Language-Guided Graph Representation Learning for Video Summarization

Wenrui Li, Wei Han, Hengyu Man, Wangmeng Zuo, *Senior Member, IEEE*
Xiaopeng Fan, *Senior Member, IEEE* Yonghong Tian, *Fellow, IEEE*

Abstract—With the rapid growth of video content on social media, video summarization has become a crucial task in multimedia processing. However, existing methods face challenges in capturing global dependencies in video content and accommodating multimodal user customization. Moreover, temporal proximity between video frames does not always correspond to semantic proximity. To tackle these challenges, we propose a novel **L**anguage-guided **G**raph **R**epresentation **L**earning **N**etwork (**LGRLN**) for video summarization. Specifically, we introduce a video graph generator that converts video frames into a structured graph to preserve temporal order and contextual dependencies. By constructing forward, backward and undirected graphs, the video graph generator effectively preserves the sequentiality and contextual relationships of video content. We designed an intra-graph relational reasoning module with a dual-threshold graph convolution mechanism, which distinguishes semantically relevant frames from irrelevant ones between nodes. Additionally, our proposed language-guided cross-modal embedding module generates video summaries with specific textual descriptions. We model the summary generation output as a mixture of Bernoulli distribution and solve it with the EM algorithm. Experimental results show that our method outperforms existing approaches across multiple benchmarks. Moreover, we proposed LGRLN reduces inference time and model parameters by 87.8% and 91.7%, respectively. Our codes and pre-trained models are available at https://github.com/liwrui/LGRLN.

Index Terms—Graph representation learning, video summarization, query suggestion.

1 Introduction

Video contents processing is a crucial field in multimedia analysis [1], [2], [3], [4], [5], [6], [7], [8]. The video summarization task aims to create concise representations of video content by selecting key frames or segments that capture the main information. This task has gained importance due to the massive daily growth in video data, especially on social media platforms [9], [10], [11], [12]. The rise of platforms like YouTube, TikTok, and Instagram has led to an explosion in short video content. As of 2023, about 900 hours of video are uploaded to YouTube every minute, amounting to approximately 1.2 million hours of content daily[1]. YouTube Shorts alone receive around 70 billion daily views, highlighting the significant consumption and production of short-form video content. These staggering numbers underscore the necessity for effective video summarization techniques to efficiently manage and navigate this vast amount of data [13], [14], [15], [16], [17], [18]. Therefore, the numerous video summarization methods have been proposed in recent years [19] [20] [21] [22] [23] [24] [25] [26] [27] [28]. Video summarization can be broadly categorized into keyframe-based summarization and key shot-based summarization. The former selects important frames to create a static storyboard, while the latter segments the video into shots and selects informative ones to generate dynamic video skimming. In this paper, we focus on generating accurate and efficient video skimming, considering practical application scenarios [29] [30].

In this work, we focus on video summarization, where a user's natural-language query guides the selection of a compact set of key frames or shots to create a temporally coherent visual summary of the video. In contrast, video captioning aims to generate free-form textual descriptions for a given video and has seen rapid progress. Li et al. [31] propose a hierarchical modular network that aligns video and language semantics across entities, verbs, predicates, and sentences, enhanced by scene-graph–based learning, to advance video captioning beyond word-level supervision. Ma et al. [32] introduce a style-aware, two-stage framework that incorporates caption styles and dynamically encodes video-style cues, enabling more precise and diverse video captions. Tian et al. [33] propose a retrieval-inspired video captioning framework featuring a learnable token-shift module for fine-grained temporal modeling and a Refine-former for cross-attentive patch–caption integration. Ma et al. [34] present a zero-shot video captioning framework that leverages learnable tokens to bridge frozen vision–language models with GPT-2 for test-time adaptation, enabling video-aware captioning without training data and yielding significant performance gains. Although related, the two tasks differ in output modality, objective, and evaluation criteria. We focus on summarization because it produces interpretable, efficient, and controllable video skims. This motivates our

- *Wenrui Li, Wei Han, Hengyu Man, Wangmeng Zuo and Xiaopeng Fan are with the Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China, and also with Harbin Institute of Technology Suzhou Research Institute, Suzhou 215104, China.*
  E-mail: liwr@stu.hit.edu.cn; 2021111641@stu.hit.edu.cn; manhengyu@hotmail.com; wmzuo@hit.edu.cn; fxp@hit.edu.cn
- *Yonghong Tian is with the School of AI for Science, the Shenzhen Graduate School, Peking University, Shenzhen, China, the Peng Cheng Laboratory, Shenzhen, China and also with the School of Computer Science, Peking University, Beijing, China.*
  E-mail: yhtian@pku.edu.cn

Corresponding author: Xiaopeng Fan

1. https://thumbnailtest.com/stats/youtube/

language-guided graph representation learning approach, which explicitly captures long-range temporal and semantic relationships while maintaining a lightweight design.

Existing video summarization methods aim to enhance the model's ability to capture global dependencies, integrate multimodal information, and detect saliency. Due to the informative and complex content of unconstrained videos (e.g., home videos or surveillance footage), Sun et al. [20] generated montage summaries representing the video's main content by detecting salient segments. DSNet [25] employs a dual-stream network architecture that simultaneously processes content and style information, creating more comprehensive summaries. However, traditional sequential models struggle to capture long-range dependencies in videos, which makes it difficult to produce high-quality summaries. Zhao et al. [26] addressed this by modeling videos as graph structures and using graph convolutional networks to capture global dependencies between shots, resulting in more representative summaries. Despite these advancements, existing methods lack the flexibility to customize summaries based on user requirements. Effectively leveraging various modalities (e.g., visual, audio, and textual) is crucial for improving the quality of summaries. CLIP-It [24] introduced a multimodal transformer guided by language to generate video summaries, allowing users to customize summaries through natural language queries and enhancing both flexibility and accuracy. QMS [23] proposed a query-based micro-video summarization method that automatically generates short video summaries relevant to user queries. VideoXum [27] employs a multimodal transformer to combine different modalities of video information. Different from previous graph methods, Hong et al. [35] model language–visual entity relationships for navigation decisions, and Shabani et al. [36] sparsify traffic graphs for forecasting. In contrast, our method represents each video frame as a node, builds forward, backward, and undirected temporal graphs, integrates language tokens through explicit edges to video nodes, and applies bi-threshold relational reasoning with a mixture-of-Bernoulli EM objective, enabling efficient and interpretable video summaries.

A practical scenario that motivates our design is query-guided video summarization on resource-constrained devices (e.g., smartphones), where computation budget and latency strongly affect user experience. In such settings, the key technical challenge is to reduce model cost while still capturing global temporal and semantic associations and supporting language-guided personalization. Our work moves in this direction by proposing a lightweight, graph-based approach. The main challenge in video summarization is to efficiently capture global temporal relationships and semantic associations in resource-constrained environments, ensuring summary completeness and consistency. Additionally, the model must offer a high level of customization to meet diverse user needs. Unlike traditional models generating generic summaries, this model is expected to create personalized summaries tailored to individual user descriptions. It should also generalize effectively across diverse video scenarios and user inputs. Consequently, to deliver an efficient and accurate solution, the model must ensure high computational efficiency while capturing comprehensive and consistent global temporal and semantic associations in resource-limited settings. Furthermore, it should adapt flexibly to specific user needs, generating highly personalized outputs. These requirements are closely interconnected: computational efficiency ensures feasibility on resource-limited devices, capturing global associations is key to high-quality summaries, and personalization is vital for addressing user-specific needs.

In general, as illustrated at the top of Fig 1, conventional methods like Transformer-based single-modal summarization face challenges like high computational costs, inconsistent semantic and temporal relevance, and limited generalization in summaries. These limitations become especially evident in scenarios requiring efficient and meaningful video summarization across diverse, large-scale datasets. To address these challenges, we propose a GNN-based multimodal summarization framework. This method is distinguished by its computational efficiency, the decoupling of semantic and temporal relevance, and its practical user customization capabilities. By constructing a query-oriented graph that integrates multiple modalities, our model generates summaries that not only capture the intrinsic nature of video content but also align closely with specific user-defined queries, such as isolating scenes containing objects like "dogs" and "birds". Our method ensures that the generated summaries are semantically informative and temporally coherent, providing significant improvements over traditional methods.

Moreover, as shown in the comparative results in the lower part of Fig. 1, our method models the summary generation output as a mixture of Bernoulli distribution and solve it with the EM algorithm, outperforms traditional approaches in metrics like cosine similarity, log Hamming distance between ground truths, and in handling the one-to-many mapping challenge. The integration of these features into our model allows it to produce highly relevant, customizable video summaries while maintaining low computational cost.

In this paper, we propose a novel **L**anguage-guided **G**raph **R**epresentation **L**earning **N**etwork (LGRLN) for video summarization. Traditional video summarization methods often struggle to accurately capture the intricate temporal and semantic relationships within video content, resulting in suboptimal outcomes. To address this challenge, our approach starts with a video graph generator that transforms video frames into a structured graph, effectively preserving both temporal order and contextual dependencies. This graph-based representation provides a solid foundation for more sophisticated summarization by maintaining the sequential nature of video content while also considering the broader context of each frame. The intra-graph relational reasoning module enhances this by employing a bi-threshold graph convolution mechanism to intelligently filter and aggregate relevant information from neighboring frames. This module effectively reduces noise by distinguishing between semantically relevant and irrelevant frames, thereby improving the precision of the generated summaries. The language-guided cross-modal embedding module represents a significant advancement in video summarization. This module integrates natural language inputs, enabling users to generate video summaries tailored to specific textual descriptions. This cross-modal capability

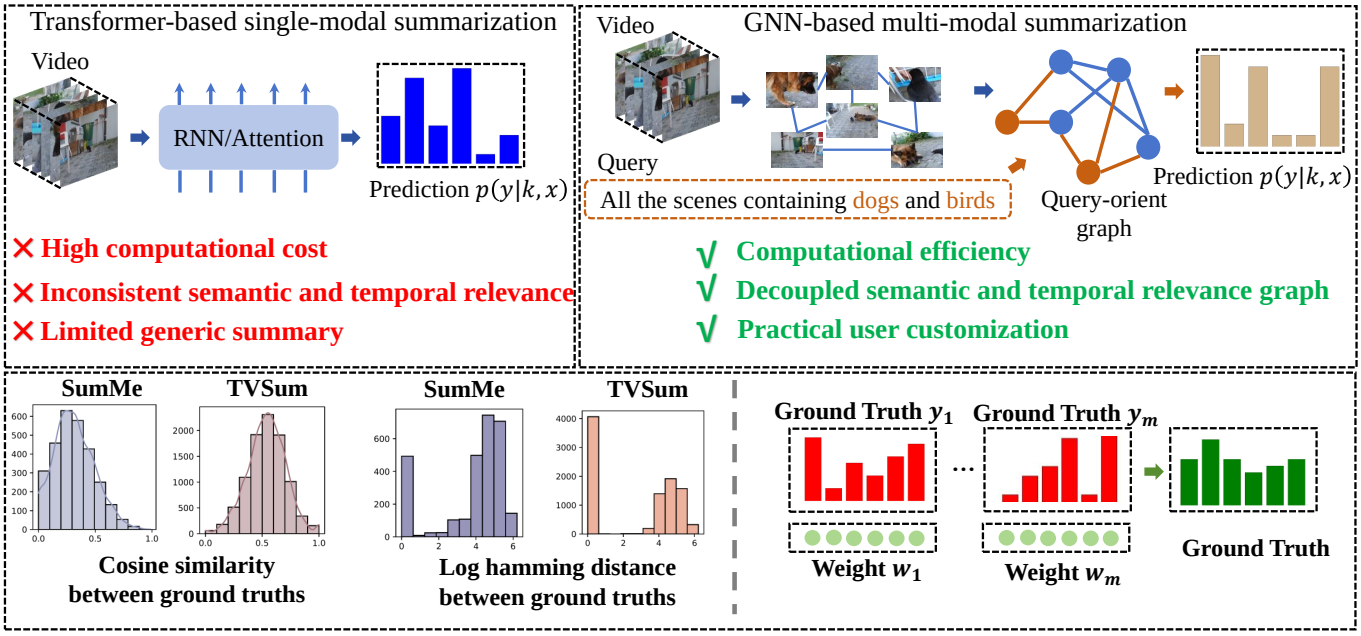

Fig. 1. The motivation of this paper. First, compared to RNN or attention-based methods, graph neural networks (GNNs) more naturally integrate multimodal information using heterogeneous graphs, while also reducing computational complexity as edge sparsity increases. Second, GNNs represent videos as graphs and conduct video summarization through node classification, providing strong interpretability. Finally, given that a video in the dataset may have multiple labels, directly averaging these labels could result in information loss, making it crucial to employ appropriate methods for processing distinct labels. To address label discrepancies in the TVSum and SumMe datasets, where cosine similarity and Hamming distance may vary between annotators, our method applies weighted averaging within an EM algorithm framework, improving label consistency and relevance in the generated summaries.

not only enhances the adaptability of the summarization process but also offers a personalized experience by aligning video content with user-defined contexts. By integrating these three components, our architecture not only improves the accuracy and relevance of video summaries but also sets a new standard in cross-modal summarization, dynamically aligning video content with multi-modal inputs and significantly enriching the summarization process. On the SumMe dataset, LGRLN achieves an F1 score of 54.7, which represents an improvement of approximately 18.8% over VideoSAGE. Similarly, in the TVSum dataset, it reaches an F1 score of 58.3, representing a 13.2% improvement over SumGraph. Despite achieving high performance, LGRLN maintains efficiency with only 3 million parameters, which is a dramatic reduction of more than 99% compared to models like V2Xum-LLaMA-7B, which has 7 billion parameters, and V2Xum-LLaMA-13B, which has 13 billion parameters. The main contributions could be summarized as follows:

- This paper introduces a novel Language-Guided Representation Learning Network (LGRLN) for video summarization. This framework directly integrates natural language input into the video summarization process, allowing users to generate summaries that closely align with specific textual queries.
- We propose an intra-graph relational reasoning module based on a dual-threshold graph convolution mechanism. This method effectively reduces noise by filtering and aggregating relevant information from neighboring frames, while dynamically updating the features of nodes and edges to address variations in significance across video frames.
- We introduce a video graph generator that converts video frames into structured graphs, preserving the temporal order and contextual dependencies of the content. To tackle the one-to-many mapping issue in video summarization, we model the summary scheme output as a mixture of Bernoulli distribution and solve it using the EM algorithm.

The rest of the paper is organized as follows: Section 2 provides a comprehensive background on Abstractive and query-oriented video summarization. Section 3 offers a detailed description of the proposed LGRLN architectures. Section 4 presents experimental results and visualizations, demonstrating the efficacy of our model. Finally, Section 5 summarizes the key findings and contributions of the study.

2 Related Work

2.1 Abstractive Video Summarization

Supervised video summarization methods rely on manually annotated summaries as references. The straightforward approach involves designing scoring models to evaluate the summary score of each frame or shot, selecting higher-scoring ones for the summary. Gygli et al. [37] proposed a new method and benchmark for video summarization, evaluating each frame's visual interest through "superframe" segmentation and selecting the best frames to create an informative and engaging summary. Gong et al. [38] introduced a supervised subset selection method using the seqDPP model, which learns to select informative and diverse sets of frames from human-created summaries, achieving high-quality video summaries. Extracting robust and efficient temporal information is crucial for enhancing video

summarization performance. Zhao et al. [39] proposed the TTH-RNN model for video summarization, addressing traditional RNNs' difficulties in handling high-dimensional video features and long sequences through a tensor-train embedding layer and a hierarchical RNN structure. Zhang et al. [40] introduced an innovative sequence-to-sequence learning model, adding a "review encoder" to embed the predicted summary and the original video into the same semantic space, comparing their distances to preserve the video's semantic information. Recent studies have shown the limitations of sequential models in video modeling, leading to the introduction of attention mechanisms or graph convolutional networks to consider global dependencies. Chen et al. [41] enhanced CNNs' ability to capture global relations by aggregating features globally, projecting them into an interaction space for efficient relation reasoning, and then mapping the relation-aware features back to the original coordinate space. Zeng et al. [42] constructed a graph of action units and their relationships, using a graph convolutional network (GCN) to capture the temporal and semantic relationships between action units, enhancing temporal action localization in videos. VideoSAGE [43] constructs a sparse graph to capture long-range frame relationships and formulates the summarization task as a binary node classification problem. Mrigank et al. [44] proposed a fully convolutional sequence model for video summarization, treating video summarization as a sequence labeling problem and using methods from semantic segmentation to select important frames for summary videos.

Unsupervised video summarization methods focus on designing learning-based criteria, transforming the task into a subset selection problem. Early approaches leveraged clustering algorithms to group frames or shots into clusters, defining the center of each cluster as key-frames or key-shots. Lee et al. [45] utilized region cues like hand proximity and gaze to predict importance, proving effective without being specific to any user or object. Mundur et al. [46] presented a fully automatic video summarization technique using Delaunay Triangulation for clustering frames, producing high-quality summaries with fewer frames and less redundancy than other methods. To address the insufficient capture of fine-grained contextual scene interactions and motion information by clustering algorithms, dictionary learning converts the problem into a sparse optimization problem. Elhamifar et al. [47] introduced a method to find representative data points that describe an entire dataset through a sparse multiple measurement vector problem, selecting representatives via convex optimization without assuming low-rank or cluster-centered data. SumGraph [48] refines frame-level semantic relationships via graph convolutions, enabling effective keyframe selection in both supervised and unsupervised settings. RSGN [26] integrates LSTM for intra-shot frame dependencies and GCN for inter-shot relationships, enabling unsupervised learning through reconstruction loss to capture both local and global video context effectively. DSAVS [49] selects semantically rich summaries by aligning video and text representations, enhanced with self-attention to capture long-range temporal dependencies.

However, traditional video summarization methods struggle with long-duration videos due to weak global dependencies, inadequate modeling of local-global relationships, and the diversity of video summaries. Our method leverages a video graph generator to transform video frames into structured graphs, thereby preserving temporal order and contextual dependencies more effectively. Our dual-threshold graph convolution mechanism effectively reduces noise, dynamically updates node and edge features, and ensures the proper modeling of frame importance. Additionally, we employ a hybrid Bernoulli distribution to model the summarization process and optimize it using the EM algorithm, effectively addressing the one-to-many mapping problem in video summarization.

2.2 Query-oriented Video Summarization

Current video summarization methods generating generic summaries are inadequate for meeting the personalized needs of different users. Consequently, generating summaries focusing on specific parts based on users' natural language descriptions has become a crucial challenge. Sharghi et al. [50] proposed a probabilistic model to select key shots from long videos based on their relevance. Sharghi et al. [51] presented a memory network-based model for query-focused summaries, addressing user subjectivity and evaluation challenges by emphasizing semantic information over visual features. Kanehira et al. [52] introduced viewpoint-specific video summarization, leveraging semantic similarities and differences among video groups to optimize diversity, representativeness, and discriminativeness. Jia et al. [23] introduced a query-oriented micro-video summarization model employing an encoder-decoder transformer to handle diverse entities and complex scenes. Narasimhan et al. [24] proposed a multimodal transformer framework for both generic and query-focused video summarization, utilizing language models to guide importance scoring.

Traditional query-focused video summarization methods struggle to accurately capture query relevance and lack explicit structured modeling. Our approach incorporates a language-guided cross-modal embedding module, seamlessly integrating natural language inputs to generate video summaries closely aligned with specific textual queries. Utilizing a GNN-based framework, our method effectively fuses multimodal information, ensuring high semantic consistency between generated summaries and user queries.

2.3 Large Language Models in Video Understanding

In recent years, video understanding methods based on Large Language Models (LLMs) have demonstrated outstanding performance in video signal processing, thanks to their powerful reasoning capabilities. For example, Videopoet [53] can synthesize high-quality videos from multiple conditional signals and match them with corresponding audio, showcasing the potential of LLMs in video generation. Unlike traditional methods that encode images and videos into different feature spaces and input them separately into LLMs, Video-LLaVA [54] innovatively unifies visual representations into a single language feature space, fundamentally addressing the issue of projection mismatch caused by the lack of a unified tokenization between images and videos. This significantly enhances the efficiency of multimodal understanding. VideoLLM [55], inspired by the

exceptional causal reasoning capabilities of LLMs, transfers these reasoning abilities to video sequence understanding tasks. By introducing the Modality Encoder and Semantic Translator, it converts video frames and other multimodal inputs (e.g., text) into a unified token sequence, overcoming the limitations of task specialization in existing video understanding models and enabling more flexible multitask handling. Furthermore, Videochat [56] integrates LLMs with video foundation models via a learnable neural interface, empowering the system with powerful video content analysis and dialogue capabilities, driving video understanding towards more interactive and human-like experiences. Finsta [57] introduces an innovative fine-grained spatiotemporal alignment approach that leverages scene graph modeling and alignment techniques, enhancing the cross-modal representation ability of video language models and improving the model's understanding of complex video content.

In the field of video summarization, LLM-based methods have also garnered widespread attention [58], [59], [60], [61]. V2Xum-LLM [62] proposed an innovative framework that unifies various video summarization tasks within the LLM text decoder, significantly improving performance and flexibility by leveraging multimodal support and unified task modeling. M3Sum [63] simplifies the cross-modal alignment issue with a parameter-free alignment mechanism and leverages LLMs' strength in long-text understanding to effectively enhance the quality of video summarization.

Despite significant performance breakthroughs achieved by LLM-based video understanding methods, they still face high training costs and inference times, which pose a major challenge to their deployment in resource-constrained environments. As a result, developing high-performance, low-power video summarization models has become a key research direction. This study presents an efficient and lightweight LLM-driven video summarization method that reduces computational resource consumption while maintaining inference effectiveness, offering a new solution for video processing in real-world applications. Experiments on standard benchmarks validate the proposed method's efficiency and effectiveness.

3 The Proposed Approach

The proposed architecture consists of four primary components: a video graph generator, a intra-graph relational reasoning module, a language-guided cross-modal embedding module, and a bi-threshold cross-entropy loss function. The video graph generator models the input video as three types of graphs: forward, backward, and undirected, based on the time intervals between frames and the direction of the video stream, to capture information from different directions of the video stream. The intra-graph relational reasoning module calculates the importance score for each node and designs aggregated weights combined with variational inference, combining prior knowledge to reduce model sensitivity and enhance robustness. The language-guided cross-modal embedding module considers both natural language sequences and video sequences as interconnected graphs, integrating external natural language instructions into the video sequence to generate personalized summaries. This information fusion method based on heterogeneous graphs can explicitly design information aggregation patterns through edge construction rules, thereby improving model interpretability and controllability. The bi-threshold loss function adapts to varying human annotation preferences. It models the output as a mixed Bernoulli distribution, aligning the model's output with specific annotated preferences rather than all possible annotations, thereby reducing the loss of label information. Tab. 1 demonstrated the notations and descriptions in detail.

TABLE 1
Key Notations and Descriptions

Notation	Description
$G_f(V, E_f)$	Forward video graph (nodes V, edges E_f)
$G_b(V, E_b)$	Backward video graph (edges E_b)
$G_u(V, E_u)$	Undirected video graph (edges E_u)
$\boldsymbol{x}_i \in \mathbb{R}^{D_x}$	Frame i feature vector (D_x: dimension)
$\mathcal{G}(V, E)$	General graph (nodes V, edges E)
$\boldsymbol{h}_i^{(t)}$	Node i feature at iteration t
$\boldsymbol{\alpha}_i^j$	Aggregation weight (cosine similarity)
$\boldsymbol{\tau}_e$	Time embedding for temporal order
L_{BCE}	Bi-threshold cross-entropy loss
$p(\boldsymbol{y}_i = 1 \mid \boldsymbol{z}_i)$	Probability node i is a keyframe

3.1 Task Description

The video summarization task is to find a mapping $f : \mathbb{R}^{|V|} \rightarrow \mathbb{R}^{|S|}$, taking the video sequence V as input and subset S of V as output, and ensure that the subset S can contain the main semantic information of V. In the model-based approach, video V will be firstly extracted frame by frame to obtain a feature sequence $\{x_i\}_{i \in V}$, and then a neural network is trained on the dataset to fit the mapping f. In the training set, each video sequence has k ground-truth summaries $\{y_i\}_{0 \leq i < k}$ generated by different annotators. Each annotation sequence y is a binary sequence with a value of 0 or 1, where whether the j-th element is 1 represents whether the j-th frame belongs to the digest subset S. Compared to the traditional video summarization tasks, the video summarization task guided by natural language adds text information t as additional input to the mapping f to control the mapping behavior. This requires the mapping f to consider more complex multimodal information.

3.2 Video Graph Generator

Consistent with VideoSAGE [43], we represent the input video sequence as three graphs: a forward graph $G^f(V, E^f)$, a backward graph $G^b(V, E^b)$, and an undirected graph $G^u(V, E^u)$. The reason for modeling video sequences as three types of graphs is to incorporate additional temporal ordering. If only an undirected graph is used, for any node, both the previous and next frames are its neighbors. The undirected graph is not clear about the chronological order of these two frames, so it is necessary to add a unidirectional flow graph of information, such as a forward graph. The reason for adding a backward graph is to bring information about the video after each frame, which helps the node locate its role in the entire video sequence.

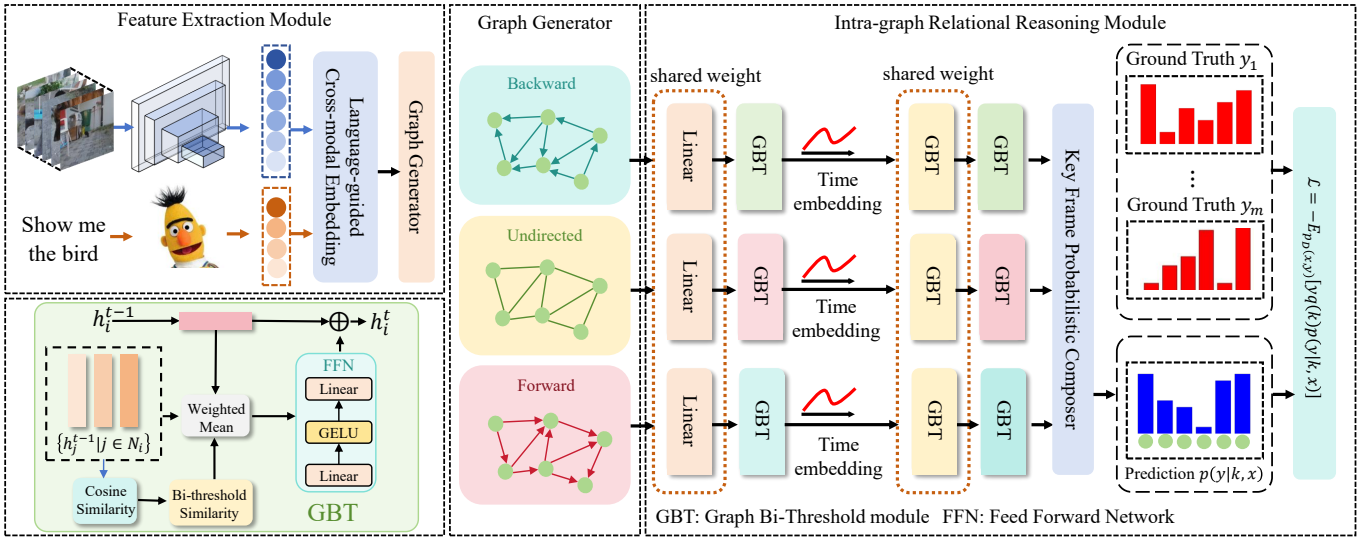

Fig. 2. The overall architectures of proposed method. Input features from video and text modalities are fused using the Language-Guided Cross-Modal Embedding module. The fused input is then transformed into three graphs by the Graph Generator, which are subsequently fed into the three branches of the intra-graph relational reasoning Module. Outputs from all branches are combined by addition to generate the model's prediction. Finally, BCE loss is used to compute the overall loss between the prediction and the ground truths.

Each video frame $i \in V$ is considered a node in the graph, with the representation vector $\boldsymbol{x}_i \in \mathbb{R}^{D_x}$ representing the features extracted from that frame using GoogLeNet [64]. If the time interval between two frames is less than the threshold τ, an edge connects them. In the undirected graph G^u, the edge is undirected. In the forward graph G^f, the edge is directed from the earlier frame to the later frame, while in the backward graph G^b, it flows in the opposite direction. After generating the graph representation of the video sequence, we use the intra-graph relational reasoning Module to extract relevant information from the graph and generate summarization scores.

3.3 Intra-graph Relational Reasoning Module

3.3.1 Graph Bi-Threshold Module

Due to the excellent performance of graph convolutional networks in graph node classification tasks, we employ graph convolution [65] for node classification, extracting sufficient features from the three graphs generated by the Video Graph Generator to generate summarization scores. The graph convolutional neural network, consists of a permutation-invariant aggregation function $g : \mathbb{R}^{(|N_i|+1)\times D_{in}} \rightarrow \mathbb{R}^{D_{in}}$ and a permutation equivariant iterative function $f : \mathbb{R}^{2\times D_{in}} \rightarrow \mathbb{R}^{D_{out}}$, where N_i is the set of neighbors adjacent to node i, D_{in} and D_{out} are the input and output dimensions of each node in the graph network.

The aggregation function g summarizes the features of the neighboring nodes $\left\{\boldsymbol{h}_j^{t-1}|j \in N_i\right\}$ in the graph, capturing the overall impact of information propagation from neighboring nodes to the target node. The iteration function f integrates the summarized information with the current node information $\boldsymbol{h}_i^{t-1} \in \mathbb{R}^{Din}$ to generate the updated node information $\boldsymbol{h}_i^t \in \mathbb{R}^{Dout}$. The process of graph convolution can be represented by :

$$\boldsymbol{h}_i^t = f(\boldsymbol{h}_i^{t-1}, g(\boldsymbol{h}_i^{t-1}, \boldsymbol{h}_j^{t-1}|j \in \boldsymbol{N}_i)), \quad (1)$$

where $f(\cdot)$ is the iterative function, $g(\cdot)$ is the aggregation function, $\boldsymbol{h}_i^t$ represents the iterative result of the i-th node in the t-th layer, and $\boldsymbol{N}_i$ represents the set of neighboring nodes of i.

The aggregation functions in classic graph convolutional networks (GCN [66], GAT [67], SAGE [68], etc.) combine all neighbor information. This approach works well when the graph structure is clearly defined. However, if the graph structure contains errors, such as edges between nodes that should not be neighbors, noisy information can be erroneously combined with the target node. For example, in videos, camera jumps, slow motion, close-ups, and other operations may cause adjacent frames to be semantically non-adjacent, even if the time interval between these frames is within the threshold. Due to special operations such as shot jumps and close ups in videos, graphs generated solely based on time information in Video Graph Generator will connect many frames that are not semantically neighbors. This will result in traditional graph networks incorporating interference information from erroneous neighbors when processing nodes. Therefore, we need to develop a new graph network structure that eliminates interference information from erroneous neighbors based on the semantic information of nodes.

Variational inference can be used to analyze which nodes are semantically true neighbors. For any node h_i, the event that h_j is a neighbor of h_i is defined as z, and the posterior probability of z is $p(z|h_i, h_j)$. Similar to the common practice of graph representation learning [69], we define the posterior probability as $softmax(cosine(h_i, h_j))$, and the probability modeled by our model is $q(z)$. However, in the early stages of training, graph networks have limited representation capabilities, and at this time, h_i and h_j may not be extracted accurately. In order to increase the robustness of the model, we need to provide strong priors for the distribution of $q(z)$ to reduce model sensitivity. The prior we propose is that $q(z)$ is a ternary distribution with values of

0, α_1, or α_2, where α_1 and α_2 are constants. This ensures that even during network training, updates to the representation vectors of graph nodes will not significantly cause changes in $q(z)$, resulting in stable convergence. We use variational inference to obtain the actual value of $q(z)$ in the network, specifically:

$$\max_{q(z)} \sum_{z} q(z) log(\frac{p(z, x_i, x_j)}{q(z)}). \tag{2}$$

By using the solution result of q (z) as the aggregation weight, we propose the aggregation function g, considers the cosine similarity between the representation of node $\boldsymbol{h}_i^{t-1}$ and the representations of neighboring nodes $\left\{\boldsymbol{h}_j^{t-1} | j \in \boldsymbol{N}_i\right\}$ when aggregating the neighbor information of node i. Neighbors with low similarity are considered semantically non-adjacent, and their aggregation weight is reset to 0. The aggregation weight of close neighbors with high similarity is reset to α_1, while the weight for the remaining neighbors is set to α_2. The aggregation function g is designed as follows:

$$\begin{aligned} \boldsymbol{m}_i^t &= g(\boldsymbol{h}_i^{t-1}, \boldsymbol{h}_j^{t-1} | j \in \boldsymbol{N}_i) = \textstyle\sum_{j \in \boldsymbol{N}_i} \alpha_{ij}^t \boldsymbol{h}_j^{t-1}, \\ \alpha_{ij}^t &= \begin{cases} 0, & \frac{\boldsymbol{h}_i^{t-1} \cdot \boldsymbol{h}_j^{t-1}}{\|\boldsymbol{h}_i^{t-1}\| \|\boldsymbol{h}_j^{t-1}\|} < \tau_1, \\ \alpha_1, & \frac{\boldsymbol{h}_i^{t-1} \cdot \boldsymbol{h}_j^{t-1}}{\|\boldsymbol{h}_i^{t-1}\| \|\boldsymbol{h}_j^{t-1}\|} > \tau_2, \\ \alpha_2, & otherwise, \end{cases} \end{aligned} \tag{3}$$

where $\boldsymbol{m}_i^t$ represents the aggregation result of the i-th node neighbor in the t-th layer, α_{ij}^t represents the attention of the i-th node to the j-th node in the t-th layer, $\|\cdot\|$ represents the 2-norm, τ is the threshold.

The iterative function $f(\cdot)$ calculates the change in node information h_i^{t-1} based on the aggregation result and normalizes it to avoid gradient explosion. This process can be expressed as:

$$f(\boldsymbol{h}_i^{t-1}, \boldsymbol{m}_i^t) = GN(\boldsymbol{h}_i^{t-1} + \boldsymbol{W}_2 \sigma(\boldsymbol{W}_1 \boldsymbol{m}_i^t + b_1) + b_2), \tag{4}$$

where the nonlinear activation function $\sigma(\cdot)$ uses GELU [70] to capture complex nonlinear information, the normalization function $GN(\cdot)$ represents Graph Normalization [71] to accelerate the training speed, and $\boldsymbol{W}_i^t$ and b represent the weight metric and bias term in the i-th MLP layer.

The output of Graph Bi-Threshold (GBT) module $\boldsymbol{z}_i$ from the final layer is designated as the score for determining whether node i is a summary frame. When this score is passed through the sigmoid function, it becomes the probability that node i is a summary frame. This process can be expressed as:

$$p(\boldsymbol{y}_i = 1 | \boldsymbol{z}_i) = Sigmoid(\boldsymbol{z}_i), \tag{5}$$

where $\boldsymbol{y}_i$ is the binary classification result of the i-th node, and $\boldsymbol{z}_i$ is the GBT output of the i-th node.

3.3.2 Time Embedding

When modeling a video as a graph, simply considering two frames with time intervals less than a threshold as neighbors can result in the loss of certain temporal features that represent the sequence order. For example, if three frames are arranged in chronological order as $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{x}_3$, and the time interval between $\boldsymbol{x}_1$ and $\boldsymbol{x}_3$ is less than the threshold τ, then both $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ will be considered neighbors of $\boldsymbol{x}_1$. From the perspective of $\boldsymbol{x}_1$, these two frames appear identical, and it is not apparent that $\boldsymbol{x}_2$ is actually closer to $\boldsymbol{x}_1$. Therefore, it is necessary to incorporate the temporal information of each frame into the input of the graph neural network.

In certain hidden layers of the network, time embedding is incorporated to modify the hidden layer representation:

$$\boldsymbol{h}_i^t \leftarrow \boldsymbol{h}_i^t + \boldsymbol{\tau}_i^e, \tag{6}$$

where $\boldsymbol{\tau}_i^e$ is the time embedding of the i-th node, that allows the model to dynamically update the time representation during training.

3.4 Language-Guided Cross-Modal Embedding Module

Traditional information fusion methods such as cross attention mechanism use fully learnable weights for fusion. This behavior of entrusting all behavior to autonomous learning by the network can lead to loss of controllability. Network behavior is completely driven by data, and it is difficult to ensure that the network learns sufficiently robust strategies in the case of limited data. Therefore, we integrate text nodes and video nodes into heterogeneous graphs to achieve multimodal information fusion. By manually constructing edges of heterogeneous graphs, we can explicitly control the behavior of information fusion, thereby ensuring controllable network behavior.

To enable video summarization guided by natural language, we incorporate the user's natural language as an integral part of the model's input. Since video sequences are already represented as distinct nodes within a graph, it is logical to treat natural language sequences as nodes as well. Consequently, connections are established between nodes representing video and natural language, forming a heterogeneous graph $G(\boldsymbol{x}_v, \boldsymbol{x}_s, E)$. In this graph, video nodes $\boldsymbol{x}_v$ are defined as before, while text nodes $\boldsymbol{x}_s$ represent tokens from the natural language sequence s, with node representation vectors $\boldsymbol{x}_s \in X_s$ being the feature vectors extracted by BERT [72]. To fully integrate linguistic information into the video sequence, each token node is connected to all video nodes via directed edges.

The primary objective of this module is to incorporate natural language sequence information into the video sequence effectively. Since the integrated video sequence will be represented as a graph with edges between adjacent frames and then processed by the GBT. Because BERT already accounts for the interrelations between natural language tokens, this layer would disregard all edges except those from natural language tokens to video nodes to minimize parameters, focusing solely on integrating natural language information into the video sequence. Due to the complexity of text features, we utilize a graph attention mechanism instead of bi-thresholds as weights for the aggregation function g, which can be defined as follows:

$$\begin{aligned} g(\boldsymbol{h}_i^{t-1}, \boldsymbol{h}_j^{t-1} | j \in N_i) &= \textstyle\sum_{j \in N_i} \alpha_{ij}^t (\boldsymbol{W}_1^t h_j^{t-1} + b_1^t), \\ \alpha_{ij}^t &= \frac{exp((\boldsymbol{W}_2^t h_i^{t-1} + b_2^t)' (\boldsymbol{W}_3^t h_j^{t-1} + b_3^t))}{\sum_{j \in N_i} exp((\boldsymbol{W}_2^t h_i^{t-1} + b_2^t)' (\boldsymbol{W}_3^t h_j^{t-1} + b_3^t)}, \end{aligned} \tag{7}$$

where α is the attention that integrates textual information into video nodes, $\boldsymbol{W}_i^t$ and b_i^t are weight metric and bias term of the i-th node, respectively.

The iterative function $f(\cdot)$ retains the same structure as GBT, employing two layers of MLP and a GELU activation function to integrate complex information from the text.

3.5 Biased Cross Entropy Loss

For the same video, different annotators may produce varied summaries based on their preferences. To leverage multiple labels from different annotators for the same video, the conventional approach is to compute the arithmetic mean of these labels as the summary score, and then use cross-entropy or MSE loss to calculate the error in the model's output. However, these methods discard the preference information embedded in these labels. For example, the average of [1,0] and [0,1] is [0.5, 0.5], which represents the highest entropy (uncertainty). In light of this, we introduce the use of BCE loss for video summarization, reevaluating the significance of each label by employing a weighted mean instead of an arithmetic mean to achieve a convergence target that incorporates preferences.

Given the possibility of multiple summarization schemes for the same video, we model the output using a mixed Bernoulli distribution and derive the preference cross-entropy loss (BCE) as the loss function for LGRLN video summarization training. If the video feature is x and the label is y, this results from a specific summarization strategy. The same video may have multiple summarization strategies k, such as extracting the first or last few frames of each scene with the same shot. However, when the strategy is well-defined, there will be only one summarization scheme per video under that strategy. Therefore, the probability of modeling video summarization is $p(y|x) = \sum_k p(k)p(y|x,k)$. Under this model, the log-likelihood function of the dataset D is $E_{p_D(y,x)}\left[log(p(y|x))\right]$, and $log(p(y|x))$ can be decomposed into $\sum_k q(k)log(\frac{p(y,k|x)}{q(k)}\frac{q(k)}{p(k|y,x)})$, which allows us to use the EM algorithm [73] to solve the maximum likelihood problem of this mixed distribution.

The closest posterior probability $q(k)$ is obtained in step E, which can be written as:

$$\min_{q(k)} E_{p_D(y|x)}\left[\sum_k q(k)log(\frac{q(k)}{p(k|y,x)})\right], \tag{8}$$

where $\sum_k q(k)log(\frac{q(k)}{p(k|y,x)})$ can be decomposed into:

$$\begin{aligned}\sum_k q(k)log(q(k)) - \sum_k q(k)log(p(y|k,x)) \\ - \sum_k q(k)log(p(k|x)) + log(p(y|x)),\end{aligned} \tag{9}$$

where q can be any distribution used to assist in training the model distribution p.

To simplify the model, assume $q(k)$ is a binary function, which makes its entropy $-\sum_k q(k)log(p(k|x))$ a constant. Meanwhile, due to the lack of a clear arrangement order for k, the prior distribution $p(k|x)$ is uniform, making $log(p(k|x))$ a constant. Additionally, in the objective function for $q(k)$, $p(y|x)$ is also a constant. Therefore, the above minimization problem is equivalent to maximizing $\sum_k q(k)log(p(y|k,x))$, which has an analytical solution:

$$\begin{aligned}q(k) = \begin{cases} b, & k \in S(x,y), \\ a, & otherwise, \end{cases} \\ s.t. \\ 0 \leq a < b < 1, \\ \sum_k q(k) = 1, \\ \forall k_1 \in S, k_2 \in \bar{S}, log(y|k_1,x) > log(y|k_2,x),\end{aligned} \tag{10}$$

where a and b are hyperparameters, and the larger the difference between them, the greater the entropy of q. S is the subset closest to the model output x in the ground truth y, and $\bar{S}$ is the complement of all ground truths to S.

The model parameters only need to be optimized in step M. Step M involves solving the maximization problem, which can be defined as:

$$\max_{p(y|k,x)} E_{p_D(y,x)}\left[\sum_k q(k)log(\frac{p(y,k|x)}{q(k)})\right], \tag{11}$$

where P_D is the distribution in the dataset, and k is the latent variable, representing which strategy to use for video summarization. Among them, Eq. (11) can also be decomposed into:

$$\begin{aligned}\sum_y \sum_k p_D(y,x)q(k)log(p(y|k,x)) + \\ \sum_y p_D(y,x) \sum_k q(k)log(p(k|x)) - \\ \sum_y p_D(y,x) \sum_k q(k)log(q(k)).\end{aligned} \tag{12}$$

As in the analysis of step E, $\sum_k q(k)log(p(k|x))$ and $\sum_k q(k)log(q(k))$ are constants variables, so only $\sum_y \sum_k p_D(y,x)q(k)log(p(y|k,x))$ needs to be optimized. To further simplify the model, it is required that the model only needs to learn the summary scheme under one strategy, meaning it only needs to model one corresponding output. In this case, the maximization problem in step M is simplified to $\sum_{y,x} p_D(y,x)q(k)log(p(y|k,x))$. Since whether a frame is a keyframe y is a binary variable, the above equation is equivalent to:

$$\sum_{y,x} p_D(y,x)q(k)(ylog(p(+|k,x)) + (1-y)log(p(-|k,x))), \tag{13}$$

where $p(+|k,x)$ is the probability that the model considers frame x a keyframe in policy k, and $p(-|k,x)$ is the probability that the model considers x a non keyframe, y is the ground truth, and $q(k)$ comes from Eq. (10), p is the predicted of the model. The difference between the above equation and standard cross-entropy is that the label is weighted by $q(k)$, which is referred to as biased cross-entropy.

This loss function enables the model output to approximate a subregion of the reasonable interval, rather than converging to the mean of the interval, which may lie outside the reasonable range.

4 Experiments

4.1 Experimental Settings

4.1.1 Datasets

Our experiments are primarily conducted on two benchmark datasets: SumMe [74] and TVSum [75]. SumMe con-

Algorithm 1 Training framework for the proposed approach

Input: Dataset X, where in any item (x, t, y) of the dataset, x is a video, t is the natural language guidance of the video summary, and y is the m ground truths of the manually provided video summary.

Output: Updated network parameters.

Extract features from the video and text: (v, t) in the dataset as (x_v, x_t).
for each epoch **do**
  for (x_t, x_v, y) in X **do**
    $V \leftarrow LGCME(x_t, x_v)$ ▷ Language Guided Cross Modal Embedding Module
    $G^f, G^b, G^u \leftarrow VGG(V)$ ▷ Video Graph Generator
    $\hat{y} \leftarrow IGRR(G^f, G^b, G^u)$ ▷ Intra-graph Relational Reasoning Module
    $L \leftarrow BCE(\hat{y}, y)$
    Optimize model parameters using optimizers such as AdamW.
  end for
end for

sists of 25 videos from diverse users and scenarios (e.g., cooking, racing), with each video annotated by 15-18 users to indicate whether each frame is a summary frame. TVSum comprises 50 videos from YouTube, each annotated by 20 users. Additionally, we employed the VideoXum [27] dataset and the QFVS [76] dataset in our analysis. The VideoXum dataset, which consists of over 10,000 videos, serves as a valuable resource for assessing the model's ability to extract insights from large-scale data.

4.1.2 Evaluation Protocol

In line with established practices, a 5-fold cross-validation method is applied to the SumMe, TVSum, and VideoXum datasets. F1 scores are calculated on 4 splits for the QFVS dataset. The datasets are randomly divided into five equal parts, with 20% designated as the test set and the remaining 80% used as the training set for the experiments. The final experimental result is obtained by averaging the outcomes of the five individual experiments. For VideoXum, we employ a distinct approach, using a partitioned dataset with a separate training set for model training, a test set for monitoring training progress, and a validation set for evaluating the final results. For evaluation criteria, we follow industry standards, utilizing the F1 score of the summary results, along with the τ and ρ scores, to assess our model's performance. The F1 score is calculated as follows:

$$F_1 = \frac{2PR}{P+R}, \tag{14}$$

where P is the accuracy calculated using $|S \cap \widehat{S}|/|\widehat{S}|$, R is the recall calculated using $|S \cap \widehat{S}|/|S|$, S is the ground truth and $\widehat{S}$ is the model prediction value. Additionally, τ and ρ represent the Kendall and Spearman correlation coefficients, respectively, which quantify the correlation between the model's output scores and the ground truth labels.

4.1.3 Implementation Details

We have implemented and conducted our experiments following a well-defined and reproducible framework. As illustrated in Algorithm 1, we have utilized GoogLeNet as the image feature extractor for the SumMe and TVSum datasets, while Blip [77] has been specifically employed for the VideoXum dataset to better capture visual representations. To optimize the training process, we have adopted AdamW [78] as the optimizer, setting the dropout rate to 0.4 and an L2 regularization term of 0.01 to effectively mitigate overfitting and improve generalization. In line with previous research, we have applied Kernel Temporal Segmentation (KTS) [79] to segment videos and utilized the Knapsack algorithm [80] to solve the knapsack problem, leveraging the scores generated by our model to determine the optimal video summary solution. This methodological approach has ensured a structured and efficient summarization process.

Furthermore, we have conducted all experiments on a single NVIDIA RTX 3090 GPU, demonstrating the computational efficiency of our method. The training has been completed within 6 hours across all dataset splits, including SumMe, TVSum, and QFVS, which contain dozens of videos, as well as the large-scale VideoXum dataset with thousands of videos. These details emphasize the efficiency and scalability of our approach, making it feasible for real-world applications, including deployment on resource-constrained devices.

4.2 Performance Comparison

4.2.1 Performance on Traditional Dataset

We conducted comprehensive performance comparisons on the SumMe, TVSum, and VideoXum datasets between our model with current mainstream baselines, as shown in Table 1. In the SumMe dataset, human performance sets a high benchmark with an F1 score of 54.0 and correlation metrics (Kendall's τ and Spearman's ρ) both at 0.21. Among non-GNN models, the performance varies, with F1 scores ranging from 38.6 to 55.6. dqqLSTM performs the lowest in this group, while A2Summ and PGL-Sum reach the highest F1 scores but with relatively low correlation values. GNN-based models generally show stronger results, with SumGraph and LGRLN (ours) standing out. LGRLN nearly matches human performance with an F1 score of 54.7 and achieves the highest correlation metrics among all models (Kendall's τ = 0.14, Spearman's ρ = 0.19), indicating its effectiveness in summarization tasks within this dataset. Compared to VideoSAGE, our LGRLN model supports query-guided multimodal summarization by integrating textual inputs, employs a bi-threshold graph convolution for more precise semantic filtering, and enables personalized summary generation tailored to user intent.

In the TVSum dataset, non-GNN models show improved F1 scores compared to their SumMe results, with SUM-DeepLab, A2Summ, and PGL-Sum achieving the highest scores, particularly PGL-Sum, which also leads in correlation metrics among non-GNN models. GNN-based models further demonstrated the superiorities in this dataset, with LGRLN (ours) achieving one of the highest F1 scores at 58.7, and strong correlation metrics (Kendall's τ = 0.29, Spearman's ρ = 0.41), closely following the highest values observed. This underscores the strong performance of GNN-based approaches, particularly LGRLN (ours), in summarization tasks for TVSum.

TABLE 2
The Performance Comparison Between the LGRLN with Other Baselines in SumMe and TVSum Datasets.

Type	Model	SumMe			TVSum		
		F1 score ↑	Kendall's τ ↑	Spearman's ρ ↑	F1 score ↑	Kendall's τ ↑	Spearman's ρ ↑
Human [26]	-	54.0	0.21	0.21	54.0	0.18	0.20
Others	SumTransfer [81]	40.9	-	-	-	-	-
	SUM-DeepLab [82]	48.8	-	-	58.4	-	-
	dqqLSTM [83]	38.6	0.04	0.06	54.7	-	-
	DR-DSN [84]	42.5	0.05	0.05	58.1	0.02	0.03
	HSA-RNN [85]	42.3	0.06	0.07	58.7	0.08	0.09
GNN	VideoSAGE [43]	46.0	0.12	0.16	58.2	0.30	0.42
	RSGN [26]	45.0	0.08	0.09	60.1	0.08	0.09
	SumGraph [48]	51.4	-	-	**63.9**	0.09	0.14
	LGRLN(ours)	**54.7**	**0.14**	**0.19**	58.3	**0.30**	**0.43**

TABLE 3
The Performance Comparison in VideoXum Dataset.

Model	VideoXum		
	F1 score ↑	Kendall's τ ↑	Spearman's ρ ↑
Human [27]	33.8	0.305	0.336
Frozen-BLIP [27]	16.1	0.008	0.011
VSUM-BLIP [27]	23.1	0.185	0.246
VTSUM-BLIP [27]	22.7	0.176	0.232
LGRLN(ours)	**32.1**	**0.198**	**0.262**

4.2.2 Performance on Large Scale VideoXum Dataset

As shown in Table 3, ours LGRLN stands out as the most effective model in the VideoXum dataset, achieving the highest F1 score of 32.1 and the best correlation metrics with human judgments (Kendall's τ = 0.198, Spearman's ρ = 0.262). Notably, LGRLN's use of graph-based representation effectively captures both temporal and contextual relationships within video frames, allowing for a more comprehensive understanding of the video content. Additionally, the bi-threshold graph convolution mechanism incorporated in LGRLN enables the model to intelligently filter and prioritize relevant information from neighboring frames, reducing noise and enhancing the precision of the summaries. These design choices contribute to LGRLN's ability to outperform other models, such as VSUM-BLIP and VTSUM-BLIP, which show moderate performance with F1 scores of 23.1 and 22.7, respectively. In contrast, the performance of Frozen-BLIP significantly decline with an F1 score of 16.1 and very low correlation scores, indicating its limited capacity to effectively summarize video content. The results suggest that LGRLN's advanced graph representation and convolution mechanisms are crucial for its superior performance in video summarization tasks.

4.2.3 Performance on Multimodal Dataset

The QFVS dataset is a multimodal resource designed for natural language-guided video summarization. As presented in Table 4, LGRLN outperforms other models, achieving the highest average F1 score of 50.91, with a precision of 53.27 and a recall of 53.76. This superior performance is consistent across all four video scenarios, where LGRLN maintains a strong balance between precision and recall, resulting in highly effective summarization outcomes. LGRLN's strong performance can be attributed to its advanced graph-based representation and bi-threshold graph convolution mechanism, enabling it to capture and integrate temporal and contextual dependencies within video data effectively. These features allow LGRLN to more effectively distinguish and prioritize relevant information from video frames, resulting in more accurate and contextually relevant summaries. In contrast, models like QC-DPP, though somewhat effective with an average F1 score of 44.19, struggle to maintain consistency and accuracy, especially when handling the variability of video content. Models such as SeqDPP and SH-DPP, with average F1 scores of 30.92 and 33.38, respectively, underscore the difficulty of achieving high recall without compromising precision. LGRLN's consistent performance across various metrics highlights the effectiveness of its graph-based architecture and convolutional mechanisms.

4.2.4 Comparison with Large Language Modeling Methods

To demonstrate the balanced performance and cost advantages of LGRLN over LLM based methods, we compared our proposed method with LLM based video summarization methods in Table 5. LGRLN shows significant advantages in both performance and computational efficiency over large video models (LVMs) like VideoXum-LLaMA-7B and VideoXum-LLaMA-13B. Although it has only 3M parameters, LGRLN achieves comparable or even superior performance, reducing the parameter count by approximately 87.8% to 91.7%. This significant reduction in parameters leads to lower computational and storage demands, substantially cutting training and inference time. For example, LGRLN can be trained and run on a single RTX3090 GPU, while the larger models require 8 A100 GPUs. In terms of performance, LGRLN excels with F1 scores of 54.7 on the SumMe dataset and 58.3 on the TVSum dataset, outperforming methods like M3SUM(SP) and M3SUM(PCoT), which have F1 scores of 43.6 and 41.9, respectively. On the VideoXum dataset, LGRLN outperforms VideoXum-

TABLE 4
The Performance Comparison Between the LGRLN with Other Baselines in QFVS Dataset.

Model	Vid 1			Vid 2			Vid 3			Vid 4			Average		
	P ↑	R ↑	F1 ↑	P ↑	R ↑	F1 ↑	P ↑	R ↑	F1 ↑	P ↑	R ↑	F1 ↑	P ↑	R ↑	F1 ↑
SeqDPP [86]	53.43	29.81	36.59	44.05	46.65	43.67	49.25	17.44	25.26	11.14	63.49	18.15	39.47	39.35	30.92
SH-DPP [87]	50.56	29.64	35.67	42.13	46.81	42.72	51.92	29.24	36.51	11.51	62.88	18.62	39.03	42.14	33.38
QC-DPP [88]	49.86	53.38	48.68	33.71	62.09	41.66	55.16	62.40	56.47	21.39	63.12	29.96	40.03	60.25	44.19
TPAN [89]	49.66	50.91	48.74	43.02	48.73	45.30	58.73	56.49	56.51	36.70	35.96	33.64	47.03	48.02	46.05
CHAN [90]	54.73	46.57	49.14	45.92	50.26	46.53	59.75	64.53	58.65	25.23	51.16	33.42	46.40	53.13	46.94
LGRLN(ours)	**60.79**	43.69	**49.72**	**51.91**	59.27	**54.94**	**70.09**	51.44	**59.03**	30.28	60.66	**39.97**	**53.27**	53.76	**50.91**

TABLE 5
Comparison of performance and computational costs with the large language model on various datasets.

Model	SumMe			TVSum			VideoXum			Parameters	Training environment
	F1 ↑	τ ↑	ρ ↑	F1 ↑	τ ↑	ρ ↑	F1 ↑	τ ↑	ρ ↑		
V2Xum-LLaMA-7B [62]	-	0.222	0.293	-	0.296	0.378	29.0	0.204	0.298	7B	8 x A100
V2Xum-LLaMA-13B [62]	-	-	-	-	-	-	31.6	0.200	0.276	13B	8 x A100
M3SUM(SP) [63]	43.6	-	-	56.9	-	-	-	-	-	over 20B	-
M3SUM(PCoT) [63]	41.9	-	-	57.6	-	-	-	-	-	over 20B	-
LGRLN(ours)	54.7	0.14	0.19	58.3	0.30	0.43	32.1	0.198	0.262	3M	1 x RTX3090

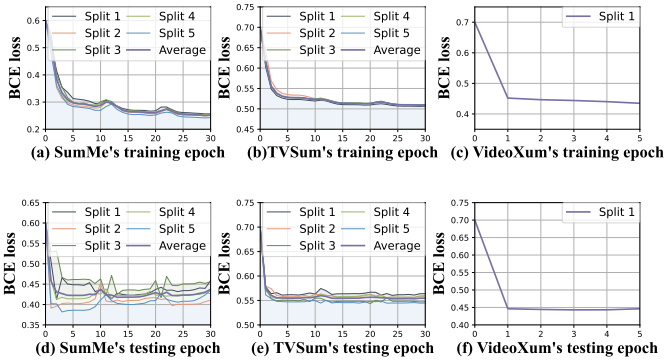

Fig. 3. Learning convergence curves on mainstream datasets.

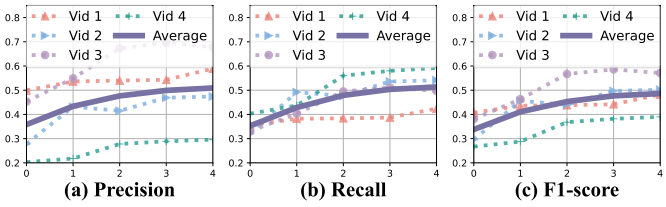

Fig. 4. Learning convergence curves on QFVS dataset.

LLaMA-7B and VideoXum-LLaMA-13B, with F1 scores of 32.1, surpassing their scores of 29.0 and 31.6, respectively.

4.2.5 Convergence Stability and Generalization Analysis

The learning curves in Fig. 4 depict the training and testing convergence of the LGRLN model on the SumMe, TVSum, and VideoXum datasets. In the SumMe and TVSum datasets, the training BCE loss decreases steadily over 30 epochs, indicating effective learning and convergence with minimal variance across different splits. This consistency across splits indicates robust model performance and stable convergence. The test BCE loss shows some fluctuations, particularly in the early epochs, possibly due to variations in the validation sets or the inherent complexity of the datasets. Despite these fluctuations, the test loss stabilizes as training progresses, confirming the model's ability to generalize well across different splits. In contrast, the VideoXum dataset exhibits rapid convergence in the first few epochs, with both training and test losses quickly reaching low values and stabilizing, suggesting either a simpler dataset or a more effective model fit. Overall, the learning curves indicate that the LGRLN model achieves efficient and stable convergence across diverse video summarization tasks, with minor fluctuations mainly in more complex datasets like SumMe and TVSum.

4.3 Ablation Study

4.3.1 The Effectiveness of Each Model Components

Table 6 presents the impact and trends of different components on model performance. The baseline achieves an F1-max of 47.1 and F1-mean of 16.9 on SumMe, and 78.3 and 56.9, respectively. For single modules, both BCE and Temb bring clear improvements on SumMe, with BCE performing slightly better, while GBT shows only minor gains. On TVSum, all three modules improve F1-max, with Temb reaching the highest value of 80.8, but the F1-mean of all single modules remains below the baseline, with values of 54.6, 55.6, and 55.6. For pairwise combinations, BCE combined with GBT achieves the best results on SumMe with F1-max of 54.1 and F1-mean of 22.1, followed by BCE combined with Temb, while Temb combined with GBT shows limited improvement. On TVSum, all three combinations surpass the baseline in F1-max, with BCE and Temb achieving the

TABLE 6
Ablation Study on Different Model Components.

Model	SumMe		TVSum	
	F1-max ↑	F1-mean ↑	F1-max ↑	F1-mean ↑
Baseline	47.1	16.9	78.3	56.9
BCE	53.9	20.5	80.0	54.6
Temb	52.2	19.9	80.8	55.6
GBT	47.7	17.2	79.9	55.6
BCE+Temb	53.9	21.0	81.1	56.3
BCE+GBT	54.1	22.1	80.1	56.8
Temb+GBT	52.3	19.9	80.2	55.9
LGRLN(ours)	**55.9**	**23.1**	**83.0**	**59.5**

TABLE 7
Ablation Study on Different Weight Hyper-parameters.

Parameters		SumMe		TVSum	
a	b	F1-max ↑	F1-mean ↑	F1-max ↑	F1-mean ↑
0.000	0.500	54.6	16.8	62.8	42.6
0.007	0.187	52.9	20.0	71.7	48.3
0.020	0.160	44.8	19.4	77.5	53.9
0.033	0.133	52.3	22.2	72.3	49.5
0.070	0.100	50.4	22.8	74.8	52.0

highest at 81.1, but none of the combinations exceed the baseline in F1-mean, where BCE and GBT perform best at 56.8. The complete model LGRLN achieves the best performance across all metrics on both datasets, reaching 55.9 and 23.1 on SumMe, and 83.0 and 59.5 on TVSum. These results not only outperform all pairwise combinations but also make LGRLN the only configuration that clearly surpasses the baseline in F1-mean on TVSum, demonstrating that the deep integration of multiple components provides significant synergistic effects and enhanced stability.

4.3.2 The Effectiveness of Different Hyper-parameters

The ablation in Table 7 examines how the weight parameters a and b affect performance. On SumMe, reducing the difference between b and a generally increases F1-mean. For example, a=0.070 and b=0.100 gives F1-mean 22.8, while a=0.000 and b=0.500 gives F1-mean 16.8. The largest difference yields the highest F1-max 54.6 but with the lowest F1-mean, indicating a trade-off between peak and average performance. On TVSum, the best results occur at intermediate differences. The setting a=0.020 and b=0.160 achieves F1-max 77.5 and F1-mean 53.9, and a=0.070 and b=0.100 remains strong with F1-max 74.8 and F1-mean 52.0. A very large difference such as a=0.000 and b=0.500 degrades performance to F1-max 62.8 and F1-mean 42.6. These observations suggest that moderate or small differences between a and b are preferable for TVSum, whereas SumMe exhibits a trade-off where a larger difference can raise F1-max at the cost of F1-mean. Overall, tuning a and b controls the entropy of the weighting distribution and mid-range settings strike a better balance between fitting a single annotation strategy and generalization across annotators.

4.4 The Effectiveness of BCE Loss

To explore the intricacies of BCE loss mechanisms, we conducted a visual analysis comparing two aspects: the weighted average of all ground truths in BCE loss and the simple average of ground truths used in traditional methods. To visualize these high-dimensional ground truths, we employed two dimensionality reduction techniques: Principal Component Analysis (PCA) and ISOMAP. PCA reduces dimensionality by projecting data onto the hyperplane that captures the highest variance. In contrast, ISOMAP views data as residing on a manifold, using geodesic distances and Multidimensional Scaling (MDS) with equidistant properties to effectively reduce the data's dimensionality, enabling a more comprehensive visualization.

As depicted in Fig. 5, the outcome derived from simply averaging all ground truths frequently lies outside the confines of the confidence interval established by kernel density estimation. This discrepancy highlights the limitations of traditional averaging methods in capturing the true variability and complexity of the data. In contrast, the result attained through the BCE-weighted average tends to be more intimately aligned with a particular confidence interval, indicating a closer adherence to the underlying probability distribution. This suggests that BCE loss not only preserves critical information but also aligns the model's outputs more closely with the true data distribution, making it a more effective method for handling diverse and complex annotations in video summarization tasks. These visualizations further underscore the robustness of BCE loss in managing high variability in annotations, demonstrating its superiority over traditional averaging methods.

4.5 Analysis of Different Graph Channel Contributions

As illustrated in Fig. 6, the outputs of the forward, backward, and undirected graph channels exhibit notable differences, particularly in the summary score values assigned to various frames. Each graph channel processes the video sequence from a distinct perspective, capturing temporal and contextual information unique to its directionality. For instance, the forward graph channel emphasizes the progression of frames in chronological order, while the backward graph retraces the sequence to capture contextual dependencies in reverse. The undirected graph, on the other hand, focuses on bidirectional relationships, offering a holistic view of frame correlations without enforcing a strict temporal order.

These differences indicate that each channel independently identifies keyframes that are valuable for summarization, leveraging complementary insights from the video sequence flow. By integrating the results from these three methods, the model ensures a comprehensive and balanced selection of summary frames. This multi-channel approach enhances the robustness and completeness of the final video summary, as it synthesizes diverse temporal perspectives to generate an accurate and semantically rich output.

4.6 Temporal Dynamics Captured by Time Embedding

As shown in Fig. 7, the time embedding mechanism demonstrates a strong correlation between the embeddings of

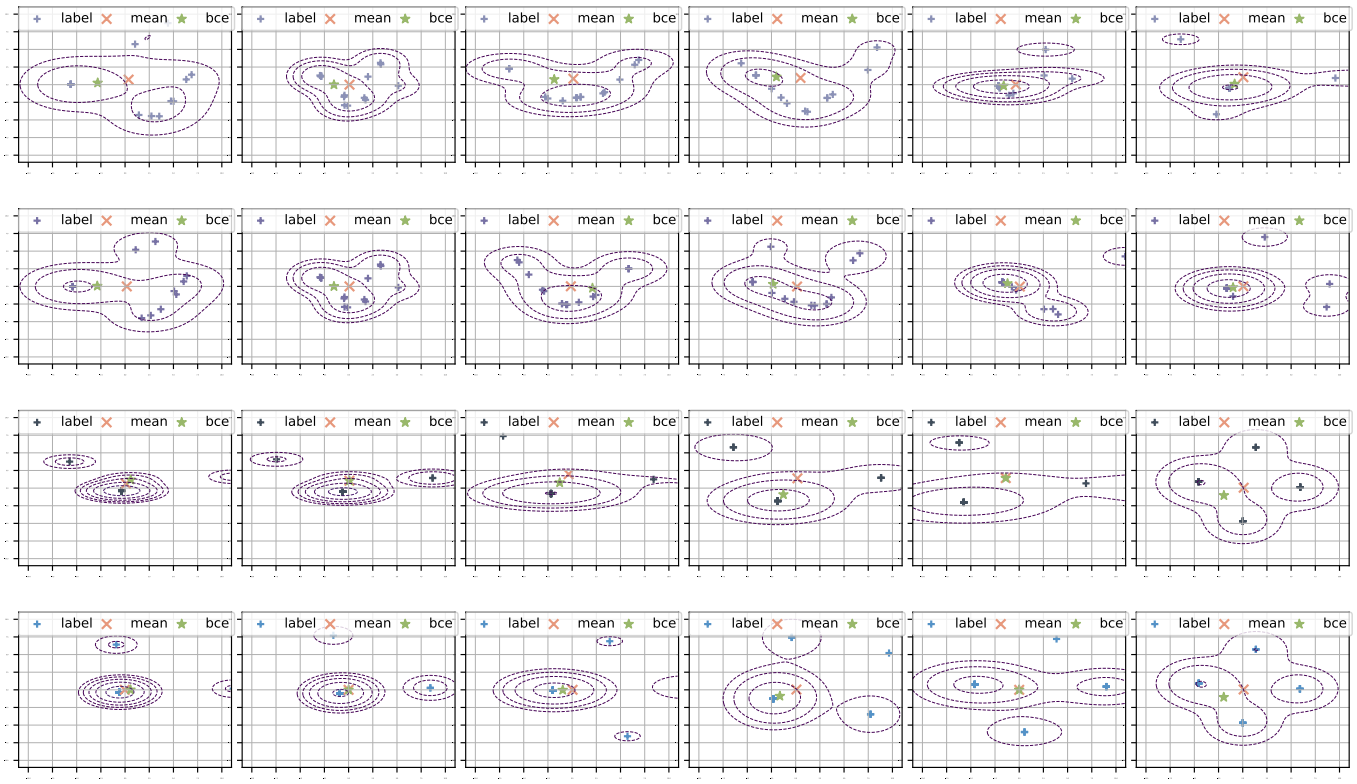

Fig. 5. The results of manually annotated summary labels for several videos in the TVSum and SumMe datasets after ISOMAP dimensionality reduction and PCA dimensionality reduction.

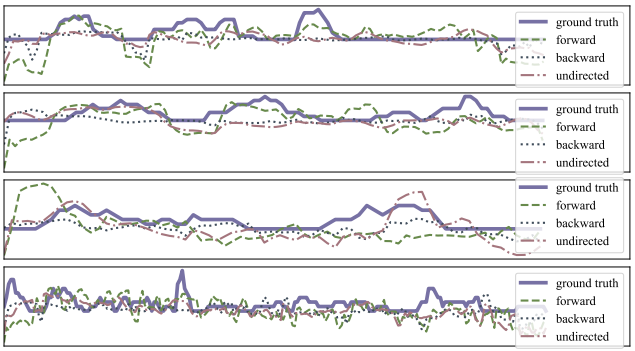

Fig. 6. Summary scores from the ground truth and the outputs of graph channels for different videos. Each graph channel captures unique temporal and contextual relationships.

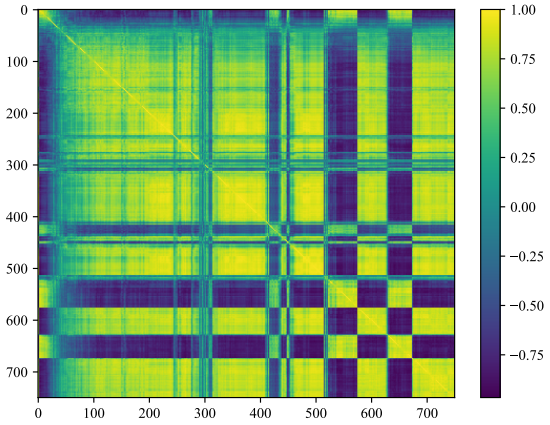

Fig. 7. Correlation heatmap of time embeddings at different time points after training. Strong correlations between adjacent moments and periodic fluctuations across intervals demonstrate the time embedding's ability to capture both immediate and long-term temporal patterns.

adjacent time points. This correlation exhibits periodic fluctuations across varying time intervals, indicating that the time embedding effectively captures temporal patterns and frequency characteristics within the video data. These periodic trends reflect the model's ability to encode temporal dependencies and cyclical behaviors inherent in the video sequence, such as repeated actions or transitions. The heatmap in Fig. 7 illustrates this phenomenon in detail, where the diagonal elements, representing the correlation of embeddings at consecutive time points, are notably strong. Additionally, the off-diagonal elements show periodic correlation patterns, signifying that the time embedding not only captures immediate temporal relationships but also recognizes long-term dependencies and periodic structures within the video. This ability to encode both short-term and long-term temporal features ensures that the time embedding aligns well with the sequential nature of video data.

By learning these temporal dynamics, the time embedding contributes to the model's understanding of sequential relationships and ensures that the temporal context is properly integrated into the summarization process. This capa-

TABLE 8
The computational cost comparison during inference in TVSum dataset.

Model	Kendall's $\tau \uparrow$	Spearman's $\rho \uparrow$	Time (ms) $\downarrow$	Parameters (MB) $\downarrow$	Total (MB) $\downarrow$
PGL-SUM [91]	0.27	0.39	113.79	36.02	55.17
A2Summ [92]	0.26	0.38	120.59	9.60	50.56
VideoSAGE [43]	**0.30**	0.42	23.55	3.52	19.27
LGRLN(ours)	**0.30**	**0.43**	**13.87**	**2.97**	**13.96**

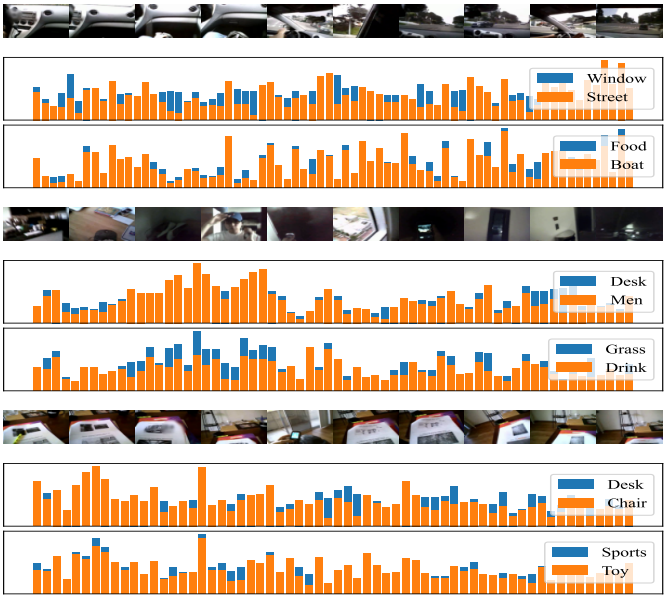

Fig. 8. The video summarization results demonstrate the effectiveness of our proposed LGRLN in handling various natural language instructions. Each row presents different instructions applied to videos, with blue and orange bars indicating the frequency of key visual elements based on the prompts.

bility enhances the model's ability to generate summaries that are not only temporally coherent but also sensitive to recurring patterns or significant time-based variations within the video content.

4.7 Summary of Videos Guided by Natural Language

We present qualitative results in Fig. 8 using BERT as the text encoder and visualize frame-level relevance as histograms, where the horizontal axis is the temporal order of frames and the vertical axis is the normalized relevance score; blue and orange bars correspond to two prompts. For each video, the first histogram employs prompts that actually occur in the clip, specifically Window versus Street for the first video, Desk versus Men for the second, and Desk versus Chair for the third. In these matched settings, LGRLN concentrates high scores on the correct temporal regions and suppresses unrelated frames, indicating accurate localization and effective language–vision alignment. To assess rejection behavior, the second histogram of each video uses prompts that are unrelated to the clip, specifically Food versus Boat in the first video, Grass versus Drink in the second, and Sports versus Toy in the third. Ideally the model should output uniformly low scores when no corresponding content is present, yet we observe non-negligible activations at certain time positions. These errors likely arise from spurious lexical associations between prompts, background co-occurrence patterns that partially resemble the scene, and temporal smoothing in graph message passing that propagates false positives across neighboring frames. This analysis clarifies both the strengths of the method in matched scenarios and its current limitation in rejecting off-topic prompts, and it motivates future improvements through negative-prompt training, confidence calibration with adaptive thresholds, contrastive suppression of irrelevant prompts, and prompt-conditioned gating in the relational reasoning module.

4.8 Comparative Analysis of Computational Efficiency

To illustrate the efficiency of our model, we performed a comparative analysis of its computational cost against several traditional video summarization models, as shown in Table 8. Our LGRLN model significantly reduces computational demands, requiring only 13.96 MB of memory and 13.87 ms for inference. Compared to the PGL-SUM model, which consumes 55.17 MB of memory and takes 113.79 ms for inference, our model reduces memory usage by approximately 74.7% and inference time by 87.8%. Even compared to the more efficient VideoSAGE model, LGRLN reduces total memory usage by 27.5% and inference time by 41.1%. Additionally, our LGRLN model uses only 2.97 MB of parameters, a reduction of 91.7% compared to PGL-SUM and 69.1% compared to A2Summ, which require 36.02 MB and 9.60 MB of parameters, respectively. Despite this dramatic reduction in resource usage, our model maintains competitive performance among the tested models, indicating a strong correlation between the predicted summaries and the ground truth.

In summary, the LGRLN model demonstrated superiorities in producing high-quality video summaries while maintaining remarkable computational efficiency. These reductions in resource usage, up to 91.7% in parameter count and 87.8% in inference time, highlight its suitability for practical applications where both performance and resource optimization are crucial.

4.9 Deployment considerations and limitations

A practical scenario that motivates our summarization, where a user's natural-language query guides the selection of a compact set of key frames or shots to create a temporally coherent visual summary of the video. In contrast, video captioning aims to generate free-form textual descriptions for a given video and has seen rapid progress. numbers reported in this paper are measured on desktop-class hardware and are intended to indicate relative efficiency. Actual

on-device deploy ability depends on factors not evaluated here, including memory footprint under the target runtime (e.g., NNAPI/Core ML), operator availability/fusion, I/O and pre/post-processing overheads, and potential quantization/distillation. While our model has only 3M parameters and shows reduced inference cost, demonstrating mobile feasibility requires hardware-specific profiling and engineering, which we leave for future work.

5 Conclusion

In this paper, we propose a novel Language-Guided Graph Representation Learning Network (LGRLN) to tackle the challenges of video summarization, with a focus on capturing temporal dependencies and integrating multimodal user inputs. By transforming video frames into structured graphs and integrating user-provided language instructions, our approach generates personalized and contextually relevant video summaries. Experimental results show that our method outperforms existing approaches across multiple benchmarks, particularly excelling in multimodal tasks. Furthermore, our application of a mixture Bernoulli distribution model for managing diverse annotation sets a new standard in accommodating user preferences in summary generation. Future work could involve further refinement of graph representation and the integration of additional modalities, such as audio or user interaction data, to enhance the adaptability of the summarization process.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (2023YFA1008501) and the National Natural Science Foundation of China (NSFC) under grant 624B2049 and U22B2035.

References

[1] W. Li, W. Han, L.-J. Deng, R. Xiong, and X. Fan, "Spiking variational graph representation inference for video summarization," *IEEE Transactions on Image Processing*, vol. 34, pp. 5697–5709, 2025.

[2] Z. Chen, J. Zhang, Z. Lai, G. Zhu, Z. Liu, J. Chen, and J. Li, "The devil is in the crack orientation: A new perspective for crack detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 6653–6663.

[3] Z. Chen, Z. Lai, J. Chen, and J. Li, "Mind marginal non-crack regions: Clustering-inspired representation learning for crack segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 12 698–12 708.

[4] Z. Li, J. Liao, C. Tang, H. Zhang, Y. Li, Y. Bian, X. Sheng, X. Feng, Y. Li, C. Gao *et al.*, "Ustc-td: A test dataset and benchmark for image and video coding in 2020s," *IEEE Transactions on Multimedia*, 2025.

[5] R. Zhao, R. Xiong, J. Zhang, Z. Yu, S. Zhu, L. Ma, and T. Huang, "Spike camera image reconstruction using deep spiking neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 6, pp. 5207–5212, 2023.

[6] R. Zhao, R. Xiong, Z. Ding, X. Fan, J. Zhang, and T. Huang, "Mrdflow: Unsupervised optical flow estimation network with multi-scale recurrent decoder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4639–4652, 2021.

[7] Z. Xiao, Z. Li, and W. Jia, "Occlusion-embedded hybrid transformer for light field super-resolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 8, 2025, pp. 8700–8708.

[8] Z. Xiao and X. Wang, "Event-based video super-resolution via state space models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 12 564–12 574.

[9] W. Li, P. Wang, R. Xiong, and X. Fan, "Spiking tucker fusion transformer for audio-visual zero-shot learning," *IEEE Transactions on Image Processing*, vol. 33, pp. 4840–4852, 2024.

[10] W. Li, Z. Ma, L.-J. Deng, X. Fan, and Y. Tian, "Neuron-based spiking transmission and reasoning network for robust image-text retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 7, pp. 3516–3528, 2023.

[11] W. Li, P. Wang, X. Wang, W. Zuo, X. Fan, and Y. Tian, "Multi-timescale motion-decoupled spiking transformer for audio-visual zero-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 11, pp. 10 772–10 786, 2025.

[12] W. Li, R. Xiong, and X. Fan, "Multi-layer probabilistic association reasoning network for image-text retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2024.

[13] W. Li, X.-L. Zhao, Z. Ma, X. Wang, X. Fan, and Y. Tian, "Motion-decoupled spiking transformer for audio-visual zero-shot learning," ser. MM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 3994–4002. [Online]. Available: https://doi.org/10.1145/3581783.3611759

[14] W. Li, Z. Ma, L.-J. Deng, P. Wang, J. Shi, and X. Fan, "Reservoir computing transformer for image-text retrieval," in *Proceedings of the 31st ACM International Conference on Multimedia*, ser. MM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 5605–5613. [Online]. Available: https://doi.org/10.1145/3581783.3611758

[15] H. Bai, Z. Zhao, J. Zhang, Y. Wu, L. Deng, Y. Cui, B. Jiang, and S. Xu, "Refusion: Learning image fusion from reconstruction with learnable loss via meta-learning," *International Journal of Computer Vision*, pp. 1–21, 2024.

[16] H. Bai, J. Zhang, Z. Zhao, Y. Wu, L. Deng, Y. Cui, T. Feng, and S. Xu, "Task-driven image fusion with learnable fusion loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 7457–7468.

[17] X. Zhang, J. Ma, G. Wang, Q. Zhang, H. Zhang, and L. Zhang, "Perceive-ir: Learning to perceive degradation better for all-in-one image restoration," *IEEE Transactions on Image Processing*, pp. 1–1, 2025.

[18] X. Zhang, H. Zhang, G. Wang, Q. Zhang, L. Zhang, and B. Du, "Uniuir: Considering underwater image restoration as an all-in-one learner," *IEEE Transactions on Image Processing*, vol. 34, pp. 6963–6977, 2025.

[19] A. Mitra, S. Biswas, and C. Bhattacharyya, "Bayesian modeling of temporal coherence in videos for entity discovery and summarization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 3, pp. 430–443, 2016.

[20] M. Sun, A. Farhadi, B. Taskar, and S. Seitz, "Summarizing unconstrained videos using salient montages," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2256–2269, 2016.

[21] B. Zhao, X. Li, and X. Lu, "Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7405–7414.

[22] Z. Wei, B. Wang, M. Hoai, J. Zhang, X. Shen, Z. Lin, R. Měch, and D. Samaras, "Sequence-to-segments networks for detecting segments in videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 3, pp. 1009–1021, 2019.

[23] M. Jia, Y. Wei, X. Song, T. Sun, M. Zhang, and L. Nie, "Query-oriented micro-video summarization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 6, pp. 4174–4187, 2024.

[24] M. Narasimhan, A. Rohrbach, and T. Darrell, "Clip-it! language-guided video summarization," *Advances in neural information processing systems*, vol. 34, pp. 13 988–14 000, 2021.

[25] W. Zhu, J. Lu, J. Li, and J. Zhou, "Dsnet: A flexible detect-to-summarize network for video summarization," *IEEE Transactions on Image Processing*, vol. 30, pp. 948–962, 2020.

[26] B. Zhao, H. Li, X. Lu, and X. Li, "Reconstructive sequence-graph network for video summarization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2793–2801, 2021.

[27] J. Lin, H. Hua, M. Chen, Y. Li, J. Hsiao, C. Ho, and J. Luo, "Videoxum: Cross-modal visual and textural summarization of videos," *IEEE Transactions on Multimedia*, vol. 26, pp. 5548–5560, 2024.

[28] H. Li, Q. Ke, M. Gong, and R. Zhang, "Video joint modelling based on hierarchical transformer for co-summarization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3904–3917, 2022.

[29] X. Li, B. Zhao, and X. Lu, "A general framework for edited video and raw video summarization," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3652–3664, 2017.

[30] T. Hussain, K. Muhammad, W. Ding, J. Lloret, S. W. Baik, and V. H. C. de Albuquerque, "A comprehensive survey of multi-view video summarization," *Pattern Recognition*, vol. 109, p. 107567, 2021.

[31] 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.

[32] Y. Ma, Z. Zhu, Y. Qi, A. Beheshti, Y. Li, L. Qing, and G. Li, "Style-aware two-stage learning framework for video captioning," *Knowledge-Based Systems*, vol. 301, p. 112258, 2024.

[33] M. Tian, G. Li, Y. Qi, S. Wang, Q. Z. Sheng, and Q. Huang, "Rethink video retrieval representation for video captioning," *Pattern Recognition*, vol. 156, p. 110744, 2024.

[34] Y. Ma, L. Qing, G. Li, Y. Qi, A. Beheshti, Q. Z. Sheng, and Q. Huang, "Retta: Retrieval-enhanced test-time adaptation for zero-shot video captioning," *Pattern Recognition*, vol. 171, p. 112170, 2026.

[35] Y. Hong, C. Rodriguez-Opazo, Y. Qi, Q. Wu, and S. Gould, "Language and visual entity relationship graph for agent navigation," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[36] N. Shabani, A. Beheshti, Y. Qi, V. Haghighi, S. Moradizeyveh, and Q. Z. Sheng, "Trffc: Efficient traffic forecasting through adaptive spatio-temporal graph reduction," in *Companion Proceedings of the ACM on Web Conference*, 2025, p. 2899–2902.

[37] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., 2014, pp. 505–520.

[38] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. MIT Press, 2014, p. 2069–2077.

[39] B. Zhao, X. Li, and X. Lu, "Tth-rnn: Tensor-train hierarchical recurrent neural network for video summarization," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3629–3637, 2021.

[40] K. Zhang, K. Grauman, and F. Sha, "Retrospective encoders for video summarization," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., 2018, pp. 391–408.

[41] Y. Chen, M. Rohrbach, Z. Yan, S. Yan, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 433–442, 2018.

[42] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional module for temporal action localization in videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6209–6223, 2022.

[43] J. M. R. Chaves and S. Tripathi, "Videosage: Video summarization with graph representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2527–2534.

[44] M. Rochan, L. Ye, and Y. Wang, "Video summarization using fully convolutional sequence networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 347–363.

[45] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1346–1353.

[46] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using delaunay clustering," *International Journal on Digital Libraries*, vol. 6, pp. 219–232, 2006.

[47] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1600–1607.

[48] J. Park, J. Lee, I.-J. Kim, and K. Sohn, "Sumgraph: Video summarization via recursive graph modeling," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*. Springer, 2020, pp. 647–663.

[49] S.-H. Zhong, J. Lin, J. Lu, A. Fares, and T. Ren, "Deep semantic and attentive network for unsupervised video summarization," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 18, no. 2, 2022.

[50] A. Sharghi, B. Gong, and M. Shah, "Query-focused extractive video summarization," in *Computer Vision – ECCV 2016*, 2016, pp. 3–19.

[51] A. Sharghi, J. S. Laurel, and B. Gong, "Query-focused video summarization: Dataset, evaluation, and a memory network based approach," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2127–2136.

[52] A. Kanehira, L. Van Gool, Y. Ushiku, and T. Harada, "Viewpoint-aware video summarization," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7435–7444.

[53] D. Kondratyuk, L. Yu, X. Gu, J. Lezama, J. Huang, G. Schindler, R. Hornung, V. Birodkar, J. Yan, M.-C. Chiu *et al.*, "Videopoet: A large language model for zero-shot video generation," *arXiv preprint arXiv:2312.14125*, 2023.

[54] B. Lin, Y. Ye, B. Zhu, J. Cui, M. Ning, P. Jin, and L. Yuan, "Video-llava: Learning united visual representation by alignment before projection," *arXiv preprint arXiv:2311.10122*, 2023.

[55] G. Chen, Y.-D. Zheng, J. Wang, J. Xu, Y. Huang, J. Pan, Y. Wang, Y. Wang, Y. Qiao, T. Lu *et al.*, "Videollm: Modeling video sequence with large language models," *arXiv preprint arXiv:2305.13292*, 2023.

[56] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao, "Videochat: Chat-centric video understanding," *arXiv preprint arXiv:2305.06355*, 2023.

[57] H. Fei, S. Wu, M. Zhang, M. Zhang, T.-S. Chua, and S. Yan, "Enhancing video-language representations with structural spatio-temporal alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 7701–7719, 2024.

[58] H. Fei, S. Wu, W. Ji, H. Zhang, M. Zhang, M.-L. Lee, and W. Hsu, "Video-of-thought: Step-by-step video reasoning from perception to cognition," in *Proceedings of the 41st International Conference on Machine Learning*, vol. 235. PMLR, 21–27 Jul 2024, pp. 13 109–13 125.

[59] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan, "Video-chatgpt: Towards detailed video understanding via large vision and language models," *arXiv preprint arXiv:2306.05424*, 2023.

[60] L. Qian, J. Li, Y. Wu, Y. Ye, H. Fei, T.-S. Chua, Y. Zhuang, and S. Tang, "Momentor: Advancing video large language model with fine-grained temporal reasoning," *arXiv preprint arXiv:2402.11435*, 2024.

[61] Z. Wang, L. Wang, Z. Zhao, M. Wu, C. Lyu, H. Li, D. Cai, L. Zhou, S. Shi, and Z. Tu, "Gpt4video: A unified multimodal large language model for lnstruction-followed understanding and safety-aware generation," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 3907–3916.

[62] H. Hua, Y. Tang, C. Xu, and J. Luo, "V2xum-llm: Cross-modal video summarization with temporal prompt instruction tuning," *arXiv preprint arXiv:2404.12353*, 2024.

[63] H. Wang, B. Zhou, Z. Zhang, Y. Du, D. Ho, and K.-F. Wong, "M3sum: A novel unsupervised language-guided video summarization," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 4140–4144.

[64] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[65] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[66] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[67] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[68] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[69] H. P. Sajjad, A. Docherty, and Y. Tyshetskiy, "Efficient representation learning using random walks for dynamic graphs," *arXiv preprint arXiv:1901.01346*, 2019.

[70] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[71] T. Cai, S. Luo, K. Xu, D. He, T.-y. Liu, and L. Wang, "Graphnorm: A principled approach to accelerating graph neural network train-

ing," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1204–1215.
[72] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
[73] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007.
[74] M. Gygli, H. Grabner, H. Riemenschneider, and L. V. Gool, "Creating summaries from user videos," in *European conference on computer vision*. Springer, 2014, pp. 505–520.
[75] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "Tvsum: Summarizing web videos using titles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5179–5187.
[76] R. Akhare and S. Shinde, "Query focused video summarization: A review," in *International Symposium on Artificial Intelligence*. Springer, 2022, pp. 202–212.
[77] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International conference on machine learning*. PMLR, 2022, pp. 12 888–12 900.
[78] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
[79] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*. Springer, 2014, pp. 540–555.
[80] D. Pisinger, "Core problems in knapsack algorithms," *Operations Research*, vol. 47, no. 4, pp. 570–575, 1999.
[81] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1059–1067.
[82] M. Rochan, L. Ye, and Y. Wang, "Video summarization using fully convolutional sequence networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 347–363.
[83] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*. Springer, 2016, pp. 766–782.
[84] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.
[85] B. Zhao, X. Li, and X. Lu, "Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7405–7414.
[86] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," *Advances in neural information processing systems*, vol. 27, 2014.
[87] A. Sharghi, B. Gong, and M. Shah, "Query-focused extractive video summarization," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer, 2016, pp. 3–19.
[88] A. Sharghi, J. S. Laurel, and B. Gong, "Query-focused video summarization: Dataset, evaluation, and a memory network based approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4788–4797.
[89] Y. Zhang, M. Kampffmeyer, X. Liang, M. Tan, and E. P. Xing, "Query-conditioned three-player adversarial network for video summarization," *arXiv preprint arXiv:1807.06677*, 2018.
[90] S. Xiao, Z. Zhao, Z. Zhang, X. Yan, and M. Yang, "Convolutional hierarchical attention network for query-focused video summarization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 426–12 433.
[91] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras, "Combining global and local attention with positional encoding for video summarization," in *2021 IEEE international symposium on multimedia (ISM)*. IEEE, 2021, pp. 226–234.
[92] B. He, J. Wang, J. Qiu, T. Bui, A. Shrivastava, and Z. Wang, "Align and attend: Multimodal summarization with dual contrastive losses," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 14 867–14 878.

Wenrui Li received the B.S. degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2021. He is currently pursuing a Ph.D. degree in the School of Computer Science at Harbin Institute of Technology (HIT), Harbin, China. His research interests include multimedia search, joint source-channel coding, and spiking neural networks. He was supported by the National Natural Science Foundation of China (NSFC) Youth Student Basic Research Program (Doctoral Student) in 2024. He has authored or co-authored more than 30 technical articles in refereed international journals and conferences.

Wei Han is currently working toward the B.S. degree from the School of Computer Science, Harbin Institute of Technology (HIT), Harbin, China. His research interests include video summarization, text-3D retrieval, reinforcement learning, and LLM agent.

Hengyu Man received the B.S. degree in communication engineering from Beijing Jiaotong University, Beijing, China, in 2018, the M.S. degree in information and network engineering from KTH Royal Institute of Technology, Stockholm, Sweden, in 2019, and the Ph.D. degree in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 2024. From 2021 to 2023, he was with Peng Cheng Laboratory, Shenzhen, China. He is currently an Associate Researcher with the Faculty of Computing, HIT. His research interests include data compression, image/video coding, and generative computer vision.

Wangmeng Zuo (Senior Member, IEEE) received the Ph.D. degree in computer application technology from Harbin Institute of Technology, Harbin, China, in 2007. He is currently a Professor with the Faculty of Computing, Harbin Institute of Technology. He has published over 200 papers in top tier academic journals and conferences. His current research interests include low level vision, image/video generation, and multimodal understanding. He served as an Associate Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, and SCIENCE CHINA Information Sciences.

Xiaopeng Fan (Senior Member, IEEE) received the B.S. and M.S. degrees from the Harbin Institute of Technology (HIT), Harbin, China, in 2001 and 2003, respectively, and the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong, in 2009. In 2009, he joined HIT, where he is currently a Professor. From 2003 to 2005, he was with Intel Corporation, China, as a Software Engineer. From 2011 to 2012, he was with Microsoft Research Asia, as a Visiting Researcher. From 2015 to 2016, he was with the Hong Kong University of Science and Technology, as a Research Assistant Professor. He has authored one book and more than 250 articles in refereed journals and conference proceedings. His research interests include video coding and transmission, image processing, and computer vision. He was the Program Chair of PCM2017, Chair of IEEE SGC2015, and Co-Chair of MCSN2015. He was an Associate Editor for IEEE 1857 Standard in 2012. He was the recipient of Outstanding Contributions to the Development of IEEE Standard 1857 by IEEE in 2013.

Yonghong Tian (S'00-M'06-SM'10-F'22) is currently a Boya Distinguished Professor and the Director of Beijing Key Laboratory of Spiking Large Models at the School of Computer Science, and the Vice Chancellor of Shenzhen Graduate School and the Executive Dean of School of AI for Science, Peking University, China. He is also the Deputy Director of Intelligence Supercomputing Division, PengCheng Laboratory, Shenzhen, China. His research interests include neuromorphic computing, distributed machine learning and AI for Science. He is the author or co-author of over 400 technical articles in refereed journals and conferences such as Nature Machine Intelligence, Nature Computational Science, Nature Communications, Science Advances, IEEE TPAMI, IJCV, ICML, NeuIPS, etc. Prof. Tian was a Senior Associate Editor of IEEE TCSVT (2024.1-now), IEEE TMM (2014.8-2018.8), IEEE Multimedia Mag. (2018.1-2022.8). He co-initiated IEEE Int'l Conf. on Multimedia Big Data (BigMM), and served as the TPC Member of more than ten conferences such as CVPR, ICCV, ACM KDD, AAAI, ACM MM and ECCV. He was the recipient of the Chinese National Science Foundation for Distinguished Young Scholars in 2018 and 2024, two National Science and Technology Awards and five ministerial-level awards in China, the 2022 IEEE SA Standards Medallion and SA Emerging Technology Award, and the 2025 IEEE Hans Karlsson Award. He is a Fellow of IEEE, a senior member of CIE and CCF, a member of ACM.