Frame-by-Frame Multi-object Tracking-Guided Video Captioning

HuiLan Luo, Xia Cai, and Lik-Kwan Shark

Abstract—**Video captioning through deep learning presents a multifaceted challenge that encompasses the extraction of complex spatio-temporal visual features and the synthesis of meaningful natural language descriptions. Most of the existing deep learning models can be broadly grouped as either convolution-based or transformer-based encoder-decoder networks, with video captions generated from features encoded at the pixel level for the former, and from features encoded at grid, frame, or video levels depending on encoder complexity for the latter. This paper advocates frame-level features as a more balanced and compact representation for fast caption generation, and introduces the Tracking-guided Information Augmentation for Captioning (Track4Cap) model, which integrates tracking-guided information augmentation to enhance frame-level features without relying on complex architectures or additional data modalities. Specifically, Track4Cap employs the Frame-by-Frame Multi-object Tracking module (FMoT) to identify the most relevant objects in the input video and the Object Relation Encoder (ORE) to model inter-object relationships as supplementary high-level cues for caption generation. By avoiding time-consuming end-to-end training and leveraging compact representations, Track4Cap achieves computational efficiency while improving captioning performance. Extensive experiments on two commonly used benchmark datasets demonstrate that Track4Cap not only achieves faster inference times but also outperforms state-of-the-art convolution-based and transformer-based video captioning models. The implementation of our method is publicly available at https://github.com/ccc000-png/Tracker4Cap.**

Index Terms—**Video captioning, Natural language descriptions, Deep learning, Spatio-temporal visual features, Frame-level features.**

I. INTRODUCTION

IN the early years of information transmission, images and text were the main mediums. However, with the rapid advancement of technology and multimedia, videos have become a new and powerful way of transmitting information. In this context, video captioning [1] came into being which aims to generate meaningful and coherent descriptions for video content. It plays an important role in daily life, with various applications such as assisting the visually impaired [2], enhancing human computer interaction [3], and improving video retrieval [4], [5].

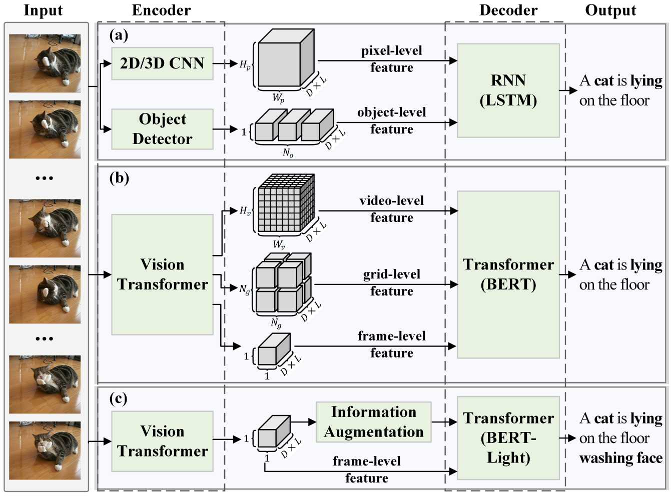


Fig. 1. Video captioning frameworks at different levels of information granularity: (a) Convolution-based models using pixel-level and object-level features; (b) Transformer-based models utilizing video-level, grid-level, or frame-level features; and (c) Proposed Track4Cap leveraging frame-level features with information augmentation.

Although video captioning shares a common foundation with image captioning [6] as both strive to capture and describe the essence of visual content, they differ significantly in terms of complexity, as video captioning expands upon image captioning by moving from interpreting a single static image to a sequence of frames over time. This shift requires understanding not only objects and their relationships in the spatial domain, but also their dynamic variations and contextual interactions in the spatio-temporal domain. Consequently, video captioning faces the added challenge of identifying the most relevant objects and their interactions within dynamic and intricate multi-component scenes, since the most salient object in a single frame may not represent the primary object of the entire video sequence, and not all object relationships are relevant for caption generation.

Most existing video captioning methods follow an encoder-decoder architecture. The encoder processes the input video to extract features, which are then mapped to textual descriptions by the decoder. These encoder-decoder frameworks can be broadly categorized into convolution-based and transformer-based models. Convolution-based models, as illustrated in

This work was supported by the Key Project of Natural Science Foundation of Jiangxi Province of China (No. 20232ACB202011), the National Natural Science Foundation of China (No. 62361032), Training Plan for Academic and Technical Leaders of Major Disciplines in Jiangxi Province of China (No. 20213BCJ22004), Jiangxi Province Key Laboratory of Multidimensional Intelligent Perception and Control of China (No. 2024SSY03161) and Jiangxi Province Graduate Innovation Special Fund Project of China (N0. YC2023-S657). *(Corresponding author: Xia Cai.)*

HuiLan Luo and Xia Cai are with the Jiangxi Province Key Laboratory of Multidimensional Intelligent Perception and Control, Jiangxi University of Science and Technology, Ganzhou 341000, China (e-mail: luohuilan@jxust.edu.cn; caixia0421@163.com).

Lik-Kwan Shark is with the School of Engineering and Computing, University of Central Lancashire, Preston PR1 2HE, UK (email: LShark@uclan.ac.uk).

Fig. 1(a), predominantly utilize pre-trained 2D and 3D convolutional neural networks (CNNs) [7], [8], [9], [10] to extract spatial and temporal features at the pixel level, where information is derived directly from the intensity or color values of individual pixels and their local neighborhoods. While providing detailed local appearance and motion information, they often suffer from high computational costs due to the large data volume and fine spatial granularity. Some convolution-based methods incorporate object detection modules [11], [12], [13] to extract salient object-level features, improving object-centric descriptions. However, these methods further increase computational overhead and are prone to errors in identifying primary objects in cluttered scenes.

Transformer-based models, as shown in Fig. 1(b), encode information at various levels of granularity depending on the encoder's complexity. Grid-level features represent coarse-grained information derived from non-overlapping patches of pixels, aggregating local details into compact representations that balance spatial resolution and computational efficiency [14], [15], [16]. Frame-level features capture the overall spatial context of individual frames, summarizing their appearance while disregarding temporal dependencies [17]. In contrast, video-level features provide a holistic global representation by integrating information across multiple frames, emphasizing long-term dependencies and overall video context [18], [19], [20].

While transformer-based models offer flexibility in processing different levels of granularity, they face significant challenges. Optimizing both encoder and decoder components during end-to-end training can be computationally demanding, and large-scale transformer-based models often require extensive training data and resources to effectively learn interdependencies between video content and captions. These challenges make it crucial to strike a balance between computational efficiency and representation quality.

To achieve a balance between computational efficiency and representation quality, this paper leverages frame-level features, which provide a compact, low-dimensional representation that significantly reduces computational overhead compared to pixel-, grid-, and video-level representations. Frame-level features effectively capture high-level contextual information but inherently lack the fine-grained spatio-temporal details necessary to model dynamic object interactions and generate accurate video captions. To address these limitations, we propose the Tracking-guided Information Augmentation for Captioning (Track4Cap) framework, as illustrated in Fig. 1(c).

Track4Cap integrates an intermediate information augmentation stage designed to enhance frame-level features with finer-grained spatio-temporal cues. This stage comprises two key modules: the Frame-by-Frame Multi-object Tracking (FMoT) module, which identifies and tracks salient objects across frames, and the Object Relation Encoder (ORE) module, which models inter-object relationships to provide high-level semantic cues. By augmenting frame-level features with dynamic and relational information, Track4Cap enables more accurate and contextually aware caption generation while maintaining computational efficiency.

The motivations for introducing the information augmentation stage are threefold. First, it enriches frame-level features with object-centric and relational information, effectively bridging the gap between simple frame-level encoding and the detailed modeling found in grid- or video-level representations. Second, it reduces computational costs by focusing on tracked objects rather than processing all pixel-level information in video frames. Third, it enhances contextual understanding by explicitly modeling interactions between tracked objects, which are critical for generating accurate and meaningful captions.

The main contributions of this paper are summarized as follows:

- We propose *Track4Cap*, a novel encoder-decoder framework that leverages frame-level features enriched with tracking-guided information for efficient and accurate video captioning.
- We introduce two core modules for information augmentation: (i) the *Frame-by-frame Multi-Object Tracking (FMoT)* module, which identifies and tracks salient objects, and (ii) the *Object Relation Encoder (ORE)*, which models inter-object relationships to enhance contextual understanding.
- We demonstrate that *Track4Cap* achieves state-of-the-art performance on benchmark datasets, with significant reductions in inference time and computational complexity. Extensive ablation studies validate the contributions of individual components and their combinations.

II. Related Works

In this section, a review of the latest video captioning models is provided, categorized into convolution-based models and transformer-based models based on their encoder architectures.

A. Convolution-based Models

The development of video captioning in deep learning began with stage-wise encoder-decoder frameworks that combined convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Typically, pre-trained 2D/3D CNNs were employed in the encoder to extract video features at the pixel level, with or without auxiliary object detection using models such as Region-based Convolutional Neural Networks (R-CNN) [11]. These pixel-level features were then used by RNN-based decoders, such as Long Short-Term Memory networks (LSTMs) [21], to generate captions.

Despite the effectiveness of 2D/3D CNNs in capturing local appearance and motion features, the lack of semantic representation in these features has been a significant limitation. Consequently, efforts have been directed toward enhancing spatial and temporal semantic representations within the encoder. For example, the Semantic Grouping Network (SGN) [22] improves spatial semantics by aligning video frames with partially decoded subtitles, reducing visual redundancy and aiding the decoder in more accurate word prediction. Similarly, the Motion Guided Region Message Passing (MGRMP) model [23] enhances temporal semantics by employing 3D CNNs to extract regional temporal features and model inter-regional relationships across frames.

On the decoder side, research has focused on refining low-level semantic information using high-level cues through knowledge distillation. For instance, the Decoder Refined Semantic enhancement towards Frequency Diffusion (RSFD) [24] employs frequency-aware diffusion to capture critical low-frequency semantic details, thereby improving captioning performance. The Two-Step Transformer-based Polishing Network (TSTPN) [25] combines 2D/3D CNNs with a generation transformer and a polishing transformer to enhance cross-modal sequence mapping via cross-modal attention and knowledge distillation.

Integrated improvements targeting both encoder and decoder architectures have also gained traction. The Hierarchical Representation Network with Auxiliary Tasks (HRNAT) [26] reconstructs visual content through auxiliary tasks such as cross-modality matching and syntax-guided learning, resulting in linguistically and grammatically accurate captions. Stay-in-Grid Video Captioning (SGCAP) [16] incorporates a bilinear sequential attention encoder for spatial-temporal modeling, complemented by a cross-modal sequential attention decoder for dynamic region representation.

Object detection has been another effective strategy to improve captioning by incorporating high-level cues about object relationships and interactions. For instance, the Object Relational Graph with Teacher-Recommended Learning (ORG-TRL) [27] leverages graph learning to represent object interactions, integrating these cues with an external language model through a teacher-recommended learning mechanism. The Spatio-Temporal Graph with Knowledge Distillation (STG-KD) [28] models object interactions across space and time, providing interpretable links while enhancing stability with a knowledge distillation framework. Transitive Visual Relationship Detection (TVRD) [29] introduces a novel action detection mechanism to extract deep semantic relationships between objects through object-action and object-object graphs. Additionally, the Long Short-Term Relation Transformer (LSRT) [30] captures both short-term spatial relations and long-term dependencies among objects, employing a global gating unit to regulate information flow.

While graph-based approaches effectively capture interaction information, they often introduce redundant connections that can reduce computational efficiency. To address this, some methods incorporate language semantics to refine object interactions. For example, Syntax-Aware Action Targeting (SAAT) [31] associates sentence structure with the actions of detected objects, providing precise action representations. Similarly, the Textual-Temporal Attention Model (TTA) [32] aligns visual tags of detected objects with descriptive words, improving the alignment of visual and textual information.

Recently, hierarchical and multi-branch networks have further advanced video captioning by integrating object-level semantics with scene-level information. For example, Reinforcement Learning with Hierarchical Modular Network (RLHMN) [33] learns multi-level visual representations—spanning entities, predicates, and sentences—thereby reducing ambiguities caused by polysemous verbs. Element-aware Video Captioning (EvCap) [34] integrates object, action, and scene features via a multi-branch encoder-decoder, enhancing the perception of specific elements through a post-fusion mechanism.

Despite these advancements, convolution-based approaches face significant computational challenges in capturing global spatio-temporal semantic features. However, their strength lies in their ability to extract fine-grained spatial and temporal details, which remain a crucial foundation for video captioning tasks.

B. Transformer-based Models

Transformers, known for their exceptional sequence processing capabilities, have transitioned successfully from natural language processing to image and video processing, establishing themselves as a dominant framework for video captioning. By extending the Swin Transformer originally designed for images, the Video Swin Transformer (VidSwin) incorporates the temporal dimension, enabling grid-level feature encoding [35]. VidSwin serves as the encoder backbone for several transformer-based video captioning models, including SwinBERT [1], Diverse Video Captioning by Adaptive Spatio-temporal Attention (VASTA) [36], and the Concept-aware and Task-specific model (CAT) [14], all of which employ Bidirectional Encoder Representations from Transformers (BERT) [37] as the decoder.

SwinBERT employs an end-to-end training strategy that integrates VidSwin and leverages sparse attention mechanisms to process video data efficiently, reducing redundancy during feature extraction. VASTA further optimizes encoding by dynamically selecting keyframes to minimize redundant information entering VidSwin. CAT enhances semantic representation by introducing a concept parser to extract high-level cues and a multi-modal graph to model relationships among visual, semantic, and textual features, enabling richer and deeper semantic representations in generated captions.

To capture more holistic spatio-temporal features, the Unified Video and Language Pre-training model (UniVL) [38] leverages large-scale pre-training to establish connections between videos and text, focusing on video-level representations. However, transformer-based video-level models face significant computational challenges, particularly during end-to-end training, where optimizing both encoder and decoder on large datasets often leads to slow convergence and inefficient loss utilization. To address these issues, the MEta Loss TRansformer (MELTR) [19] was introduced as a plug-in module that dynamically integrates multiple loss functions to improve the optimization balance between the encoder and decoder. While MELTR enhances performance, its additional computational overhead increases overall complexity. To alleviate these computational demands, CoCap [20] directly utilizes compressed video data, extracting video-level information from I-frames, motion vectors, and residuals. This approach significantly reduces the data volume and inference time, offering a more efficient alternative for video-level processing.

Recent advancements in cross-modal learning have spurred the development of transformer backbones for encoding features at various levels of granularity. The Contrastive Language-Image Pre-training model (CLIP) [39], built on the Vision Transformer (ViT) [40], specializes in frame-level

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. 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/TCSVT.2025.3541965

feature generation through extensive pre-training to associate images and text. CLIP-based video captioning models, such as CLIP4Clip [41] and Expectation-Maximization Contrastive Learning (EMCL) [42], demonstrate the utility of CLIP as an encoder. CLIP4Clip identifies the most relevant frames by comparing frame-level features with captions, while EMCL iteratively optimizes CLIP's feature space, yielding compact representations. Both methods have shown significant improvements across multiple evaluation metrics.

The Concept-awARE video captioning framework (CARE) [17] addresses the limitations of frame-level features by incorporating additional modalities, such as audio and text, and employing multimodal-driven concept detection to uncover latent video themes, thereby enhancing decoding performance. Hierarchical Semantic Representation and Aggregation (HSRA) [43] complements frame-level features by integrating finer-grained pixel-level visual information. HSRA reconfigures visual semantics into a hierarchical ”object-action-event” structure, effectively capturing key object details and dynamic global context.

The proposed Track4Cap model distinguishes itself from existing transformer-based frame-level approaches through its simplicity and computational efficiency. Unlike CLIP4Clip, EMCL, and HSRA, Track4Cap does not rely on additional ground-truth captions for feature enhancement learning. It also avoids using additional modalities, such as audio or text, integral to CARE, and does not incorporate finer-grained pixel-level visual information, as employed by HSRA. Instead, Track4Cap exclusively utilizes frame-level features and adopts a stage-wise framework that leverages a pre-trained CLIP model as the decoder without requiring additional fine-tuning. By integrating a high-level information augmentation module, Track4Cap enhances frame-level features with salient object-relation cues, effectively addressing the limitations of frame-level representations. This design achieves an optimal trade-off between computational efficiency and captioning performance, eliminating the need for complex encoders to model temporal dependencies or additional data modalities for caption quality improvement.

III. Methodology

A. Overall Architecture

As illustrated in Fig. 2 (a), the proposed Tracking-guided Information Augmentation for Captioning (Track4Cap) framework follows an encoder-decoder structure with an intermediate information augmentation stage. This design aims to balance computational efficiency and captioning performance by leveraging frame-level features enriched with tracking and relational information.

The encoder utilizes the ViT-L/14 variant of the pre-trained CLIP model to extract frame-level features, which capture the overall appearance of each video frame and map it to a compact latent space. This approach significantly reduces dimensionality compared to video-level or grid-level representations while preserving context-rich information.

For a video uniformly sampled into L frames, denoted as $I = \{I_1, I_2, ..., I_L\}$, the frame-level features are computed as:

$$\mathcal{F} = \{f_i^v\}_{i=1}^{L} = CLIP(I) \tag{1}$$

where $f_i^v \in \mathbb{R}^D$ represents the visual features of the i-th frame, D is the dimensionality of the latent space, and $\mathcal{F} \in \mathbb{R}^{L\times D}$ denotes the feature representation for all frames. The pre-trained CLIP model is used without fine-tuning, ensuring computational efficiency and leveraging its robust feature extraction capabilities.

Despite their efficiency, frame-level features have inherent limitations, including the inability to capture temporal object dynamics and inter-object relationships—critical for generating coherent and descriptive captions. Additionally, they may include redundant or irrelevant information due to the lack of explicit mechanisms for prioritizing salient objects. To address these shortcomings, an intermediate information augmentation stage is introduced.

The information augmentation stage comprises two complementary modules:

- **Frame-by-Frame Multi-object Tracking (FMoT):** This module dynamically identifies and tracks salient objects across frames using attention-based mechanisms. By focusing on relevant objects and updating their representations over time, FMoT ensures temporal coherence and minimizes redundancy.
- **Object Relation Encoder (ORE):** This module models inter-object relationships, capturing high-level semantic cues to enhance contextual understanding and enrich frame-level features with interaction information.

The enhanced features from FMoT and ORE are passed to a lightweight Caption Decoder, which generates coherent and contextually accurate captions. The decoder's streamlined design further ensures computational efficiency while maintaining high-quality outputs.

By effectively integrating tracking and relational information, Track4Cap addresses the limitations of frame-level features and achieves a superior trade-off between computational efficiency and captioning performance, as demonstrated in subsequent experiments.

B. Frame-by-Frame Multi-object Tracking (FMoT)

Video scenes often involve multiple objects interacting dynamically within complex and diverse environments. Not all objects or their parts are relevant for caption generation, and the most salient object in a single frame may not represent the primary object of the entire video. Furthermore, primary objects can exhibit significant variations in position, size, and visual characteristics across frames and may not appear consistently throughout the video sequence. These challenges hinder the direct use of frame-level features for accurate captioning, as they lack the mechanisms to dynamically focus on the most relevant objects while accounting for their temporal evolution.

To address these challenges, the proposed FMoT module is designed to track salient objects across frames, ensuring that their temporal and contextual significance is captured. The motivation behind FMoT lies in the need to bridge

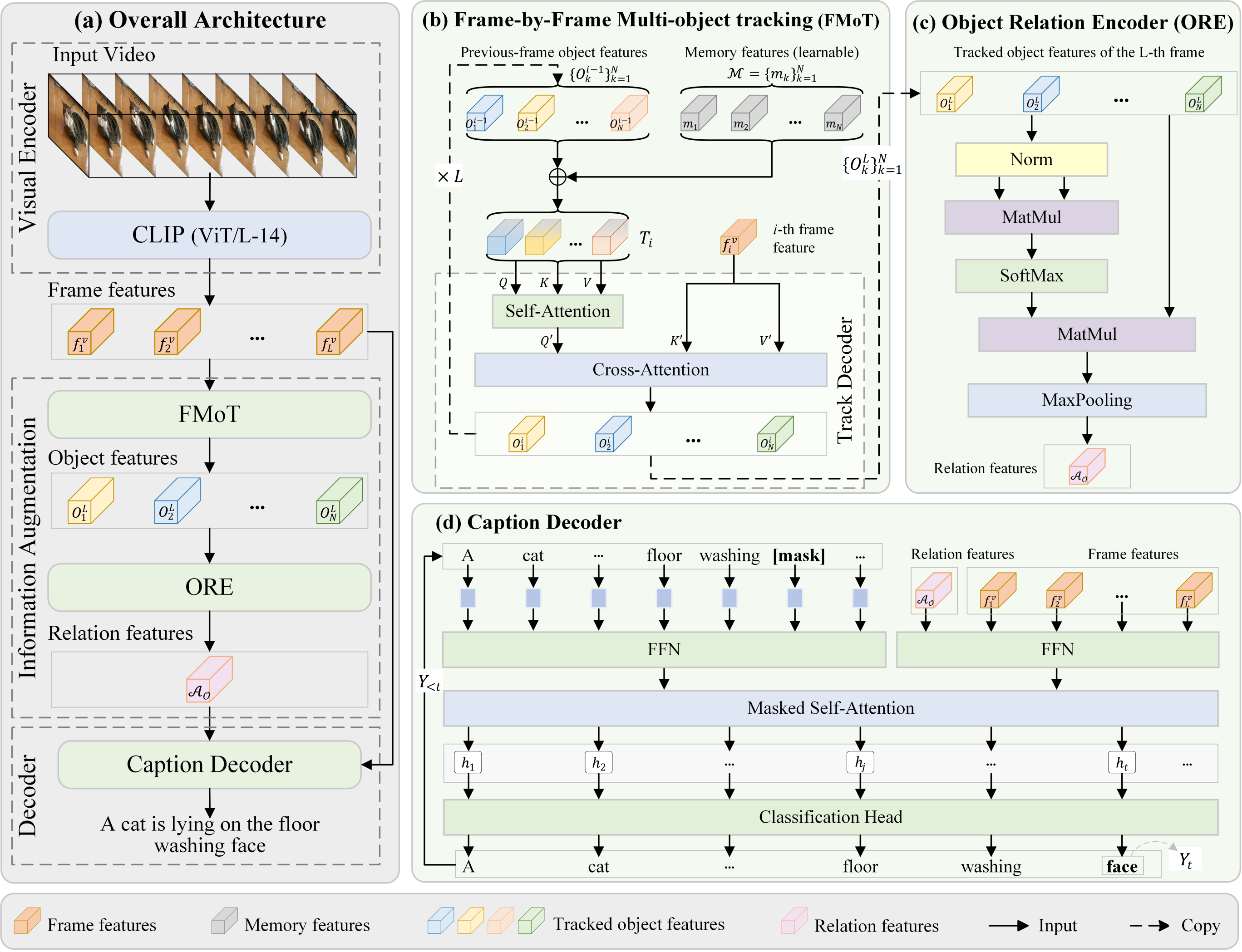


Fig. 2. Overview of the proposed Track4Cap framework and its key modules: (a) Overall architecture with a stage-wise design; (b) Frame-by-Frame Multi-object Tracking (FMoT) module for identifying and tracking salient objects; (c) Object Relation Encoder (ORE) module for modeling inter-object relationships; and (d) Caption Decoder, a lightweight BERT-based module with a single masked self-attention layer and no cross-attention layers for improved efficiency.

the gap between static frame-level representations and the dynamic nature of video content. By incorporating attention-based tracking mechanisms, FMoT selectively propagates and updates object features across frames, enabling a more precise and coherent understanding of object relevance over time. This approach not only reduces redundancy by focusing on key objects but also mitigates the effects of short-term dynamic variations in appearance.

As illustrated in Fig. 2 (b), FMoT operates as a loop iterating over a single Track Decoder for L frames. At the i-th iteration, the input to the Track Decoder consists of the frame-level features of the i-th frame, f_i^v, and the tracked object features from the previous frame, denoted as $\{O_k^{i-1}\}_{k=1}^N \in \mathbb{R}^{N\times D}$. Here, N represents the number of tracked objects. To initialize tracking, the frame-level features of the first frame are repeated N times to generate the initial object features.

The Track Decoder utilizes the standard query-key-value mechanism to perform self-attention and cross-attention operations. The self-attention mechanism identifies relevance and dependencies among tracked object features, emphasizing their significance for caption generation. To enhance focus on relevant objects, a randomly initialized learnable matrix, $\mathcal{M} = \{m_k\}_{k=1}^N \in \mathbb{R}^{N\times D}$, acts as a feature memory, storing statistical variations of feature patterns learned during training. During inference, $\mathcal{M}$ biases the relative importance of tracked object features. The track query for self-attention at the i-th iteration of the Track Decoder is defined as:

$$T_i = \begin{cases} \{f_1^v\}_{\times N} \oplus \mathcal{M}, & i = 1 \\ \{O_k^{i-1}\}_{k=1}^N \oplus \mathcal{M}, & i = 2, 3, ..., L \end{cases} \tag{2}$$

where $\oplus$ denotes element-wise addition.

The query, key, and value matrices for the track query are given by $Q = W_q T_i$, $K = W_k T_i$, and $V = W_v T_i$, respectively, where W_q, W_k, and W_v are learnable matrices optimized during training to minimize captioning errors. The self-attention output at the i-th iteration is computed as:

$$A_s^i = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right) V \tag{3}$$

where D is the dimensionality of the track query.

The cross-attention mechanism updates the tracked object features by associating them with the current frame-level features. Let $Q^{'}$, $K^{'}$, and $V^{'}$ denote the query, key, and value matrices of cross-attention. These matrices are derived as $Q^{'} = W^{'}_{q}A^{i}_{s}$, $K^{'} = W^{'}_{k}f^{v}_{i}$, and $V^{'} = W^{'}_{v}f^{v}_{i}$, where $W^{'}_{q}$, $W^{'}_{k}$, and $W^{'}_{v}$ are learnable matrices. The cross-attention output at the i-th iteration is expressed as:

$$\{O^{i}_{k}\}^{N}_{k=1} = \text{softmax}\left(\frac{Q^{'}K^{'T}}{\sqrt{D}}\right)V^{'} \tag{4}$$

The cross-attention output is fed back into the next iteration, enabling the progressive tracking and updating of object features across frames. After L iterations, the final output of FMoT, denoted as $\{O^{L}_{k}\}^{N}_{k=1}$, encapsulates the most relevant object features by considering long-term dependencies and relationships while remaining robust to short-term dynamic variations. By dynamically focusing on key objects and propagating their relevance across frames, FMoT addresses the limitations of static frame-level representations, thereby enhancing the contextual accuracy of video captions.

C. Object Relation Encoder (ORE)

The Object Relation Encoder (ORE) module enhances the tracked object features from FMoT by extracting high-level relational cues that model inter-object relationships, addressing the limitations of frame-level features in capturing dynamic interactions. By enriching the contextual representation of object interactions, ORE plays a critical role in generating coherent and descriptive video captions.

The motivation for ORE stems from the need to bridge the gap between static frame-level representations and the dynamic nature of video content. Frame-level features alone lack explicit modeling of spatial and semantic relationships among objects, which are essential for understanding complex scenes. To address this, ORE utilizes tracked object features to infer inter-object dependencies, augmenting the frame-level features with relational information that better reflects the interactions within a scene.

ORE is designed with computational efficiency as a priority. It employs parameter-free matrix operations to model inter-object relationships, avoiding the additional overhead associated with learnable parameters. This design not only ensures minimal computational complexity but also facilitates seamless integration into the Track4Cap framework. By augmenting frame-level features with object-centric and relational information, ORE enhances the contextual understanding necessary for accurate video captioning.

As illustrated in Fig. 2 (c), ORE consists of two key matrix multiplication operations (denoted as MatMul). The first operation computes the matrix multiplication of the normalized tracked object features, followed by a softmax function to derive an object relation weight matrix:

$$\mathcal{R}_{O} = softmax\left(\frac{MatMul(O^{L},(O^{L})^{T})}{||O^{L}||\cdot||O^{L}||}\right) \tag{5}$$

where $O^{L} = norm\{O^{L}_{k}\}^{N}_{k=1}$ represents the normalized tracked object features from FMoT, and $\mathcal{R}_{O} \in \mathbb{R}^{N\times N}$ is an affinity matrix. Each element $\mathcal{R}_{O}(i,j)$ quantifies the relational strength between objects i and j in the frame-level latent space.

Objects with strong semantic relationships—those that frequently co-occur during training—are mapped to nearby regions in the joint embedding space of visual and textual data. This proximity reflects their semantic ties, allowing $\mathcal{R}_{O}$ to encode the relative strength of pairwise inter-object relationships. Consequently, $\mathcal{R}_{O}$ captures the contextual importance of object interactions, enriching the representation of tracked object features for improved video captioning.

The second matrix multiplication aggregates the relationships encoded in $\mathcal{R}_{O}$ to provide supplementary cues for interaction inference. Specifically, the matrix multiplication between $\mathcal{R}_{O}$ and O^{L} is computed, followed by a max-pooling operation that selects the most significant value for each feature dimension across all objects:

$$\mathcal{A}_{O} = maxpool(MatMul(\mathcal{R}_{O},O^{L})) \in \mathbb{R}^{1\times D} \tag{6}$$

This operation performs a weighted aggregation, where each row represents the influence of other objects on the tracked object features based on their relationships in $\mathcal{R}_{O}$. The max-pooling step extracts the most salient features from these aggregated relationships, providing high-level semantic cues that enhance the contextual understanding of object interactions.

ORE enriches frame-level features with dynamic relational information, enhancing video captioning performance while maintaining efficiency by avoiding additional learnable parameters.

D. Caption Decoder

The structure of the Caption Decoder, shown in Fig. 2 (d), is based on the BERT [37] architecture and is designed to iteratively generate the video caption $Y = \{y_1, y_2, \ldots, y_T\}$ of length T word by word. At each time step, the word y_t is produced based on (a) the frame-level feature representation from the encoder, (b) the augmented object relation features from ORE, and (c) the previously generated words denoted by $Y_{<t}$, leveraging masked self-attention.

To ensure computational efficiency, the Caption Decoder employs a lightweight design by retaining only one masked self-attention layer while removing cross-attention layers from the original BERT structure. This streamlined approach significantly reduces the computational overhead while preserving the ability to model dependencies between visual features and textual inputs.

Specifically, the frame-level feature representation and object relation features from ORE are concatenated to form a unified visual representation $\tilde{V} \in \mathbb{R}^{(L+1)\times D}$:

$$\tilde{V} = [\mathcal{A}_{O};\mathcal{F}] \tag{7}$$

where $\mathcal{A}_{O}$ represents the aggregated object relation features, and $\mathcal{F}$ denotes the frame-level features.

Both the previously generated text and the visual representation are passed through separate feedforward networks, each consisting of a fully connected layer and a rectified

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. 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/TCSVT.2025.3541965

linear unit (ReLU) activation function. This provides enhanced embeddings for text ($T^{'}$) and visual features ($\tilde{V}^{'}$) as follows:

$$T^{'} = ReLU(W_T \cdot \delta(Y_{<t}) + b_T) \tag{8a}$$

$$\tilde{V}^{'} = ReLU(W_V \cdot \tilde{V} + b_V) \tag{8b}$$

where $\delta(Y_{<t})$ represents one-hot vectors of the previously generated words, and W_T, W_V, b_T, and b_V are learnable parameters.

The text and visual embeddings are concatenated to form a combined representation:

$$X = [\tilde{V}^{'}; T^{'}] + E_p + E_t \tag{9}$$

where E_p and E_t represent positional and type embeddings, respectively, to distinguish between visual and textual inputs.

Masked self-attention is applied to the combined representation to compute the hidden state at time step t:

$$h_t = softmax\left(\frac{Q_m K_m^T + M}{\sqrt{D}}\right) V_m \tag{10}$$

where M masks subsequent words, and $Q_m = W_q^{'} X$, $K_m = W_k^{'} X$, and $V_m = W_v^{'} X$ are the query, key, and value matrices with learnable weights.

The hidden state h_t is passed to a classification head consisting of a feedforward layer and a softmax function to generate a probability distribution over the vocabulary:

$$P(y_t | Y_{<t}, \tilde{V}) = softmax(W^{cls} h_t) \tag{11}$$

where $P(y_t | Y_{<t}, \tilde{V}) \in \mathbb{R}^{|w|}$ is the probability of each word in the vocabulary, $|w|$ is the vocabulary size, and $W^{cls} \in \mathbb{R}^{D \times |w|}$ is a learnable parameter matrix.

The word with the highest probability is selected as the next word in the caption. This process is repeated iteratively until the entire caption is generated. By simplifying the architecture and removing redundant layers, the Caption Decoder achieves a balance between efficiency and performance, aligning with the lightweight design principles of the Track4Cap framework.

E. Optimization

The proposed model employs stage-wise training to enhance computational efficiency, with the pre-trained encoder parameters kept frozen during training. The model is optimized using the standard cross-entropy loss, which measures the divergence between the generated caption and the ground-truth caption. The loss function is defined as:

$$\mathcal{L}_{cap} = -\sum_{t=1}^{T} \delta(y_t^*) \log P(y_t \mid Y_{<t}, \tilde{V}) \tag{12}$$

where y_t^* represents the t-th word of the ground-truth caption, and $\delta(y_t^*) \in \mathbb{R}^{|w|}$ is its one-hot encoding. Here, $P(y_t \mid Y_{<t}, \tilde{V})$ denotes the probability of generating the word y_t given the previously generated words and the visual representation $\tilde{V}$.

IV. EXPERIMENTS

This section outlines the experimental setup, evaluates the proposed Track4Cap model against recent state-of-the-art methods discussed in section II, and presents ablation studies to analyze the contributions of individual modules.

A. Experimental Setup

1) Datasets: Experiments were conducted on two benchmark datasets: Microsoft Video Description Corpus (MSVD) [44] and Microsoft Research Video to Text (MSRVTT) [45]. MSVD comprises 1,970 video clips with approximately 80K captions, divided into 1,200, 100, and 670 clips for training, validation, and testing, respectively, following prior works [16], [20], [33], [43]. MSRVTT contains 10,000 open-domain video clips, split into 6,513, 497, and 2,990 clips for training, validation, and testing, respectively, using the official split.

2) Evaluation metrics: Performance was evaluated using four standard metrics: BLEU-4 (B4) [46] for syntactic accuracy, METEOR (M) [47] for semantic relevance, ROUGE-L (R) [48] for word overlap, and CIDEr (C) [49] for overall relevance. Metric scores were computed using the Microsoft COCO Assessment Server.

3) Implementation details: Videos were scaled to 1,600 frames, and 16 frames were uniformly sampled per video (hyperparameter $L = 16$ in section III) to ensure consistent representation across varying durations. Uniform sampling mitigated biases by evenly capturing content throughout the video, avoiding overemphasis on specific segments. The ViT-L/14 variant of CLIP was used as the encoder, extracting frame-level features with a fixed feature dimension of 768 (hyperparameter $D = 768$ in section III). The encoder parameters were frozen to utilize the pre-trained CLIP model without fine-tuning.

The number of tracked objects (hyperparameter N in section III) was set to 3 for MSVD (single-activity videos) and 4 for MSRVTT (multi-activity videos), as determined by ablation studies. The Caption Decoder utilized a word embedding dimension of 768 to align with the frame-level features, and the vocabulary size was set to 49,408.

Training was conducted using the Adam optimizer [50] with an initial learning rate of 10^{-4}, a batch size of 128, and 20 epochs.

B. Performance Comparison with State-of-the-Art Methods

Table I compares the video captioning performance of the proposed Track4Cap model with recent state-of-the-art (SOTA) methods, categorized into convolution-based models (with or without object detection) and transformer-based models, evaluated on the MSVD and MSRVTT datasets. The comparison highlights both similarities and differences in performance trends across approaches, as well as the unique advantages of Track4Cap. The results are summarized below.

1) Comparison with convolution-based models without object detection: The proposed Track4Cap model achieves first place on seven out of eight metric scores across the MSVD and MSRVTT datasets, with the exception of a lower BLEU-4 (B4) score on the MSRVTT dataset compared to TSTPN [25]. Among convolution-based models without object detection, TSTPN achieves the best performance on both MSVD and MSRVTT, leveraging its combination of 2D/3D CNNs with a generation transformer and a polishing transformer to enhance cross-modal sequence mapping. Similar to TSTPN, Track4Cap focuses on efficient feature representation

TABLE I

COMPARISON OF VIDEO CAPTIONING PERFORMANCE ON THE MSVD AND MSRVTT DATASETS IN TERMS OF BLEU-4 (B4), METEOR (M), ROUGE-L (R), AND CIDEr (C) SCORES; AND MODEL EFFICIENCY IN TERMS OF THE NUMBER OF PARAMETERS (PARAMS), COMPUTATIONAL COMPLEXITY (FLOPS), AND INFERENCE TIME (TIME). RED INDICATES THE BEST SCORE, AND CYAN INDICATES THE SECOND-BEST SCORE.

Method	**Visual Encoder**	**Model Efficiency**			**MSVD**				**MSRVTT**			
		Params ↓	**FLOPs ↓**	**Time ↓**	**B4 ↑**	**M ↑**	**R ↑**	**C ↑**	**B4 ↑**	**M ↑**	**R ↑**	**C ↑**
Convolution-based Models Without Objection Detection												
SGN (2021) [22]	R101 + C3D	15M	1.047G	0.094s	52.8	35.5	72.9	94.3	40.8	28.3	60.8	49.5
MGRMP (2021) [23]	IRV2+3D-RX	-	-	0.410s	55.8	36.9	74.5	98.5	41.7	28.9	62.1	51.4
HRNAT (2022) [26]	IRV2+I3D	53M	2.729G	0.056s	55.7	36.8	74.1	98.1	42.1	28.0	61.6	48.2
TSTPN (2022) [25]	RNX+ECO	-	-	-	60.6	38.9	75.7	110.2	48.3	29.9	64.3	54.1
SGCAP (2023) [16]	IRV2+C3D	-	-	0.340s	-	-	-	-	44.5	28.7	62.6	51.6
RSFD (2023) [24]	R101+3D-RX	29M	1.646G	102.8s	51.2	35.7	72.9	96.7	43.4	29.3	62.3	53.1
Convolution-based Models With Object Detection												
ORG-TRL (2020) [27]	IRV2+C3D+Faster	-	-	0.250s	54.3	36.4	73.9	95.2	43.6	28.8	62.1	50.9
SAAT (2020) [31]	IRV2+C3D+Faster	29M	4.018G	0.032s	46.5	33.5	69.4	81.0	39.9	27.7	61.2	51.0
STG-KD (2020) [28]	R101+I3D+Faster	-	60.00G	-	52.2	36.9	73.9	93.0	40.5	28.3	60.9	47.1
TTA (2021) [32]	R152+C3D+Faster	-	-	-	51.8	35.5	72.4	87.7	41.4	27.7	61.1	46.7
TVRD (2022) [29]	IRV2+C3D+Faster	226M	190.3G	-	50.5	34.5	71.7	84.3	43.0	28.7	62.2	51.8
LSRT (2023) [30]	IRV2+I3D+Faster	-	-	-	55.6	37.1	73.5	98.5	42.6	28.3	61.0	49.5
RLHMN (2024) [33]	IRV2+C3D+Faster	76M	34.00G	0.043s	59.9	36.2	74.2	104.7	45.1	28.8	63.6	54.2
EvCap (2024) [34]	R101+I3D+Faster	-	0.780G	-	53.6	36.7	74.3	107.2	45.5	29.4	64.5	53.8
Transformers-based Models												
SWINBERT (2022) [1]	VidSwin	225M	1154G	0.498s	55.7	39.6	75.7	109.4	41.9	29.8	62.1	53.7
VASTA (2022) [36]	VidSwin	-	-	-	56.0	39.1	74.5	106.3	43.4	30.2	62.5	55.0
CAT (2023) [14]	VidSwin	-	-	-	59.9	41.7	78.4	122.9	42.1	30.2	62.5	54.5
MELTR (2023) [19]	UniVL	199M	24.40G	0.079s	-	-	-	-	44.2	29.3	62.4	52.8
CoCap (2023) [20]	H.264+CLIP	441M	102.5G	0.259s	60.1	41.4	78.2	121.5	44.4	30.3	63.4	57.2
EMCL (2022) [42]	CLIP	-	-	-	-	-	-	-	45.3	30.2	63.2	54.6
HSRA (2024) [43]	CLIP+C3D+Faster	128M	-	0.190s	62.2	39.2	78.4	110.1	46.9	30.9	64.8	55.3
Track4Cap	CLIP	91M	1.322G	0.022s	62.1	42.5	79.8	127.2	44.6	30.5	63.6	57.7

- C3D: Convolutional 3D, I3D: Inflated 3D ConvNet, IRV2: Inception Resnet V2, RNX: ResNeXt, R101: ResNet-101, R152: ResNet-152, 3D-RX: 3D ResNeXt, Faster: Faster R-CNN, H.264: Advanced Video Coding (AVC), a widely used video compression standard.

but extends this with its information augmentation modules, enabling it to better capture object interactions and achieve higher overall relevance.

2) Comparison with convolution-based models with object detection: The proposed Track4Cap model achieves first place in six out of eight metric scores across the MSVD and MSRVTT datasets, with a slightly lower ROUGE-L (R) score on the MSRVTT dataset compared to EvCap [34] and a slightly lower BLEU-4 (B4) score than both EvCap and RLHMN [33]. Among convolution-based models with object detection, EvCap integrates object, action, and scene features through a multi-branch encoder-decoder and achieves the best ROUGE-L (R) and CIDEr (C) scores on the MSVD dataset and the highest BLEU-4 (B4), METEOR (M), and ROUGE-L (R) scores on the MSRVTT dataset. Interestingly, as shown in Table I, compared with convolution-based models without explicit object detectors, incorporating object-level information via explicit detection does not necessarily guarantee performance improvements. Track4Cap achieves superior results without relying on explicit object detection, instead leveraging its Frame-by-Frame Multi-object Tracking (FMoT) and Object Relation Encoder (ORE) modules to infer relationships dynamically, which avoids the dependency on pre-detected object features seen in EvCap.

3) Comparison with transformer-based models: The proposed Track4Cap model stands out as the only transformer-based approach to achieve the best performance in four evaluation metrics and the second-best performance in two metrics across the MSVD and MSRVTT datasets. Track4Cap consistently attains the highest CIDEr scores, a metric strongly aligned with human judgment, highlighting its ability to generate captions of superior overall quality. Furthermore, Track4Cap is the most efficient among transformer-based models, requiring fewer parameters, lower FLOPs (Floating Point Operations Per Second), and shorter inference times. In comparison, the second-best method, HSRA [43], integrates pixel-level information with frame-level features and uses ground-truth captions for hierarchical feature supervision. While HSRA achieves the highest performance in three out of eight metrics, its reliance on additional pixel-level details and caption supervision contrasts with Track4Cap's simpler yet effective design.

As observed in Table I, transformer-based models generally outperform convolution-based models due to their superior capacity to model long-range dependencies. Among grid-level feature methods utilizing the VidSwin encoder, CAT [14] performs better on MSVD compared to SWINBERT [1] and VASTA [36], highlighting the benefits of multi-modal semantic representations. However, MELTR [19], despite employing a more complex video-level encoder, demonstrates the lowest METEOR (M) and CIDEr (C) scores among transformer-based models, suggesting diminishing returns for excessive architectural complexity. CoCap [20], leveraging H.264 video compression to extract video-level features with

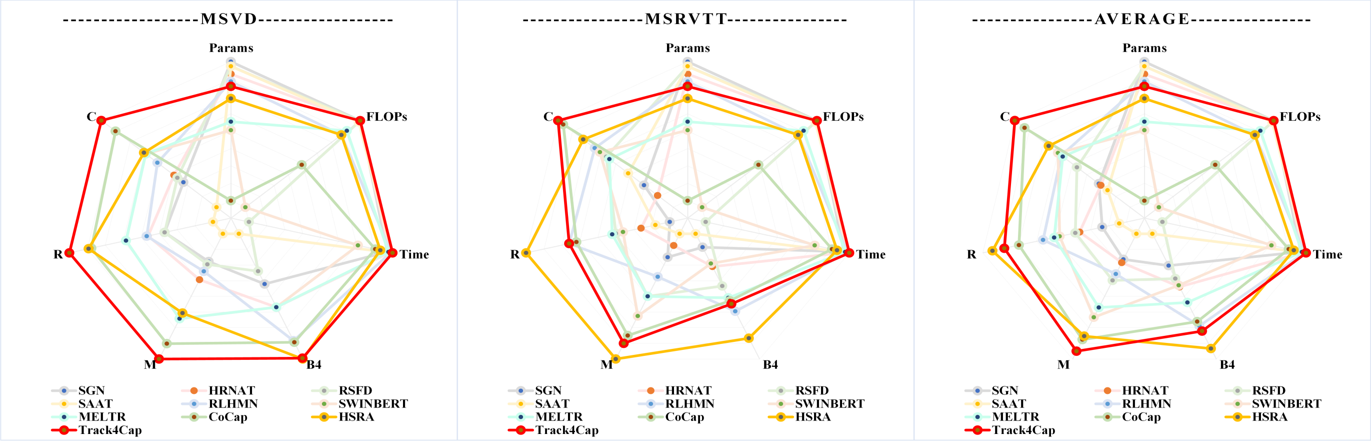


Fig. 3. Comparison of overall performance (BLEU-4 (B4), METEOR (M), ROUGE-L (R), and CIDEr scores) and computational complexity (parameters (Params), FLOPs, and inference time (Time)) across the MSVD and MSRVTT datasets, as well as the averaged scores across the two datasets. All metrics were normalized to a scale of 0–1 for comparability, with higher values indicating better performance.

CLIP, achieves better results across four metrics on MSRVTT compared to MELTR and the three grid-level feature methods using VidSwin. This emphasizes the practicality and efficiency of compressed video representations. Track4Cap, in contrast, adopts a simpler framework but achieves better results by focusing on efficient frame-level representations augmented with relational cues.

Additionally, EMCL [42], by iteratively optimizing CLIP's feature space, achieves superior BLEU-4 (B4) scores on MSRVTT compared to the three grid-level feature methods using VidSwin and the two video-level feature methods, underscoring the effectiveness of enhancing frame-level representations for compactness and accuracy. In contrast, Track4Cap outperforms EMCL on all metrics by leveraging a more direct augmentation strategy with fewer preprocessing steps, streamlining the feature extraction process.

In summary, Track4Cap demonstrates that leveraging frame-level features with efficient information augmentation can outperform more complex models that rely on additional modalities, hierarchical features, or compressed representations. These results reinforce the advantages of a well-optimized design in balancing computational demands with high-quality performance, achieving top accuracy and efficiency simultaneously.

C. Computational Performance

The computational complexity of the proposed Track4Cap model was quantitatively compared with recent state-of-the-art (SOTA) methods, as summarized in Table I. Track4Cap demonstrates the fastest inference speed, along with relatively low parameter counts and computational demands. Specifically, among transformer-based methods, Track4Cap achieves the lowest parameters, FLOPs, and inference times. Parameters (Params) represent the model size and its inference capacity, while FLOPs indicate the computational cost, offering insights into hardware requirements. Inference time measures the model's runtime efficiency during inference. These metrics collectively highlight the computational efficiency of Track4Cap.

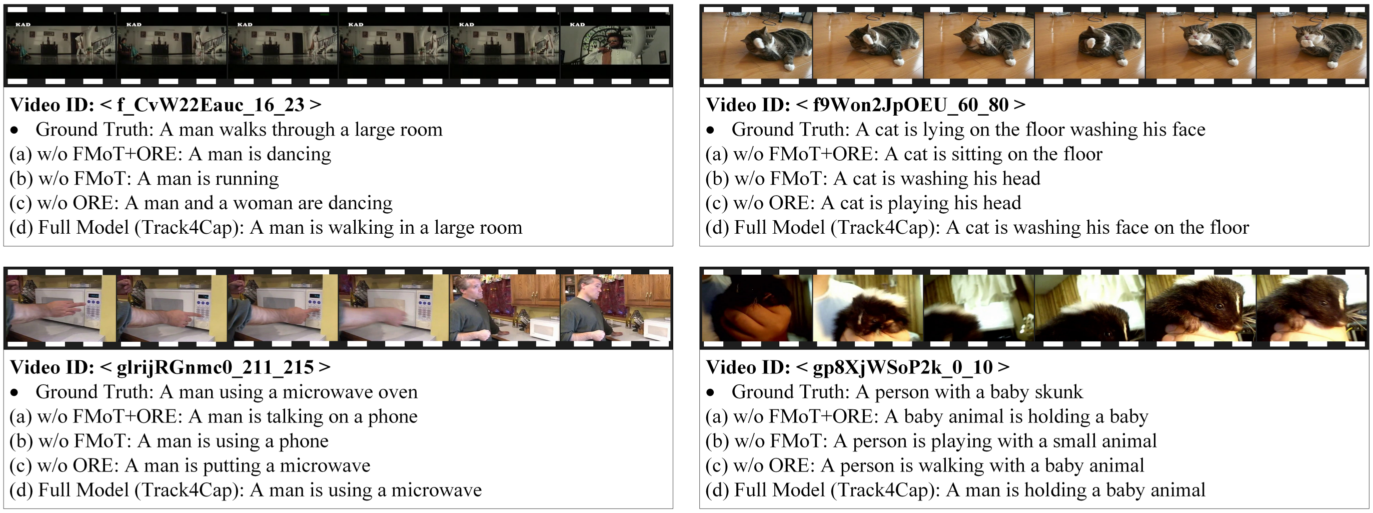
To evaluate the trade-off between effectiveness and efficiency, a comprehensive comparison of captioning performance and computational metrics was conducted. A radar chart in Fig. 3 illustrates the effectiveness in terms of BLEU-4 (B4), METEOR (M), ROUGE-L (R), and CIDEr scores, alongside efficiency metrics, including parameters (Params), FLOPs, and inference time (Time), on the MSVD and MSRVTT datasets, as well as the averaged scores across the two datasets. Track4Cap was compared with five convolution-based models [22], [24], [26], [31], [33] and four transformer-based models [1], [19], [20], [43].

To ensure direct comparability in the radar chart, all metrics were normalized to a scale of 0–1, with higher values indicating better performance. As shown in Fig. 3, Track4Cap outperforms current SOTA methods in the overall trade-off between captioning performance and inference efficiency on the MSVD dataset, achieving the highest overall generation quality with the lowest computational cost. On the MSRVTT dataset, both Track4Cap and HSRA [43] demonstrate strong performance, with Track4Cap achieving higher CIDEr scores and lower FLOPs and inference times. When averaging across the two datasets, Track4Cap achieves the best overall trade-off, highlighting its effectiveness in balancing captioning quality with computational efficiency.

D. Ablation Study

This section presents a series of ablation experiments to evaluate the impact of the proposed modules, the number of tracked objects, different configurations, and types of visual features on video captioning performance. The ablation analysis aims to highlight the contributions of each component and their roles in enhancing the overall effectiveness of the Track4Cap model.

1) Proposed Modules: The contributions of the proposed FMoT and ORE modules were evaluated through ablation experiments, with results summarized in Table II. The inclusion and exclusion of each module are indicated by "✓" and "×," respectively.



Video ID: < f_CvW22Eauc_16_23 >
- Ground Truth: A man walks through a large room

(a) w/o FMoT+ORE: A man is dancing
(b) w/o FMoT: A man is running
(c) w/o ORE: A man and a woman are dancing
(d) Full Model (Track4Cap): A man is walking in a large room

Video ID: < f9Won2JpOEU_60_80 >
- Ground Truth: A cat is lying on the floor washing his face

(a) w/o FMoT+ORE: A cat is sitting on the floor
(b) w/o FMoT: A cat is washing his head
(c) w/o ORE: A cat is playing his head
(d) Full Model (Track4Cap): A cat is washing his face on the floor

Video ID: < glrijRGnmc0_211_215 >
- Ground Truth: A man using a microwave oven

(a) w/o FMoT+ORE: A man is talking on a phone
(b) w/o FMoT: A man is using a phone
(c) w/o ORE: A man is putting a microwave
(d) Full Model (Track4Cap): A man is using a microwave

Video ID: < gp8XjWSoP2k_0_10 >
- Ground Truth: A person with a baby skunk

(a) w/o FMoT+ORE: A baby animal is holding a baby
(b) w/o FMoT: A person is playing with a small animal
(c) w/o ORE: A person is walking with a baby animal
(d) Full Model (Track4Cap): A man is holding a baby animal

Fig. 4. Qualitative analysis of the impact of augmented information from FMoT and ORE on the proposed Track4Cap model's video captioning performance using examples from the MSVD dataset. Each example includes six extracted video frames, the ground truth caption, and generated captions produced under different module configurations: (a) without FMoT and ORE (w/o FMoT and ORE), (b) with ORE only (w/o FMoT), (c) with FMoT only (w/o ORE), and (d) with both FMoT and ORE (Full Model).

TABLE II
ABLATION STUDIES OF THE PROPOSED MODULES ON THE MSVD AND MSRVTT DATASETS IN TERMS OF BLEU-4 (B4), METEOR (M), ROUGE-L (R), AND CIDEr (C) SCORES.

Module		MSVD				MSRVTT			
FMoT	ORE	B4	M	R	C	B4	M	R	C
×	×	60.5	41.6	79.2	122.8	42.1	29.8	59.7	54.9
×	✓	61.3	41.6	79.0	124.1	43.2	30.2	62.8	56.0
✓	×	61.3	42.0	79.5	125.8	44.0	30.4	63.5	57.4
✓	✓	**62.1**	**42.5**	**79.8**	**127.2**	**44.6**	**30.5**	**63.6**	**57.7**

The first row in Table II corresponds to the baseline model without FMoT and ORE, where captions are generated directly from frame-level features without supplementary cues. This configuration yields the lowest scores across all metrics for both datasets, emphasizing the necessity of information augmentation.

The second row represents the inclusion of ORE without FMoT, using object relation features derived from the last encoded frame as supplementary cues. While this improves performance over the baseline, particularly on MSRVTT, it shows limitations in METEOR and ROUGE scores on MSVD, likely due to single-frame-derived object relations lacking temporal coherence.

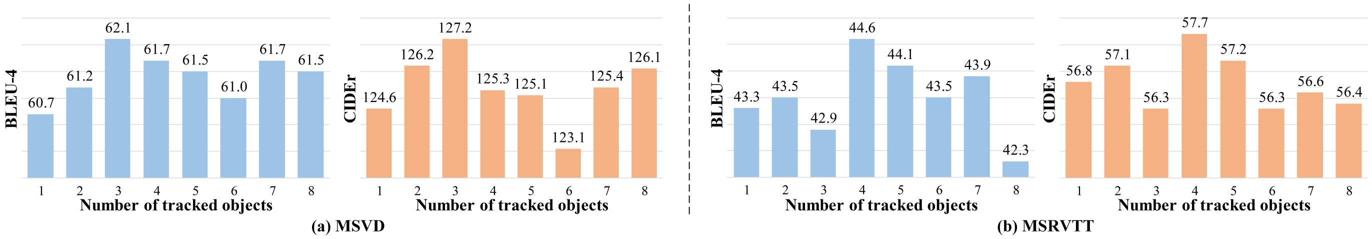
The third row demonstrates results for FMoT without ORE, utilizing tracked object features across multiple frames. This configuration achieves higher scores than ORE-only across all metrics, highlighting the importance of FMoT in capturing relevant object features for meaningful video captioning.

The last row of Table II shows the inclusion of both FMoT and ORE, achieving the best performance across all metrics for both datasets. This result underscores the complementary nature of the two modules and the efficacy of the proposed information augmentation approach.

To further illustrate the impact of the FMoT and ORE modules, Fig. 4 presents qualitative results based on examples from the MSVD dataset. Each example includes six extracted video frames, the ground truth caption, and captions generated under different module configurations. The following observations can be made:

- When neither FMoT nor ORE is included (w/o FMoT and ORE), the model struggles to correctly identify subjects and objects in the videos. For instance, in the bottom-left video, the object "phone" is misidentified instead of the correct "microwave", and in the bottom-right video, both the subject "baby animal" and the object "baby" are incorrectly identified instead of "person" and "animal", respectively.
- Including only ORE (w/o FMoT) results in improved captions with fewer subject-object errors. For example, in the bottom-right video, subject-object identification improves, but the object identification issue persists in the bottom-left video, where "microwave" remains unrecognized.
- When FMoT is included without ORE (w/o ORE), the generated captions exhibit further improvement, accurately identifying subjects and objects across all examples. However, interaction descriptions remain imprecise in all videos, with verbs such as "dancing" used instead of "walking" or "playing" instead of "washing".
- The inclusion of both FMoT and ORE (Full Model) achieves the most accurate captions, effectively addressing both subject-object identification and interaction description. This result highlights the complementary strengths of FMoT, which captures relevant object features through tracking, and ORE, which enriches contextual understanding by modeling inter-object relationships.

2) **Number of Tracked Objects:** The number of objects tracked by FMoT is a key hyperparameter that influences both computational overhead and video captioning performance. To



(a) MSVD (b) MSRVTT

Fig. 5. Impact of the number of tracked objects on performance (BLEU-4 and CIDEr scores) for MSVD and MSRVTT datasets. Performance trends show that tracking 3 objects yields optimal results for MSVD, while tracking 4 objects is best for MSRVTT. The results highlight the balance required to avoid incomplete relationships with fewer objects and interference from irrelevant objects with more.

investigate the sensitivity of performance to this parameter, ablation experiments were conducted by varying the number of tracked objects from 1 to 8. Fig. 5 presents the BLEU-4 and CIDEr scores for both datasets under these configurations.

The BLEU-4 and CIDEr metrics were specifically chosen for their complementary strengths in evaluating video captioning performance. BLEU-4 measures n-gram overlaps between generated captions and ground truth, emphasizing syntactic accuracy and the structural coherence of captions. CIDEr, on the other hand, prioritizes semantic relevance by incorporating human judgment of the importance of words and phrases. Together, these metrics provide a comprehensive evaluation of caption quality from both syntactic and semantic perspectives.

Consistent performance trends are observed across both metrics and datasets. For MSVD, optimal performance is achieved when tracking 3 objects, while for MSRVTT, which contains more objects and complex interactions, the best results are observed with 4 tracked objects. Inferior performance with fewer tracked objects is attributed to incomplete modeling of object relationships, while tracking more objects degrades performance due to interference from less relevant objects.

Notably, even with a single tracked object, the proposed model demonstrates competitive performance. On MSVD, it outperforms existing models listed in Table I in terms of CIDEr scores, highlighting its superior ability to generate captions aligned with human judgment. On MSRVTT, the model achieves the second-highest CIDEr score, closely following CoCap. These results underscore the effectiveness of the FMoT module in identifying and focusing on the most significant object within a video, enabling accurate caption generation even with minimal computational settings.

3) Configurations of FMoT: To evaluate the effectiveness of the proposed FMoT, designed with the proposed Track Decoder, ablation experiments were conducted to compare its performance against two alternative implementations: a standard encoder-decoder transformer and a Bi-LSTM architecture.

Table III presents the performance results for the three FMoT configurations. In Table III, “Transformer” refers to the FMoT implementation using a conventional encoder-decoder transformer architecture. This configuration consists of a stack of 6-layer encoders followed by another stack of 6-layer decoders, which process the inputs to yield the object features as the FMoT output.

TABLE III
ABLATION STUDIES OF FMOT CONFIGURATIONS ON MSVD AND MSRVTT, COMPARING THE PROPOSED TRACK DECODER WITH TRANSFORMER AND BI-LSTM ARCHITECTURES.

FMoT	Params	MSVD				MSRVTT			
		B4	M	R	C	B4	M	R	C
Transformer	44M	61.3	41.6	79.0	124.1	43.3	30.2	62.8	56.5
Bi-LSTM	29M	61.9	41.9	79.4	124.2	43.5	30.4	62.9	56.7
Track Decoder	3M	**62.1**	**42.5**	**79.8**	**127.2**	**44.6**	**30.5**	**63.6**	**57.7**

“Bi-LSTM” denotes the FMoT implementation based on a standard Bi-LSTM architecture. This design incorporates temporal context from both forward and backward directions. Bi-LSTM models are known for capturing temporal and long-term dependencies due to their sequential structure, whereas transformer encoder-decoder models excel at modeling complex relationships across frames using attention mechanisms. The performance of these two architectures is competitive, with small differences across all metrics for both datasets, as shown in Table III. Notably, the Bi-LSTM-based implementation achieves slightly higher scores compared to the encoder-decoder transformer.

The proposed Track Decoder, which combines the transformer’s attention mechanism with an iterative looping structure akin to LSTM, integrates the strengths of both architectures. This hybrid design achieves significantly higher scores across all metrics for both datasets, as evidenced in Table III. Furthermore, the Track Decoder demonstrates a substantial reduction in parameters compared to the other configurations, highlighting its computational efficiency while delivering superior captioning performance.

4) Feature types on caption generation: The proposed model provides three feature types as visual inputs for the Caption Decoder: frame-level feature representation $\mathcal{F}$, tracked object features $\{O_k^L\}_{k=1}^N$ (abbreviated as O^L), and object relation features $\mathcal{A}_O$. Ablation experiments were conducted to analyze the impact of different combinations of these features on video captioning performance, implemented via Equation (7) by including and excluding O^L and $\mathcal{A}_O$ in concatenation with $\mathcal{F}$. The results, shown in Table IV for both datasets, are summarized below.

The first row in Table IV corresponds to the configuration using only $\mathcal{F}$, yielding the lowest scores across all metrics. This highlights the limitations of relying solely on frame-level features, as they lack the detailed context provided by object-level features. The second row, using $\tilde{V} = [O^L; \mathcal{F}]$,

TABLE IV
ABLATION STUDIES EVALUATING THE IMPACT OF DIFFERENT FEATURE TYPES ON CAPTION GENERATION FOR THE MSVD AND MSRVTT DATASETS. THE CONFIGURATIONS INCLUDE FRAME-LEVEL FEATURE REPRESENTATION ($\mathcal{F}$), TRACKED OBJECT FEATURES (O^L), AND OBJECT RELATION FEATURES ($\mathcal{A}_O$), USED INDIVIDUALLY AND IN COMBINATION.

Input Features	MSVD				MSRVTT			
	B4	M	R	C	B4	M	R	C
$[\mathcal{F}]$	60.5	41.6	79.2	122.8	42.1	29.8	59.7	54.9
$[O^L; \mathcal{F}]$	61.3	42.0	79.5	125.8	44.0	30.4	63.5	57.4
$[\mathcal{A}_O; \mathcal{F}]$	**62.1**	**42.5**	**79.8**	**127.2**	**44.6**	**30.5**	**63.6**	**57.7**
$[\mathcal{A}_O; O^L; \mathcal{F}]$	60.8	**42.5**	79.6	125.8	43.7	30.2	63.2	57.6

shows improved performance, emphasizing the importance of tracked object features in enhancing captioning accuracy by capturing temporal coherence and key object information. The third row with $\tilde{V} = [\mathcal{A}_O; \mathcal{F}]$ demonstrates the highest performance across all metrics for both datasets, underscoring the informativeness of object relation features in capturing inter-object relationships crucial for generating meaningful captions.

Interestingly, the configuration combining all three feature types ($\tilde{V} = [\mathcal{A}_O; O^L; \mathcal{F}]$), as shown in the last row of Table IV, yields inferior performance compared to $\tilde{V} = [\mathcal{A}_O; \mathcal{F}]$. It performs similarly to $\tilde{V} = [O^L; \mathcal{F}]$, with the same CIDEr score for MSVD and a higher CIDEr score for MSRVTT. The increased model complexity from including all feature types does not result in better performance, likely due to feature overlap between $\mathcal{F}$ and O^L. This overlap may cause the model to lose appropriate focus on $\mathcal{A}_O$, the most distinct and informative feature type for caption generation. These findings highlight the critical importance of balancing feature diversity and avoiding redundancy to optimize model performance.

E. Qualitative Analysis

To better illustrate the performance advantages of the proposed Track4Cap model, qualitative comparisons of the generated captions are provided against two other transformer-based models, using video examples from (a) the MSVD dataset and (b) the MSRVTT dataset, as shown in Fig. 6. The baseline model corresponds to Track4Cap with FMoT implemented as a standard encoder-decoder transformer, while SwinBERT [1] is included as a reference for being the earliest end-to-end transformer-based video captioning model. The following advantages of the proposed Track4Cap model emerge from these examples:

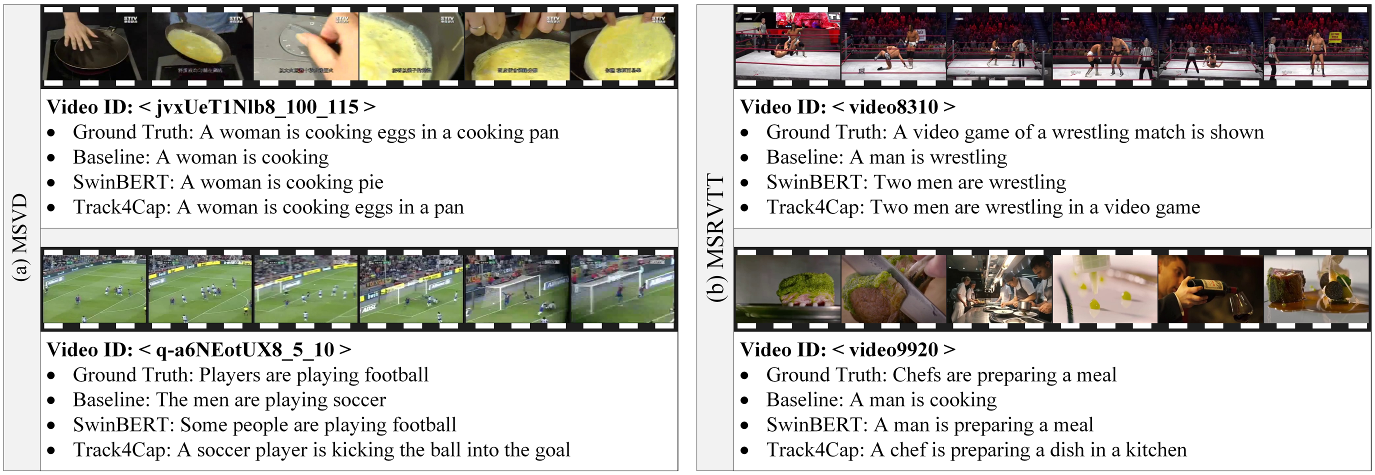
- *More accurate subject and object descriptions:* Tracked object features from FMoT based on the track decoder enable Track4Cap to generate captions with precise subject and object details. For instance, SwinBERT misidentifies "eggs" as "pie" in the first video of Fig. 6(a), and the baseline model inaccurately describes "two men" as "a man" in the first video of Fig. 6(b). These results highlight the effectiveness of the proposed track decoder in improving object recognition.
- *Improved caption quality:* The integration of tracked object features from FMoT and object relation features from ORE allows Track4Cap to generate longer and more detailed sentences compared to the short and simple captions produced by the baseline model and SwinBERT. For example, in the first video of Fig. 6(a), Track4Cap describes interactions among three entities ("A woman is cooking eggs in a pan"), whereas the baseline model provides only a basic description ("A woman is cooking"). Similarly, for the second video in Fig. 6(a), Track4Cap highlights the key action of scoring in soccer, surpassing the ground truth and other models that only describe playing football. These examples demonstrate the ability of the proposed track decoder and ORE to enrich caption quality.
- *Enhanced scene context awareness:* Track4Cap incorporates scene context into captions, such as mentioning "video game" and "kitchen" in the captions for the two videos in Fig. 6(b). These details are absent in the captions produced by the baseline model and SwinBERT, further showcasing the contributions of the proposed Track4Cap framework to detailed and contextually aware captioning.

V. CONCLUSION

This paper presents Track4Cap, a novel framework for video captioning designed to achieve an optimal balance between computational efficiency and high performance. By leveraging frame-level features and incorporating two computationally efficient modules—Frame-by-Frame Multi-object Tracking (FMoT) and Object Relation Encoder (ORE)—the proposed framework identifies salient object features across frames and captures inter-object relationships to enhance captioning performance. Track4Cap avoids the need for complex architectures, additional data modalities, or end-to-end training, making it practical for real-world applications with constrained computational resources.

Extensive ablation studies highlight the contributions of the proposed modules and the impact of different types of augmented information on performance improvement. Results demonstrate that Track4Cap achieves significant performance gains on the MSVD dataset, excelling in scenarios involving single-activity videos with simpler activity structures. On the more complex MSRVTT dataset, Track4Cap remains highly competitive despite the inherent challenges posed by multi-activity videos and intricate object relationships. These findings emphasize the robustness and adaptability of Track4Cap across datasets with varying levels of complexity.

The implications of this research extend to several practical domains, including video indexing, automated content generation, and accessibility solutions, where both computational cost and caption quality are critical. Future studies could explore ways to further improve Track4Cap's performance on multi-activity videos, such as integrating advanced relational modeling techniques or utilizing external knowledge bases to enhance context understanding. Additionally, the modular design of Track4Cap provides opportunities for adaptation to related tasks, such as video summarization and event detection, enabling broader applications in video understanding. These



Video ID: < jvxUeT1Nlb8_100_115 >
- Ground Truth: A woman is cooking eggs in a cooking pan
- Baseline: A woman is cooking
- SwinBERT: A woman is cooking pie
- Track4Cap: A woman is cooking eggs in a pan

Video ID: < q-a6NEotUX8_5_10 >
- Ground Truth: Players are playing football
- Baseline: The men are playing soccer
- SwinBERT: Some people are playing football
- Track4Cap: A soccer player is kicking the ball into the goal

Video ID: < video8310 >
- Ground Truth: A video game of a wrestling match is shown
- Baseline: A man is wrestling
- SwinBERT: Two men are wrestling
- Track4Cap: Two men are wrestling in a video game

Video ID: < video9920 >
- Ground Truth: Chefs are preparing a meal
- Baseline: A man is cooking
- SwinBERT: A man is preparing a meal
- Track4Cap: A chef is preparing a dish in a kitchen

Fig. 6. Qualitative comparisons of generated captions for video examples from (a) the MSVD dataset and (b) the MSRVTT dataset. Each example includes selected video frames, the corresponding ground truth caption, and captions generated by three models: the baseline model (Track4Cap with FMoT implemented as a standard encoder-decoder transformer), SwinBERT, and the proposed Track4Cap model. These comparisons demonstrate the superior ability of Track4Cap to accurately identify subjects and objects, describe interactions in detail, and produce contextually rich captions.

directions pave the way for advancing efficient and effective video captioning solutions.

References

[1] K. Lin, L. Li, C.-C. Lin, F. Ahmed, Z. Gan, Z. Liu, Y. Lu, and L. Wang, “SWINBERT: End-to-End Transformers with Sparse Attention for Video Captioning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17949–17958, 2022.

[2] Z. Yue, Q. Zhang, A. Hu, L. Zhang, Z. Wang, and Q. Jin, “Movie101: A New Movie Understanding Benchmark,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4669–4684, 2023.

[3] K. Yamazaki, S. Truong, K. Vo, M. Kidd, C. Rainwater, K. Luu, and N. Le, “VLCAP: Vision-Language with Contrastive Learning for Coherent Video Paragraph Captioning,” in *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 3656–3661, 2022.

[4] S. Chen, H. Li, Q. Wang, Z. Zhao, M. Sun, X. Zhu, and J. Liu, “Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[5] Y. Wang, K. Li, X. Li, J. Yu, Y. He, C. Wang, G. Chen, B. Pei, R. Zheng, J. Xu, Z. Wang, *et al.*, “Internvideo2: Scaling video foundation models for multimodal video understanding,” *arXiv preprint arXiv:2403.15377*, 2024.

[6] J. Li, D. Li, S. Savarese, and S. Hoi, “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models,” in *International conference on machine learning*, pp. 19730–19742, PMLR, 2023.

[7] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.

[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks,” in *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[10] 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*, pp. 6299–6308, 2017.

[11] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *Advances in neural information processing systems*, vol. 28, 2015.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[14] B. Wu, B. Liu, P. Huang, J. Bao, P. Xi, and J. Yu, “Concept Parser With Multimodal Graph Learning for Video Captioning,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 9, pp. 4484–4495, 2023.

[15] S. Yan, T. Zhu, Z. Wang, Y. Cao, M. Zhang, S. Ghosh, Y. Wu, and J. Yu, “VideoCoCa: Video-Text Modeling with Zero-Shot Transfer from Contrastive Captioners,” *arXiv preprint arXiv:2212.04979*, 2022.

[16] M. Tang, Z. Wang, Z. Zeng, X. Li, and L. Zhou, “Stay in grid: Improving video captioning via fully grid-level representation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 7, pp. 3319–3332, 2023.

[17] 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.

[18] H. Xu, Q. Ye, M. Yan, Y. Shi, J. Ye, Y. Xu, C. Li, B. Bi, Q. Qian, W. Wang, G. Xu, J. Zhang, S. Huang, F. Huang, and J. Zhou, “mPLUG: Effective and Efficient Vision-Language Learning by Cross-modal Skip-connections,” *arXiv preprint arXiv:2205.12005*, 2022.

[19] D. Ko, J. Choi, H. K. Choi, K.-W. On, B. Roh, and H. J. Kim, “MELTR: Meta Loss Transformer for Learning to Fine-tune Video Foundation Models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20105–20115, 2023.

[20] Y. Shen, X. Gu, K. Xu, H. Fan, L. Wen, and L. Zhang, “Accurate and Fast Compressed Video Captioning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15558–15567, 2023.

[21] S. Hochreiter and J. Schmidhuber, “LONG SHORT-TERM MEMORY,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] H. Ryu, S. Kang, H. Kang, and C. D. Yoo, “Semantic Grouping Network for Video Captioning,” in *proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 2514–2522, 2021.

[23] S. Chen and Y.-G. Jiang, “Motion Guided Region Message Passing for Video Captioning,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1543–1552, 2021.

[24] X. Zhong, Z. Li, S. Chen, K. Jiang, C. Chen, and M. Ye, “Refned Semantic Enhancement towards Frequency Diffusion for Video Captioning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 3724–3732, 2023.

[25] W. Xu, Z. Miao, J. Yu, Y. Tian, L. Wan, and Q. Ji, “Bridging video and text: A two-step polishing transformer for video captioning,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6293–6307, 2022.

[26] L. Gao, Y. Lei, P. Zeng, J. Song, M. Wang, and H. T. Shen, “Hierarchical Representation Network With Auxiliary Tasks for Video Captioning and Video Question Answering,” *IEEE Transactions on Image Processing*, vol. 31, pp. 202–215, 2021.

[27] Z. Zhang, Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu, and Z.-J. Zha, "Object Relational Graph with Teacher-Recommended Learning for Video Captioning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13278–13288, 2020.

[28] B. Pan, H. Cai, D.-A. Huang, K.-H. Lee, A. Gaidon, E. Adeli, and J. C. Niebles, "Spatio-Temporal Graph for Video Captioning with Knowledge Distillation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10870–10879, 2020.

[29] B. Wu, G. Niu, J. Yu, X. Xiao, J. Zhang, and H. Wu, "Towards Knowledge-Aware Video Captioning via Transitive Visual Relationship Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 10, pp. 6753–6765, 2022.

[30] L. Li, X. Gao, J. Deng, Y. Tu, Z.-J. Zha, and Q. Huang, "Long Short-Term Relation Transformer With Global Gating for Video Captioning," *IEEE Transactions on Image Processing*, vol. 31, pp. 2726–2738, 2022.

[31] Q. Zheng, C. Wang, and D. Tao, "Syntax-Aware Action Targeting for Video Captioning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13096–13105, 2020.

[32] Y. Tu, C. Zhou, J. Guo, S. Gao, and Z. Yu, "Enhancing the Alignment between Target Words and Corresponding Frames for Video Captioning," *Pattern Recognition*, vol. 111, p. 107702, 2021.

[33] G. Li, H. Ye, Y. Qi, S. Wang, L. Qing, Q. Huang, and M.-H. Yang, "Learning Hierarchical Modular Networks for Video Captioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 2, pp. 1049–1064, 2024.

[34] S. Liu, A. Li, Y. Zhao, J. Wang, and Y. Wang, "Evcap: Element-aware video captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 10, pp. 9718–9731, 2024.

[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*, pp. 3202–3211, 2022.

[36] Z. Ghaderi, L. Salewski, and H. P. Lensch, "Diverse Video Captioning by Adaptive Spatio-temporal Attention," in *DAGM German Conference on Pattern Recognition*, pp. 409–425, Springer, 2022.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[38] H. Luo, L. Ji, B. Shi, H. Huang, N. Duan, T. Li, J. Li, T. Bharti, and M. Zhou, "UniVL: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation," *arXiv preprint arXiv:2002.06353*, 2020.

[39] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE," *ICLR*, 2021.

[41] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, and T. Li, "CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval," *Neurocomputing*, vol. 508, pp. 293–304, 2022.

[42] P. Jin, J. Huang, F. Liu, X. Wu, S. Ge, G. Song, D. Clifton, and J. Chen, "Expectation-Maximization Contrastive Learning for Compact Video-and-Language Representations," *Advances in neural information processing systems*, vol. 35, pp. 30291–30306, 2022.

[43] T. Han, Y. Xu, J. Yu, Z. Yu, and S. Zhao, "Action-driven semantic representation and aggregation for video captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–13, 2024.

[44] D. Chen and W. B. Dolan, "Collecting Highly Parallel Data for Paraphrase Evaluation," in *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pp. 190–200, 2011.

[45] J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A Large Video Description Dataset for Bridging Video and Language," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5288–5296, 2016.

[46] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[47] S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

[48] C.-Y. Lin and F. J. Och, "Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics," in *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*, pp. 605–612, 2004.

[49] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "CIDEr: Consensus-based Image Description Evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

[50] D. Kingma and J. Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION," in *3rd International Conference on Learning Representations*, (San Diega, CA, USA), 2015.

HuiLan Luo earned her Ph.D. in Computer Science and Technology from Zhejiang University in 2008 and served as a Postdoctoral Fellow at Fudan University until 2011. She is currently a Professor and Discipline Leader at the School of Information Engineering, Jiangxi University of Science and Technology, and Director of the Jiangxi Province Key Laboratory of Multidimensional Intelligent Perception and Control. Her research focuses on image recognition, computer vision, and machine learning. Prof. Luo has secured four grants from the National Natural Science Foundation of China and is recognized as one of the Leading Talent Training Targets for Major Discipline Technology Leaders in Jiangxi Province.

Xia Cai received the B.S. degree in Information Security from Jiangxi University of Science and Technology, Jiangxi, China, in 2022, and entered Jiangxi University of Science and Technology as a master's student with a major in computer technology in 2022. Her research interests include video understanding and video captioning.

Lik-Kwan Shark is a Fellow of the IET and Emeritus Professor at the University of Central Lancashire, UK. He founded and led two research centers: the Applied Digital Signal and Image Processing Research Centre (ADSIP) established in 2001, and the Advanced Digital Manufacturing Technology Research Centre (ADMT) established in 2008. He has presided as the general chair of three international conferences (AECRIS, ICIGP and IFSP), and serves on the advisory board of IET Image Processing among several editorial board roles.