEXITY COMPARISON WITH SOTA METHODS ON ACTIVITYNET DATASET. THE BEST RESULTS ARE REPORTED IN **BOLD**, WHILE THE SECOND-BEST ARE UNDERLINED.

Methods	Feature Embeddings	Network Types	Params (M)	FLOPs (G)	Latency (ms)	mAP@IoU Threshold (%)			
						0.5	0.75	0.95	Avg.
E2E-TAD [61]	SlowFast-R50 [62]	Convolution	8.5	56.9	12.0	50.5	36.0	10.3	35.1
ActionFormer [38]	I3D [29]	Transformer	6.9	2.7	12.7	53.5	36.3	8.2	35.6
TALLFormer [36]	VideoSwin-B [35]	Transformer	-	-	-	54.1	36.2	7.9	35.6
ASL [63]	I3D [29]	Transformer	-	-	-	54.1	37.4	8.0	36.2
BRTAL [64]	I3D [29]	Transformer	-	-	-	54.8	37.9	8.3	36.7
TriDet [65]	I3D [29]	Convolution	14.0	11.0	27.0	54.7	38.0	8.4	36.8
Re^{2}TAL [66]	Re-VideoSwin-T [66]	Transformer	-	-	-	54.8	37.8	9.0	36.8
CLTDR-GMG [67]	R(2+1)D [68]	Transformer	-	-	-	55.0	38.0	8.6	37.1
DiGIT [69]	R(2+1)D [68]	Transformer	-	-	-	54.4	38.2	10.7	37.3
LSGNet [70]	R(2+1)D [68]	Hybrid	-	-	-	57.6	40.0	8.3	38.5
DyFADet [71]	VideoMAE-v2 [72]	Convolution	22.4	5.4	39.4	58.1	39.6	8.4	38.5
Causal-TAD [18]	InternVideo2-6B [73]	Hybrid	14.8	2.4	12.0	-	-	-	39.0
ActionFormer [38]	InternVideo2-6B [73]	Transformer	9.0	2.3	12.8	60.4	42.2	8.7	40.8
AdaTAD [74]	VideoMAEv2-g [72]	Transformer	-	-	-	61.7	43.4	10.9	41.9
MambaVision [19] (Our impl.)	InternVideo2-6B [73]	Hybrid	20.2	5.3	32.1	62.0	43.1	10.8	41.9
DiGIT [69]	InternVideo2-6B [73]	Transformer	-	-	-	62.0	43.1	**11.3**	42.0
ActionMamba [42]	InternVideo2-6B [73]	Mamba	**6.3**	**1.2**	**10.2**	62.4	43.5	10.2	42.0
NAS-TAD (ours)	I3D [29]	Hybrid	13.0	3.1	18.0	55.1	37.8	9.1	37.2
	SlowFast-R50 [62]	Hybrid	13.2	3.2	18.1	59.3	40.5	10.0	39.5
	InternVideo2-6B [73]	Hybrid	13.9	3.4	18.5	**63.4**	**44.6**	11.1	**43.1**

TABLE II
COMPARISON WITH SOTA METHODS ON HACS DATASET. †: VIDEO-MAE FEATURE EMBEDDINGS. ‡: INTERNVIDEO2-6B FEATURE EMBEDDINGS. OTHERS ARE SLOWFAST FEATURE EMBEDDINGS.

Methods	mAP@IoU Threshold (%)			
	0.5	0.75	0.95	Avg.
BMN [75]	52.5	36.4	10.4	35.8
ActionFormer [38]	54.9	36.9	9.5	36.4
TALLFormer [36]	55.0	36.1	11.8	36.5
TCANet [76]	54.1	37.2	11.3	36.8
TriDet [65]	56.7	39.3	11.7	38.6
DyFADet [71]	57.8	39.8	11.8	39.2
CLTDR-GMG [67]	57.6	39.9	12.0	39.3
TriDet [65] †	62.4	44.1	13.1	43.1
DyFADet [71] †	64.0	44.8	**14.1**	44.3
ActionMamba [42] ‡	64.0	45.7	13.3	44.6
NAS-TAD (ours)	58.0	40.2	11.9	39.5
NAS-TAD (ours) †	64.5	44.7	13.5	45.0
NAS-TAD (ours) ‡	**64.8**	**46.7**	**14.1**	**45.9**

among hybrid architectures. With InternVideo2-6B features, our searched model uses only 13.9M parameters and 3.4G FLOPs. Compared with the hand-crafted hybrid baseline MambaVision [19], NAS-TAD employs 31.2% fewer parameters (13.9M vs. 20.2M), 35.8% fewer FLOPs (3.4G vs. 5.3G), and 42.4% lower latency (18.5 ms vs. 32.1 ms), while still improving the average mAP by 1.2% (43.1% vs. 41.9%). Furthermore, compared with another hybrid baseline Causal-TAD [18], NAS-TAD reduces the parameter count by 6.1% (13.9M vs. 14.8M) and substantially raises the average mAP by 4.1% (43.1% vs. 39.0%), at the cost of only 6.5 ms of additional latency. Although NAS-TAD is heavier than the lightweight pure-Mamba design ActionMamba [42], it still attains a 1.1% higher average mAP and a notably larger gain of 0.9% at the strict IoU threshold of 0.95 (11.1% vs. 10.2%), which indicates substantially better fine-grained boundary localization. These results jointly confirm that the architecture searched by NAS-TAD is structurally more compact than the existing hybrid approaches and delivers state-of-the-art accuracy at a deployment-friendly inference latency.

2) HACS: As reported in Table II, our NAS-TAD with InternVideo2-6B embeddings achieves an average mAP of 45.9%, outperforming ActionMamba by 1.3% under the same feature embedding. With SlowFast features, our NAS-TAD framework improves the average mAP over ActionFormer by 3.1% (39.5% vs. 36.4%). For Video-MAE feature embeddings, NAS-TAD attains 45.0% in average mAP, surpassing DyFADet by 0.7% and TriDet by 1.9%, demonstrating the searched architecture's effectiveness across different feature representations. These results indicate that the NAS-optimized hybrid architecture effectively handles HACS's dense annotations and overlapping segments, showcasing its capability to discern complex temporal patterns and action boundaries in large-scale video datasets.

3) THUMOS14: As illustrated in Table III, our NAS-TAD with InternVideo2-6B embeddings attains an average mAP of 74.7%, exceeding ActionMamba by 2.0% under equivalent conditions. In comparison to I3D embeddings, NAS-TAD achieves 70.2% in average mAP, surpassing TriDet by 0.9% (70.2% vs. 69.3%). Notably, at the stringent IoU threshold of 0.7, our approach with InternVideo2-6B yields 52.4%, matching AdaTAD and outperforming ActionMamba by 1.6% (52.4% vs. 50.8%), indicating superior boundary precision. These outcomes reflect the NAS-optimized hybrid design's capability to address THUMOS14's demanding temporal localization in dynamic sports videos, as supported by the consistent advancements in average mAP and high IoU metrics across embeddings.

4) FineAction: As shown in Table IV, our NAS-TAD is evaluated under Video-MAE and InternVideo2-1B feature embeddings, achieving the best average mAP of 28.6% and 30.4%, respectively. Under Video-MAE features embeddings, it surpasses DyFADet [71] by 4.8%, while with InternVideo2-1B, it outperforms ActionMamba [42] by 1.4% in average mAP. At higher IoU thresholds, our method shows particularly

TABLE III
COMPARISON WITH SOTA METHODS ON THUMOS14 DATASET. †: VIDEO-MAE FEATURE EMBEDDINGS. *: SLOWFAST FEATURE EMBEDDINGS. ‡: INTERNVIDEO2-6B FEATURE EMBEDDINGS. OTHERS ARE I3D FEATURE EMBEDDINGS.

Methods	mAP@IoU Threshold (%)					
	0.3	0.4	0.5	0.6	0.7	Avg.
TadTR [2]	62.4	57.4	49.2	37.8	26.3	46.6
E2E-TAD [61] *	69.4	64.3	56.0	46.4	34.9	54.2
AdaTAD [74] *	81.0	76.2	69.4	59.0	44.5	66.0
ActionFormer [38]	82.1	77.8	71.0	59.4	43.9	66.8
TransGMC [77]	82.3	78.8	71.4	60.0	45.1	67.5
ASL [63]	83.1	79.0	71.7	59.7	45.8	67.9
TriDet [65]	83.6	80.1	72.9	62.4	47.4	69.3
TriDet [65] †	84.8	80.0	73.3	63.8	48.8	70.1
ActionFormer [38] *	82.3	81.9	75.1	65.8	50.3	71.9
ActionMamba [42] ‡	86.9	83.1	76.9	65.9	50.8	72.7
AdaTAD [74] †	87.7	84.1	76.7	66.4	**52.4**	73.5
NAS-TAD (ours)	84.6	80.8	73.1	63.5	48.8	70.2
NAS-TAD (ours) *	87.2	83.1	76.2	66.1	49.9	72.5
NAS-TAD (ours) ‡	**88.2**	**84.3**	**78.0**	**68.1**	**52.4**	**74.7**

TABLE IV
COMPARISON WITH SOTA METHODS ON FINEACTION DATASET. †: VIDEO-MAE FEATURE EMBEDDINGS. ‡: INTERNVIDEO2-1B FEATURE EMBEDDINGS. OTHERS ARE I3D FEATURE EMBEDDINGS.

Methods	mAP@IoU Threshold (%)			
	0.50	0.75	0.95	Avg.
G-TAD [78]	13.7	8.8	3.1	9.1
BMN [75]	14.4	8.9	3.1	9.3
ActionFormer [38] †	29.1	17.7	5.1	18.2
DyFADet [71] †	37.1	23.7	5.9	23.8
ActionFormer [38] ‡	43.1	27.1	5.3	27.2
ActionMamba [42] ‡	45.4	28.8	6.8	29.0
NAS-TAD (ours)	28.3	16.2	4.3	17.1
NAS-TAD (ours) †	45.0	28.7	6.1	28.6
NAS-TAD (ours) ‡	**45.8**	**29.9**	**7.5**	**30.4**

strong performance, achieving 7.5% at IoU=0.95 compared to ActionMamba's 6.8%, demonstrating superior temporal boundary precision. These improvements demonstrate the efficacy of our neural architecture search in optimizing for fine-grained action hierarchies and variable durations, showcasing the framework's ability to capture subtle temporal patterns essential for discriminating between closely related action categories.

C. Ablation Study and Analysis

1) Search Space: To assess the impact of search space components, ablation experiments are presented in Table V. The baseline (without any added block) achieves an average mAP of 40.8% with InternVideo2-6B feature embeddings, while consuming 9.0M parameters, 2.3G FLOPs, and 12.8 ms of latency. Adding only Transformer blocks raises the average mAP to 42.3% (+1.5%), benefiting from the global self-attention mechanism that captures spatial relationships and inter-frame dependencies for accurate action classification. However, this gain comes at the cost of more parameters (19.2M) and higher latency (20.6 ms). In contrast, the Mamba-only sub-space yields 42.4% (+1.6%) average mAP with the lowest complexity (8.7M parameters, 2.0G FLOPs, and 11.2 ms of latency), thanks to its efficient long-range temporal modeling with linear complexity that enables better temporal boundary localization.

TABLE V
ABLATION STUDY ON THE IMPACT OF DIFFERENT BLOCKS ON THE ACTIVITYNET DATASET.

Methods	Transformer	Mamba	Hybrid*	Params (M)	FLOPs (G)	Latency (ms)	mAP @Avg.(%)
Baseline	-	-	-	9.0	2.3	12.8	40.8
	✓	-	-	19.2	3.7	20.6	42.3
	-	✓	-	8.7	2.0	11.2	42.4
	-	-	✓	13.9	3.4	18.5	43.1

*Hybrid corresponds to the optimal architecture discovered by NAS-TAD over the hybrid Mamba and Transformer search space.

TABLE VI
ABLATION STUDY ON THE IMPACT OF THE TRADE-OFF HYPER-PARAMETER β ON THE ACTIVITYNET DATASET.

β	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
mAP (%)	42.7	42.3	42.8	42.9	**43.1**	42.7	42.5	41.7	41.2	41.0

Moreover, the hybrid architecture searched by NAS-TAD attains 43.1% (+2.3%) average mAP with 13.9M parameters, 3.4G FLOPs, and 18.5 ms of latency. Compared with the Transformer-only sub-space, the discovered hybrid uses 27.6% fewer parameters (13.9M vs. 19.2M), 8.1% fewer FLOPs (3.4G vs. 3.7G), and 10.2% lower latency (18.5 ms vs. 20.6 ms), while still improving average mAP by 0.8%. Compared with the more efficient Mamba-only sub-space, the discovered hybrid trades a moderate increase in complexity for a consistent 0.7% gain in average mAP. These observations indicate that the IB-guided search does not blindly pursue accuracy at the price of efficiency. Instead, it identifies an intermediate operating point that combines Transformer's spatial-modeling capability with Mamba's temporal efficiency, thereby achieving the best accuracy-efficiency trade-off within the unified search space.

2) Hyperparameter β: According to the IB theory, the hyperparameter β is used to balance the spatio-temporal MI and input MI in Eq. 2. To identify the optimal value, we performed comparative experiments across a range of β values. As illustrated in Table VI, our method attained the best performance when β is set to 0.5. If continuously increasing β to 1, there is an obvious performance drop. It is because the network will 'forget' the information from the input and tend to be overfitting with the target label when β increases [50]. Therefore, $\beta = 0.5$ is selected as the optimal value for best performance.

3) Spatio-Temporal Aware NAS Objective Function: To validate the effectiveness of our proposed spatio-temporal objective function, we compare it against the conventional mAP-based NAS objective function, with results shown in Fig. 4. From this cross-dataset evaluation, the mAP-based method proves effective only when the search and test datasets are the same, while it exhibits limited performance improvements when architectures are transferred to different datasets (e.g., ΔmAP is only 0.7% when searching on ActivityNet and testing on HACS), since conventional mAP-based objectives tend to overfit to specific dataset characteristics and fail to capture generalizable spatial and temporal patterns essential for diverse TAD scenarios.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2026.3720512

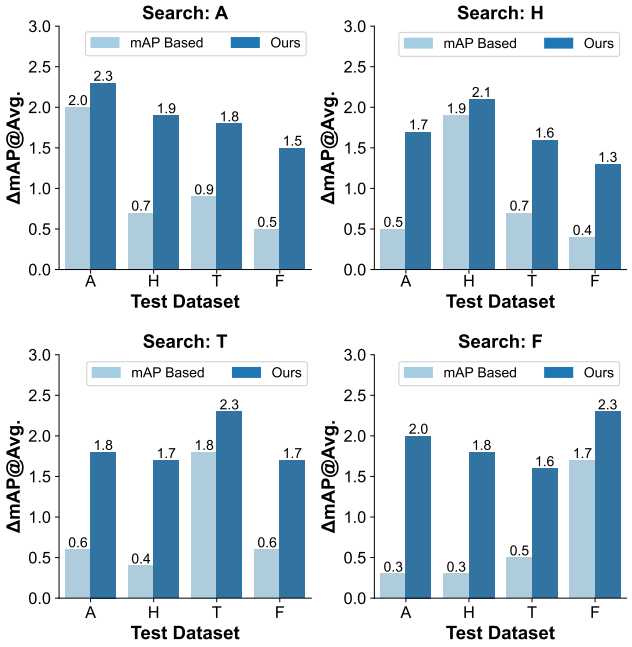

Fig. 4. Cross-dataset comparison of performance improvements in baseline (ΔmAP@Avg.) between the mAP-based NAS objective function and ours. Each panel corresponds to a search dataset (A: ActivityNet, H: HACS, T: THUMOS14, F: FineAction), with test datasets along the x-axis.

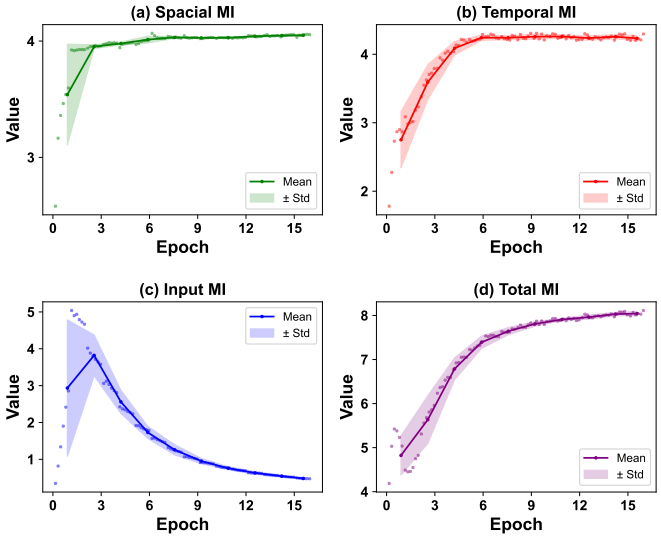

Fig. 5. Mean $\pm$ Std curves of mutual information (MI) values over training epochs, where $\mathcal{I}$ denotes the MI value. (a) Spatial MI $\mathcal{I}_{\text{Spatial}}$; (b) Temporal MI $\mathcal{I}_{\text{Temporal}}$; (c) Input MI $\mathcal{I}_{\text{Input}}$; (d) According to the information bottleneck theory, the total MI is formulated as $\mathcal{I}_{\text{total}} = \mathcal{I}_{\text{Spatial}} + \mathcal{I}_{\text{Temporal}} - \beta\mathcal{I}_{\text{Input}}$.

TABLE VII
SEARCH-COST AND PERFORMANCE COMPARISON ON ACTIVITYNET DATASET WITH THE SAME SEARCH SPACE.

Methods	Search Strategy	Costs (GPU hours)	mAP@IoU (%)			
			0.5	0.75	0.95	Avg.
DARTS [22]	Gradient	33.5	60.8	41.7	9.7	40.6
SPOS [59]	EA	75.7	61.6	42.7	9.6	41.2
FairNAS [79]	EA	78.9	62.8	43.8	9.9	42.4
ShapleyNAS [80]	Gradient	38.1	61.0	42.2	9.6	40.9
BossNAS [81]	EA	83.2	62.2	43.2	9.4	41.7
PO-NAS [82]	EA	89.4	63.1	44.4	10.6	42.8
Ours	IB-guided EA	63.1	63.4	44.6	11.1	43.1

In contrast, our method maintains higher consistency and robustness across datasets, yielding substantial ΔmAP improvements even in cross-domain scenarios (e.g., up to 1.9% when searching on ActivityNet and testing on HACS). This superior transferability is attributed to our method's emphasis on discovering architectures that capture more fundamental task-relevant spatial and temporal features through mutual information maximization, rather than relying on dataset-specific performance metrics in conventional black-box NAS approaches. These results demonstrate that our information-bottleneck-based objective function effectively guides the search toward architectures with better generalization capabilities, enabling the discovery of hybrid networks that excel across diverse video domains and action categories.

4) Search Costs: The complete architecture search of NAS-TAD on ActivityNet is accomplished in approximately 63.1 GPU hours on a single NVIDIA RTX 3090 GPU, comprising 25.3 hours of super-net training and 37.8 hours of IB-guided evolutionary search. To position this cost within the broader NAS methods, we further re-implement representative gradient-based and evolutionary-algorithm (EA) NAS frameworks on the proposed unified search space. The comparison results are reported in Table VII.

For gradient-based NAS methods, DARTS [22] and Shapley-NAS [80] consume the least computation, with 33.5 and 38.1 GPU hours, respectively. However, their continuous-relaxation optimization is poorly aligned with the discrete and heterogeneous Mamba-Transformer search space, yielding inferior average mAP values of only 40.6% and 40.9%, which trail NAS-TAD by 2.5% and 2.2%, respectively.

Within the EA-based methods, conventional fitness functions in SPOS [59], FairNAS [79], BossNAS [81], and PO-NAS [82] require executing the full detection pipeline (i.e., dense proposal generation, NMS, and mAP evaluation) for every sampled candidate, which drives the total cost to a range of 75.7 to 89.4 GPU hours. In contrast, our IB-guided fitness only needs a single forward pass to estimate the temporal and spatial mutual information terms, reducing the cost to 63.1 GPU hours and saving 16.6% to 29.4% of computation over the EA baselines. Consequently, our method attains the most favorable accuracy and efficiency trade-off among all examined NAS paradigms, which further confirms its practical applicability.

D. Convergence and Mutual Information Analysis

To analyze the convergence of our NAS-TAD framework, we focus on the behavior of mutual information throughout the training process. The key aspect to ensure convergence is the stability of MI values, since we employ variational inference to approximate intractable posterior distributions, while optimizing the model parameters via a maximum likelihood estimation (MLE) loss during training. This setup enables tractable MI estimation but necessitates verification of convergence. To this end, we plot the evolution of MI values across training epochs, as shown in Fig. 5.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2026.3720512

Fig. 6. Visualization of temporal action detection results on ActivityNet datasets across three representative scenarios: (a) daily activities, (b) sports, and (c) work environments. The colored bars represent detection outputs from different TAD methods (ActionFormer, Causal-TAD, and our NAS-TAD), with ground truth annotations shown above the predictions for reference. The red dashed boxes indicate missed detections.

In general, the MI values exhibit high variance in early epochs while becoming stable and convergent as training progresses. This instability arises from MLE's reliance on limited effective samples, such as noisy gradients, which lead to erratic inferred distributions in variational inference. As training progresses and cumulative data exposure refines the likelihood estimates, parameter updates become more stable, resulting in smoother, convergent MI trajectories. This pattern demonstrates the robustness and convergence of combining variational inference with MLE for MI computation.

Notably, from the perspective of information theory, both spatial MI (Fig. 5 (a)) and temporal MI (Fig. 5 (b)) display a consistent upward trend, gradually increasing before plateauing. This aligns with IB theory, where effective representations maximize mutual information with task-relevant aspects (e.g., spatial and temporal dependencies in video streams). The rising trends indicate that the model progressively extracts more relevant information in latent features.

In contrast, input MI (Fig. 5 (c)) initially rises, then declines, and eventually stabilizes. This is because, early in training, the model depends on raw input features, causing latent representations Z to closely resemble the input distribution X and thus increasing MI. As variational inference facilitates the abstraction of higher-level semantic features, the latent distribution Z diverges from the input distribution X, reducing redundancy and thus decreasing MI.

Finally, total MI (Fig. 5 (d)) steadily increases toward stability. This overall growth reflects the model's optimization of the trade-off parameter β, prioritizing relevant (spatial and temporal) information while compressing input noise. Such results also validate information bottleneck theory as a quantitative and interpretable guidance for the NAS process.

E. Detection Visualization

To demonstrate the effectiveness of the proposed NAS-TAD framework, temporal action detection results are visualized across three representative scenarios spanning daily activities, sports, and work environments, as illustrated in Fig. 6. In general, the proposed NAS-TAD consistently outperforms other representative TAD methods in accurately localizing action boundaries and across diverse scenarios, demonstrating the robustness and generalization of the automatically discovered architectures.

As detailed analysis of the detection results reveals, ActionFormer [38] tends to fragment continuous actions into multiple independent segments (as shown in Fig. 6(a)), since it lacks

sufficient spatial modeling capability to determine whether different temporal segments belong to the same action instance. Causal-TAD employs a parallel Transformer-Mamba architecture to achieve improved detection performance compared to the single-architecture ActionFormer through enhanced spatio-temporal modeling.

However, both ActionFormer and Causal-TAD exhibit detection failures for short-duration actions (as demonstrated in Fig. 6(c)), whereas the proposed NAS-TAD successfully localizes these challenging instances. This phenomenon can be partly attributed to the topology of the searched network architecture. Specifically, the shallow and intermediate stages of the discovered network are dominated by Mamba blocks, while Transformer blocks are mainly placed in the deeper stages. The early Mamba blocks process the input with linear-time selective scanning, which helps preserve the fine-grained temporal variations that mark the onset and offset of brief actions. The deeper Transformer blocks then aggregate longer-range context for category-level reasoning. Therefore, the searched topology suggests that short-duration action localization benefits from preserving transient boundary cues in early layers before introducing stronger global semantic aggregation in deeper layers. Furthermore, this topology provides a data-driven reference for manually designing Mamba-Transformer hybrids in temporal action detection and related video understanding tasks.

V. Conclusion

In this paper, we introduce NAS-TAD, the first neural architecture search framework for temporal action detection. Our framework systematically explores hybrid Mamba-Transformer architectures through a unified search space, addressing optimal spatial and temporal feature integration for TAD. The proposed information-bottleneck-based objective function provides interpretable guidance for architecture evaluation, moving beyond conventional black-box NAS approaches. Comprehensive experiments across four benchmark datasets demonstrate the superiority of our automatically discovered architectures, achieving new state-of-the-art performance. This work establishes a new paradigm for automated TAD system design and opens promising directions for neural network architecture design for video processing.

References

[1] Y. Liu, L. Wang, Y. Wang, X. Ma, and Y. Qiao, "Fineaction: A fine-grained video dataset for temporal action localization," *IEEE Transactions on Image Processing*, vol. 31, pp. 6937–6950, 2022.

[2] X. Liu, Q. Wang, Y. Hu, X. Tang, S. Zhang, S. Bai, and X. Bai, "End-to-end temporal action detection with transformer," *IEEE Transactions on Image Processing*, vol. 31, pp. 5427–5441, 2022.

[3] E. Vahdani and Y. Tian, "Deep learning-based action detection in untrimmed videos: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4302–4320, 2022.

[4] J. Xu, G. Chen, J. Lu, and J. Zhou, "Unintentional action localization via counterfactual examples," *IEEE Transactions on Image Processing*, vol. 31, pp. 3281–3294, 2022.

[5] L. Yang, J. Han, T. Zhao, N. Liu, and D. Zhang, "Structured attention composition for temporal action localization," *IEEE Transactions on Image Processing*, 2022, Early Access, doi: 10.1109/TIP.2022.3180925.

[6] M. Lu, X. Lu, and J. Liu, "Momentum contrastive teacher for semi-supervised skeleton action recognition," *IEEE Transactions on Image Processing*, vol. 34, pp. 295–305, 2025.

[7] B. Yang, M. Cao, and Y. Zou, "Concept-aware video captioning: Describing videos with effective prior information," *IEEE Transactions on Image Processing*, vol. 32, pp. 5366–5378, 2023.

[8] L. Chen, X. Wei, J. Li, X. Dong, P. Zhang, Y. Zang, Z. Chen, H. Duan, Z. Tang, L. Yuan *et al.*, "Sharegpt4video: Improving video understanding and generation with better captions," *Advances in Neural Information Processing Systems*, vol. 37, pp. 19 472–19 495, 2024.

[9] Y. Li, L. Chen, R. He, Z. Wang, G. Wu, and L. Wang, "Multisports: A multi-person video dataset of spatio-temporally localized sports actions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 536–13 545.

[10] L. Ventura, A. Yang, C. Schmid, and G. Varol, "Covr: Learning composed video retrieval from web video captions," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 38, no. 6, 2024, pp. 5270–5279.

[11] Z. Chen, X. Jiang, X. Xu, Z. Cao, Y. Mo, and H. T. Shen, "Joint searching and grounding: Multi-granularity video content retrieval," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 975–983.

[12] Y. Zheng, H. Huang, X. Wang, X. Yan, and L. Xu, "Spatio-temporal fusion for human action recognition via joint trajectory graph," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 7579–7587.

[13] L. Sui, C.-L. Zhang, L. Gu, and F. Han, "A simple and efficient pipeline to build an end-to-end spatial-temporal action detector," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 5999–6008.

[14] Y. Weng, Z. Pan, M. Han, X. Chang, and B. Zhuang, "An efficient spatio-temporal pyramid transformer for action detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 358–375.

[15] Y. Qiu, F. Sha, and L. Niu, "Efficient temporal attention with state space model for temporal action localization," in *International Conference on Neural Information Processing*. Springer, 2024, pp. 183–197.

[16] K. Li, X. Li, Y. Wang, Y. He, Y. Wang, L. Wang, and Y. Qiao, "Videomamba: State space model for efficient video understanding," in *European Conference on Computer Vision*. Springer, 2024, pp. 237–255.

[17] W. Huang, J. Zhang, G. Li, L. Zhang, S. Wang, F. Dong, J. Jin, T. Ogawa, and M. Haseyama, "Manta: Enhancing mamba for few-shot action recognition of long sub-sequence," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 39, no. 4, 2025, pp. 3751–3759.

[18] S. Liu, L. Sui, C.-L. Zhang, F. Mu, C. Zhao, and B. Ghanem, "Harnessing temporal causality for advanced temporal action detection," *arXiv preprint arXiv:2407.17792*, 2024.

[19] A. Hatamizadeh and J. Kautz, "Mambavision: A hybrid mamba-transformer vision backbone," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025, pp. 25 261–25 270.

[20] Y. Liu and L. Yi, "Map: Unleashing hybrid mamba-transformer vision backbone's potential with masked autoregressive pretraining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025, pp. 9676–9685.

[21] B. Zoph and Q. Le, "Neural architecture search with reinforcement learning," in *International Conference on Learning Representations*, 2017.

[22] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[23] L. Xu, Y. Guan, S. Jin, W. Liu, C. Qian, P. Luo, W. Ouyang, and X. Wang, "Vipnas: Efficient video pose estimation via neural architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 072–16 081.

[24] C. White, A. Zela, R. Ru, Y. Liu, and F. Hutter, "How powerful are performance predictors in neural architecture search?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 454–28 469, 2021.

[25] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[26] L. Wang, Y. Qiao, X. Tang *et al.*, "Action recognition and detection by combining motion and appearance features," *THUMOS14 Action Recognition Challenge*, vol. 1, no. 2, p. 2, 2014.

[27] H. Zhao, A. Torralba, L. Torresani, and Z. Yan, "Hacs: Human action clips and segments dataset for recognition and temporal localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8668–8678.

[28] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[29] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[30] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1049–1058.

[31] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017, pp. 2914–2923.

[32] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao, "A multi-stream bi-directional recurrent neural network for fine-grained action detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1961–1970.

[33] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, and W. Zeng, "Multi-modality multi-task recurrent neural network for online action detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, pp. 2667–2682, 2018.

[34] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3163–3172.

[35] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3202–3211.

[36] F. Cheng and G. Bertasius, "Tallformer: Temporal action localization with a long-memory transformer," in *European Conference on Computer Vision*. Springer, 2022, pp. 503–521.

[37] M. Yang, H. Gao, P. Guo, and L. Wang, "Adapting short-term transformers for action detection in untrimmed videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 570–18 579.

[38] C.-L. Zhang, J. Wu, and Y. Li, "Actionformer: Localizing moments of actions with transformers," in *European Conference on Computer Vision*. Springer, 2022, pp. 492–510.

[39] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7094–7103.

[40] X. Chen, Y. Guo, J. Liang, S. Zhuang, R. Zeng, and X. Hu, "Temporal action detection model compression by progressive block drop," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 29 225–29 236.

[41] R. Zeng, J. Mao, M. Lai, M. H. Phan, Y. Dong, W. Wang, Q. Chen, and X. Hu, "Ovg-hq: Online video grounding with hybrid-modal queries," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 21 085–21 096.

[42] G. Chen, Y. Huang, J. Xu, B. Pei, Z. Chen, Z. Li, J. Wang, K. Li, T. Lu, and L. Wang, "Video mamba suite: State space model as a versatile alternative for video understanding," *arXiv preprint arXiv:2403.09626*, 2024.

[43] A. Sinha, M. S. Raj, P. Wang, A. Helmy, and S. Das, "Ms-temba: Multi-scale temporal mamba for efficient temporal action detection," *arXiv preprint arXiv:2501.06138*, 2025.

[44] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4095–4104.

[45] S. Liu, C. Zheng, K. Lu, S. Gao, N. Wang, B. Wang, D. Zhang, X. Zhang, and T. Xu, "Evsrnet: Efficient video super-resolution with neural architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2480–2485.

[46] X. Yang, S. Wang, J. Dong, J. Dong, M. Wang, and T.-S. Chua, "Video moment retrieval with cross-modal neural architecture search," *IEEE Transactions on Image Processing*, vol. 31, pp. 1204–1216, 2022.

[47] Z. Wang, C. Lin, L. Sheng, J. Yan, and J. Shao, "Pv-nas: practical neural architecture search for video recognition," *arXiv preprint arXiv:2011.00826*, 2020.

[48] O. Lieber, B. Lenz, H. Bata, G. Cohen, J. Osin, I. Dalmedigos, E. Safahi, S. Meirom, Y. Belinkov, S. Shalev-Shwartz *et al.*, "Jamba: A hybrid transformer-mamba language model," *arXiv preprint arXiv:2403.19887*, 2024.

[49] Z. Fei, M. Fan, C. Yu, D. Li, Y. Zhang, and J. Huang, "Dimba: Transformer-mamba diffusion models," *arXiv preprint arXiv:2406.01159*, 2024.

[50] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *IEEE Information Theory Workshop*, 2015, pp. 1–5.

[51] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5156–5165.

[52] T. Dao and A. Gu, "Transformers are ssms: Generalized models and efficient algorithms through structured state space duality," in *Proceedings of the 41st International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 235, 2024, pp. 10 041–10 071.

[53] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *International Conference on Learning Representations (ICLR)*, 2017.

[54] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, "Club: A contrastive log-ratio upper bound of mutual information," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1779–1788.

[55] M. Contributors, "Openmmlab's next generation video understanding toolbox and benchmark," https://github.com/open-mmlab/mmaction2, 2020.

[56] S. Liu, C. Zhao, F. Zohra, M. Soldan, A. Pardo, M. Xu, L. Alssum, M. Ramazanova, J. L. Alcázar, A. Cioppa *et al.*, "Opentad: A unified framework and comprehensive study of temporal action detection," in *Proceedings of the Computer Vision and Pattern Recognition Conference Workshops*, 2025, pp. 2625–2635.

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

[58] M. Verma, M. S. K. Reddy, Y. R. Meedimale, M. Mandal, and S. K. Vipparthi, "Automer: Spatiotemporal neural architecture search for microexpression recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6116–6128, 2021.

[59] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun, "Single path one-shot neural architecture search with uniform sampling," in *European Conference on Computer Vision*. Springer, 2020, pp. 544–560.

[60] Y. Tang, M. Liu, B. Li, Y. Wang, and W. Ouyang, "Nas-ped: Neural architecture search for pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 3, pp. 1800–1817, 2025.

[61] X. Liu, S. Bai, and X. Bai, "An empirical study of end-to-end temporal action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 010–20 019.

[62] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6202–6211.

[63] J. Shao, X. Wang, R. Quan, J. Zheng, J. Yang, and Y. Yang, "Action sensitivity learning for temporal action localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 13 457–13 469.

[64] H. Liu, X. Li, B. Fan, and J. Xu, "Brtal: Boundary refinement temporal action localization via offset-driven diffusion models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 8, pp. 8174–8186, 2025.

[65] D. Shi, Y. Zhong, Q. Cao, L. Ma, J. Li, and D. Tao, "Tridet: Temporal action detection with relative boundary modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 857–18 866.

[66] C. Zhao, S. Liu, K. Mangalam, and B. Ghanem, "Re2tal: Rewiring pretrained video backbones for reversible temporal action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 637–10 647.

[67] Q. Li, D. Liu, J. Kong, S. Li, H. Xu, and J. Wang, "Temporal action localization with cross layer task decoupling and refinement," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 39, no. 5, 2025, pp. 4878–4886.

[68] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[69] H.-J. Kim, Y. Lee, J.-H. Hong, and S.-W. Lee, "Digit: Multi-dilated gated encoder and central-adjacent region integrated decoder for temporal action detection transformer," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 24 286–24 296.

[70] Z. Gao, W. Yang, Y. Zhao, C. Ma, C. Li, and R. Wang, "Lsgnet: A local-pattern separation and global-aware network for temporal action

detection," *IEEE Transactions on Image Processing*, vol. 35, pp. 4643–4658, 2026.

[71] L. Yang, Z. Zheng, Y. Han, H. Cheng, S. Song, G. Huang, and F. Li, "Dyfadet: Dynamic feature aggregation for temporal action detection," in *European Conference on Computer Vision*. Springer, 2024, pp. 305–322.

[72] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, and Y. Qiao, "Videomae v2: Scaling video masked autoencoders with dual masking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 549–14 560.

[73] Y. Wang, K. Li, X. Li, J. Yu, Y. He, G. Chen, B. Pei, R. Zheng, Z. Wang, Y. Shi *et al.*, "Internvideo2: Scaling foundation models for multimodal video understanding," in *European Conference on Computer Vision*. Springer, 2024, pp. 396–416.

[74] S. Liu, C.-L. Zhang, C. Zhao, and B. Ghanem, "End-to-end temporal action detection with 1b parameters across 1000 frames," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 591–18 601.

[75] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "Bmn: Boundary-matching network for temporal action proposal generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3889–3898.

[76] Z. Qing, H. Su, W. Gan, D. Wang, W. Wu, X. Wang, Y. Qiao, J. Yan, C. Gao, and N. Sang, "Temporal context aggregation network for temporal action proposal refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 485–494.

[77] J. Yang, P. Wei, Z. Ren, and N. Zheng, "Gated multi-scale transformer for temporal action localization," *IEEE Transactions on Multimedia*, vol. 26, pp. 5705–5717, 2023.

[78] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, "G-tad: Sub-graph localization for temporal action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 156–10 165.

[79] X. Chu, B. Zhang, and R. Xu, "Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 239–12 248.

[80] H. Xiao, Z. Wang, Z. Zhu, J. Zhou, and J. Lu, "Shapley-nas: Discovering operation contribution for neural architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 892–11 901.

[81] C. Li, T. Tang, G. Wang, J. Peng, B. Wang, X. Liang, and X. Chang, "Bossnas: Exploring hybrid cnn-transformers with block-wisely self-supervised neural architecture search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 281–12 291.

[82] M. Lin and J. Luo, "Per-architecture training-free metric optimization for neural architecture search," in *Advances in Neural Information Processing Systems*, 2025, pp. 92 152–92 193.