ProPy: Building Interactive Prompt Pyramids upon CLIP for Partially Relevant Video Retrieval

Yi Pan[1,2] **Yujia Zhang**[1*] **Michael Kampffmeyer**[3] **Xiaoguang Zhao**[1]
[1] State Key Laboratory of Multimodal Artificial Intelligence Systems,
Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] Department of Physics and Technology, UiT The Arctic University of Norway
{panyi2022,zhangyujia2014,xiaoguang.zhao}@ia.ac.cn, michael.c.kampffmeyer@uit.no

Abstract

Partially Relevant Video Retrieval (PRVR) is a practical yet challenging task that involves retrieving videos based on queries relevant to only specific segments. While existing works follow the paradigm of developing models to process unimodal features, powerful pretrained vision-language models like CLIP remain underexplored in this field. To bridge this gap, we propose **ProPy**, a model with systematic architectural adaption of CLIP specifically designed for PRVR. Drawing insights from the semantic relevance of multi-granularity events, ProPy introduces two key innovations: (1) A **Prompt Pyramid** structure that organizes event prompts to capture semantics at multiple granularity levels, and (2) An **Ancestor-Descendant Interaction Mechanism** built on the pyramid that enables dynamic semantic interaction among events. With these designs, ProPy achieves SOTA performance on three public datasets, outperforming previous models by significant margins. Code is available at https://github.com/BUAAPY/ProPy.

1 Introduction

Partially Relevant Video Retrieval (PRVR) (Dong et al., 2022; Wang et al., 2024b) is a challenging task that retrieves videos based on queries relevant to only specific segments. Unlike traditional Text-to-Video Retrieval (T2VR) (Gabeur et al., 2020; Luo et al., 2022), which requires the query to match the entire video, PRVR better aligns with real-world scenarios – where long videos often consist of multiple events, and users may only be interested in specific segments. This makes PRVR more practical and promising for real-world applications.

Despite significant progress, most PRVR methods (Dong et al., 2022; Wang et al., 2024b; Dong et al., 2023; Jiang et al., 2023; Wang et al., 2024a; Li et al., 2025) follow the paradigm of developing models to process extracted unimodal features.

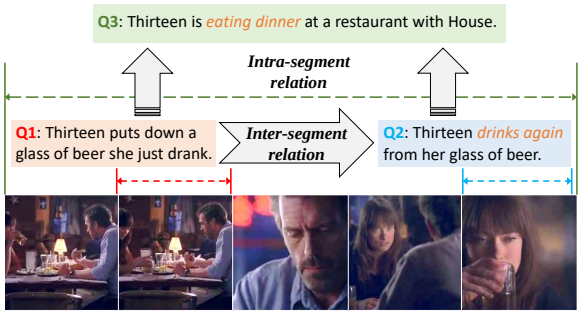

Figure 1: Intra-segment relations and Inter-segment relations. The semantic understanding of Q2's 'drink again' depends on contextual information from previous segments. Meanwhile, the high-level action 'eating dinner' (relevant to Q3) is composed of lower-level, intra-segment events that correspond to Q1 and Q2.

While pretrained vision-language models such as CLIP (Radford et al., 2021) have shown remarkable success in T2VR (Luo et al., 2022), their potential remains underexplored for PRVR. A recent work QASIR (Nishimura et al., 2023) introduces adapters on top of CLIP to process super-image features. However, this approach does not involve in-depth structural adaptions, leaving CLIP's capabilities not fully exploited. To bridge this gap, we propose a model with systematic architectural adaption of CLIP specifically designed for PRVR. Our approach builds upon recent prompt-based T2VR methods (Yang et al., 2024; Zhang et al., 2024; Liu et al., 2025), which demonstrate both effectiveness and efficiency by aggregating video semantics through an explicit global token. However, PRVR presents unique challenges: videos must be modeled as compositions of multiple events rather than encoded as single vector representations. A naive extension of T2VR approaches – treating each segment as an independent sub-video – fails to capture the rich semantic relationships between events. Concretely, there are two fundamental types of event relationships that are crucial to model, as illustrated in Figure 1: **intra-segment**

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 519–533
November 4-9, 2025 ©2025 Association for Computational Linguistics

relations representing compositional semantics between events with hierarchical inclusion relationships, and **inter-segment relations** capturing contextual dependencies between temporally distinct events. The former ones are important in comprehending long events (*eat dinner*) composed of multiple sub-events (*drink*), while the latter ones are crucial in scenes where context semantics are required (*drink again*). Modeling these relations and ensuring semantic interactions of relevant events are beneficial for a comprehensive video understanding (Fei et al., 2024; Yang et al., 2023).

To effectively model the aforementioned event relations and their semantic interactions, we propose **ProPy** (Interactive **Pro**mpt **Py**ramid), a novel CLIP-based architecture for PRVR. ProPy leverages a set of event prompts focusing on segments of varying granularity and organizes them into a **Prompt Pyramid** based on the lengths and positions of their segments. This hierarchically structures a video into multi-granularity events. To account for the distinction between intra-segment and inter-segment relations, we design an **Ancestor-Descendant Interaction Mechanism**, which facilitates *direct* interactions for intra-segment relations and *indirect* interactions for inter-segment relations. Specifically, an ancestor-descendant relationship is established between two event prompts when their governed segments exhibit an inclusion relationship. Direct interactions are permitted only for intra-segment events, while inter-segment interactions are conducted indirectly through upper-level event prompts. With these carefully designed architectures and mechanisms, ProPy achieves SOTA performance on three challenging datasets, demonstrating the superiority of our method. Overall, our contributions can be summarized as follows:

- We propose ProPy, a novel solution to the PRVR task. To the best of our knowledge, ProPy is the first work that involves systematic architectural designs on pretrained vision-language models in the PRVR field.
- Based on the unique characteristics of PRVR, we design a Prompt Pyramid structure to process events with varying granularity, and an Ancestor-Descendant Interaction Mechanism to ensure sufficient semantic interactions for events with intra-segment and inter-segment relations .
- ProPy achieves SOTA performance with notable improvements on three public datasets, demonstrating its effectiveness and superiority.

2 Related Work

Text-to-Video Retrieval focuses on retrieving videos that fully match given textual queries (Gabeur et al., 2020; Luo et al., 2022; Yang et al., 2024; Huang et al., 2023). Since the introduction of pretrained vision-language models (Li et al., 2022, 2023) like CLIP (Radford et al., 2021) in the image domain, significant research efforts (Jia et al., 2022; Deng et al., 2023; Luo et al., 2022; Yang et al., 2024; Cao et al., 2024; Liu et al., 2025) have been directed toward adapting these models for T2VR. Notably, recent prompt-based methods (Zang et al., 2022; Yang et al., 2024; Liu et al., 2025; Zhang et al., 2024; Huang et al., 2023) have demonstrated competitive performance while maintaining efficiency through the use of only a small number of prompt tokens.

Partially Relevant Video Retrieval addresses the task of retrieving videos based on queries relevant to partial segments (Dong et al., 2022; Wang et al., 2024b). Current PRVR approaches (Dong et al., 2022; Wang et al., 2024b; Dong et al., 2023; Cheng et al., 2024; Jun et al., 2025; Ren et al., 2025; Li et al., 2025) predominantly adopt the Multiple Instance Learning (MIL) paradigm (Waqas et al., 2024), employing coarse-fine two-branch architectures to model multiple events during both training and inference. While some recent works (Song et al., 2025; Moon et al., 2025) have incorporated pretrained vision-language models, they primarily utilize them for basic feature extraction without architectural innovations, or source for feature distillation (Dong et al., 2023; Zhang et al., 2025). QASIR (Nishimura et al., 2023), the most similar work to ours, merely introduces adapters on top of CLIP to process super-image features while leaving the core CLIP layers unchanged. We argue that such surface-level modifications are insufficient to fully leverage CLIP's capabilities for the PRVR task.

3 Methodology

We formally define the PRVR task: Given a set of videos $\mathbb{V} = \{V_1, V_2, ...V_{|\mathbb{V}|}\}$, each video V_i can be represented as a list of N_f frames: $V_i = \{f_1^i, f_2^i, ..., f_{N_f}^i\}$. The PRVR task aims to retrieve videos with queries T^i relevant only to certain segment m_j^i: $V_i = \arg\max_{V \in \mathbb{V}} P(V|T^i)$, where $m_j^i \subseteq V_i$ is a subset with consecutive frames.

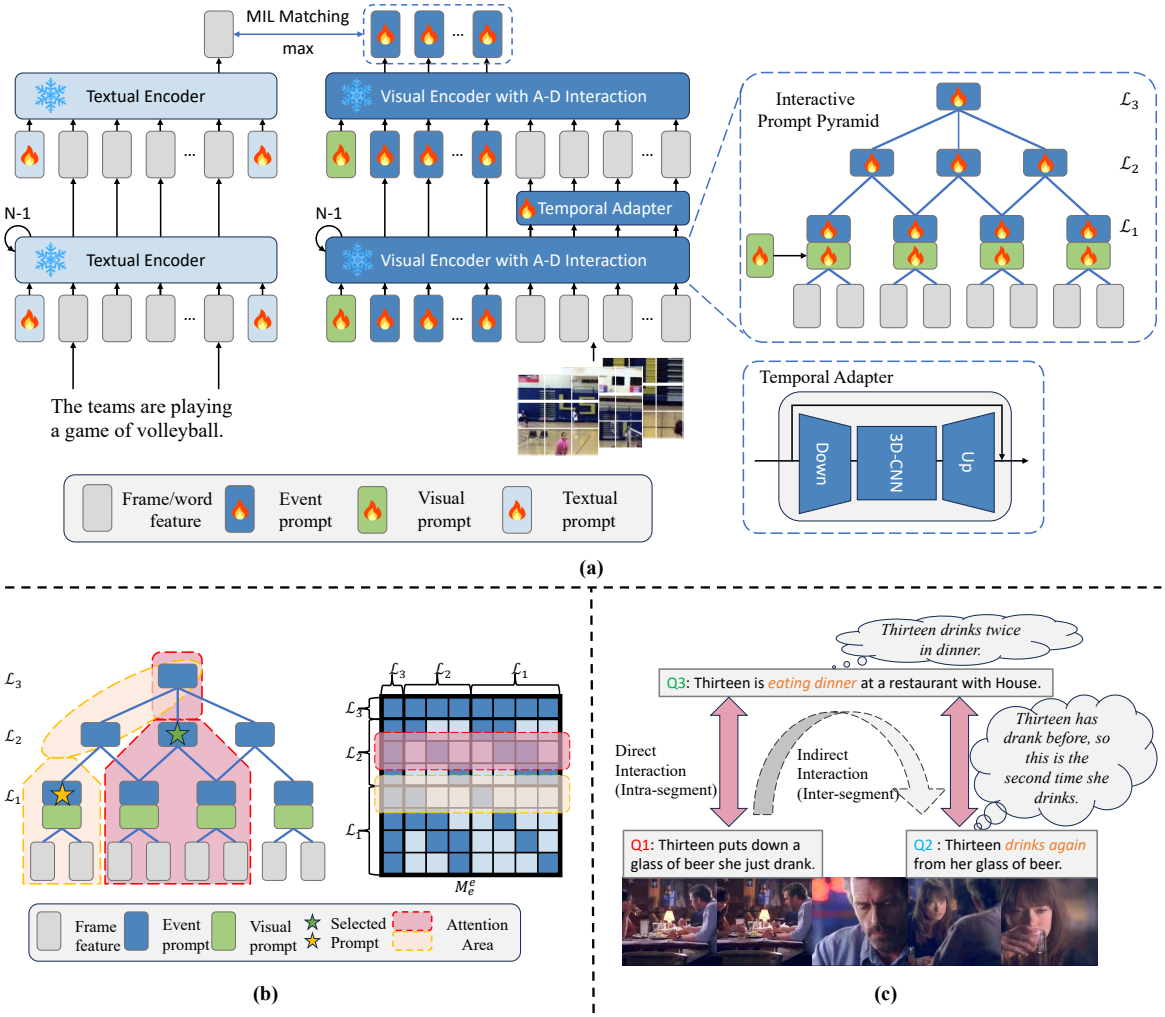

Figure 2: **(a)**: Overview of ProPy. For the visual branch, the event Prompt Pyramid is built upon frame sequence and visual prompts, then event prompts are updated based on the Ancestor-Descendant (A-D) Interaction mechanism. Temporal Adapters are adopted for frame features to strengthen temporal semantics. For the textual branch, prefix and postfix textual prompts are added. We only show 8 frames and a 3-layer pyramid for clarity. **(b)** Details of the Ancestor-Descendant Interaction Mechanism. **Left**: Attention areas of query event prompts. 'selected prompt' means the event prompt served as queries during attention operation. **Right**: Attention mask M_e^e of event prompts. Positions with attention scores are shown in dark blue. **(c)**: An example of direct interactions for intra-segment semantics and indirect interactions for inter-segment semantics.

3.1 Overview of ProPy

As shown in Figure 2 (a), ProPy deeply integrates with CLIP's visual and textual branches. The visual branch utilizes N_e **event prompts** $E \in \mathbb{R}^{N_e \times d_v}$ (where d_v is the ViT dimension) organized in a **Prompt Pyramid** to extract multi-granularity segment features. For the l-th ViT layer, we add N_v **visual prompts** (Yang et al., 2024) $P_l^v \in \mathbb{R}^{N_v \times d_v}$ and a **temporal adapter** (Pan et al., 2022) Ω_l to extract spatial and temporal information. The **Ancestor-Descendant Interaction Mechanism** updates E based on intra/inter-segment relations. The textual branch incorporates N_t **text prompts** $P_l^t \in \mathbb{R}^{N_t \times d}$ (where d is the dimension of CLIP) following DGL (Yang et al., 2024). The model is trained based on Contrastive Learning (Radford et al., 2021) and Multiple Instance Learning (Waqas et al., 2024) paradigms.

3.2 Visual Branch

Prompt Pyramid We first detail the construction of the proposed Prompt Pyramid. Used notations are detailed in Table 1. Given a video V with N_f frames, there are theoretically $N_f \times (N_f + 1)/2$ segments with lengths ranging from 1 to N_f. To save memory, we empirically set $N_f = 2^K, K >$

Notation	Meaning
E	set of all event prompts
$\mathcal{L}_k$	the k-th event prompt layer
e_j^k	the j-th event prompt from $\mathcal{L}_k$
m_j^k	the segment corresponding to e_j^k
n_k	number of event prompts from $\mathcal{L}_k$
c_k	the number of children of e_j^k
o_k	the offset of children between e_j^k and e_{j+1}^k
$\mathcal{A}(e_j^k)$	the ancestors set of e_j^k
$\mathcal{D}(e_j^k)$	the descendants set of e_j^k
$\mathcal{P}(e_j^k)$	the parent set of e_j^k
$\mathcal{C}(e_j^k)$	the children set of e_j^k

Table 1: Notations of prompt pyramid

1, and select segments with lengths of $2^k (1 \le k \le K)$. We then evenly and sparsely sample n_k segments from those of length 2^k, and pair them with hierarchically arranged learnable event prompts:

$$E = \{\mathcal{L}_k = \{e_j^k | 1 \le j \le n_k\} | 1 \le k \le K\} \quad (1)$$

where $\mathcal{L}_k$ is the k-th prompt layer containing n_k event prompts sorted by position. e_j^k corresponds to segment m_j^k of length 2^k. In total, there are $N_e = \sum_{k=1}^{K} n_k$ event prompts.

A prompt pair $(e_{j_1}^{k_1}, e_{j_2}^{k_2})$ forms an **Ancestor-Descendant** (A-D) relation if their governed segments satisfy an inclusion relationship, formally:

$$\begin{aligned} e_{j_1}^{k_1} \in \mathcal{A}(e_{j_2}^{k_2}) &\Leftrightarrow e_{j_2}^{k_2} \in \mathcal{D}(e_{j_1}^{k_1}) \\ &\Leftrightarrow m_{j_2}^{k_2} \subsetneq m_{j_1}^{k_1} \end{aligned} \quad (2)$$

Specially, if $k_1 = k_2 + 1$, then $(e_{j_1}^{k_1}, e_{j_2}^{k_2})$ forms a Parent-Child (P-C) relation, with corresponding sets denoted as $\mathcal{P}(e_{j_2}^{k_2})$ and $\mathcal{C}(e_{j_1}^{k_1})$. To construct a symmetrical pyramid, we set the number of children c_k and the offset of their leftmost children o_k as constants for prompts in the k-th layer, formally:

$$\begin{cases} c_k = |\mathcal{C}(e_{j_1}^k)|, 1 \le j_1 \le n_k \\ \mathscr{L}_k(j_1) = \arg\min_{j_2} \{e_{j_2}^{k-1} | e_{j_2}^{k-1} \in \mathcal{C}(e_{j_1}^k)\} \\ o_k = \mathscr{L}_k(j_1 + 1) - \mathscr{L}_k(j_1), 1 \le j_1 < n_k \end{cases} \quad (3)$$

where $\mathscr{L}_k(j_1)$ is an operation to find index of the leftmost child of $e_{j_1}^k$. c_k and o_k are subject to the following constraint:

$$\frac{n_k - c_{k+1}}{o_{k+1}} + 1 = n_{k+1}, o_{k+1} \mid (n_k - c_{k+1}) \quad (4)$$

This resembles the kernel-stride constraint in CNN (Li et al., 2021), treating c_{k+1} as kernel size and o_{k+1} as stride, but with two differences: 1) No padding is applied. 2) o_{k+1} must divide $n_k - c_{k+1}$ exactly. For the top layer ($k = K$), the offset o_K is set to 1 ($n_{K-1} = c_K, n_K = 1$). Given the frame count N_f, the prompt pyramid is uniquely determined once the hyperparameters $\mathcal{H} = \{(c_k, o_k)\}$ are specified. The impact of structure configuration is discussed in Sec 4.3.

Ancestor-Descendant Interaction Mechanism Next, we describe the update mechanism for event prompts in the visual branch. Given a video $\mathbf{V}$ with N_f frames, CLIP first splits and embeds each frame into sequential features $F \in \mathcal{R}^{N_f \times N_s \times d_v}$, where N_s denotes the sequence length (including the appended [CLS] token). These features are then processed by a ViT with N layers. For the l-th ViT layer, three components participate in the updating process: the event prompts $E_l \in \mathbb{R}^{N_e \times d_v}$, the frame features $F_l \in \mathcal{R}^{N_f \times N_s \times d_v}$ and the per-layer visual prompts $P_l^v \in \mathbb{R}^{N_v \times d_v}$, which guide the spatial-temporal attention between event prompts and frame features.

We update E_l using ViT's attention layer while keeping its weights frozen. For clarity, we first describe the update process (Figure 2 (b)) for a single event prompt e_l^k (from the k-th layer; we omit the prompt index j for simplicity), then generalize to parallel computation. The update consists of attention operations on three components: Firstly, e_l^k attends to frames within its governed segments, denoted as $F_l(e_l^k) \in \mathbb{R}^{2^k \times N_s \times d_v}$, to produce segment features. Secondly, the prompt *directly* interacts with its complete hierarchical context, including its ancestors, descendants and itself, denoted as $E_l(e_l^k)$:

$$E_l(e_l^k) = \mathcal{A}(e_l^k) \cup \mathcal{D}(e_l^k) \cup \{e_l^k\} \quad (5)$$

Thirdly, e_l^k incorporates visual prompts P_l^v to capture spatial-temporal semantics. To preserve the structure information, we replicate P_l^v for n_1 times (n_1 is the number of event prompts from the bottom prompt layer $\mathcal{L}_1$), resulting in $\widetilde{P}_l^v \in \mathbb{R}^{n_1 \times N_v \times d_v}$. These augmented visual prompts correspond one-to-one with the bottom prompt layer $\mathcal{L}_1$. Given e_l^k, we first refer to its descendant prompts in $\mathcal{L}_1$ (or itself if $l = 1$), then incorporate all corresponding visual prompts, denoted as $\widetilde{P}_l^v(e_l^k)$. In this way, for any prompt pairs with an Ancestor-Descendant relation, their visual prompts also exhibit an inclusion relation. These components serve as keys and

values for the attention operation:

$$\begin{cases} K/V(e_l^k) = [F_l(m^k), E_l(e_l^k), \widetilde{P}_l^v(e_l^k)] \\ e_{l+1}^k = Attn(e_l^k, K(e_l^k), V(e_l^k)) \end{cases} \tag{6}$$

where $[\cdot,\cdot,\cdot]$ denotes the concatenation operation on the first dimension, and $F_l(m^k)$, $\widetilde{P}_l^v(e_l^k)$ are flattened into 2d tensor before concatenation. In practice, we use three attention masks to realize parallel computation, formally:

$$\begin{cases} K_l = V_l = [F_l, E_l, \widetilde{P}_l^v] \\ M = [M_f^e, M_e^e, M_v^e] \\ E_{l+1} = Attn(E_l, K_l, V_l, mask = M) \end{cases} \tag{7}$$

where $M_f^e \in \mathbb{R}^{N_e \times (N_f \times N_s)}$, $M_e^e \in \mathbb{R}^{N_e \times N_e}$, $M_v^e \in \mathbb{R}^{N_e \times (n_1 \times N_v)}$ are attention masks regarding frames, event prompts and visual prompts, respectively. A fast construction algorithm for these three masks is detailed in Appendix B.

Note that an event prompt can only access frames from its own segments, and other high-level semantics are exchanged only through other event prompts. This helps to prohibit feature leakage while preserving semantic interactions. Any two event prompts are guaranteed to share at least one common prompt (e.g, the global prompt from the top layer) for direct interaction, offering an indirect communication channel to exchange inter-segment semantics for them, as shown in Figure 2 (c). This design ensures all event prompts are interconnected, with closely positioned events maintaining denser interaction pathways while distant events exhibit sparser connections, naturally mirroring both intra-segment and inter-segment relationships.

Frame Feature Update Previous T2VR works (Yang et al., 2024; Zhang et al., 2024) incorporate global prompts in frame-wise attention to capture temporal semantics. However, for PRVR, we find this approach ineffective, even underperforming models without frame feature updates (Section 4.3). We attribute this to the inherent uncertainty in MIL training that creates unstable information pathways. Instead, we adopt a more stable approach using adapters (Pan et al., 2022) Ω_l to mine temporal semantics directly from frame features, independent of event prompts. The temporal adapter Ω_l is composed of a down-projection, a 3d-CNN, and an up-projection. In detail, given the frame features $F_l \in \mathcal{R}^{N_f \times N_s \times d_v}$, where N_f is the number of frames, and $N_s = H \times W + 1$ is the length of flattened patch tokens with the [CLS] token, the temporal adapter only operates on patch tokens. Features are resized to 2d shapes before the CNN, then back to 1d sequences before the up-projection:

$$\begin{cases} \widetilde{F}_l = F_l[:, 1:, :] \in \mathbb{R}^{N_f \times (H \times W) \times d_v} \\ F_l^{down} = Down_l(\widetilde{F}_l) \in \mathbb{R}^{N_f \times H \times W \times (d_v//2)} \\ F_l^{temp} = CNN_l(F_l^{down}) \in \mathbb{R}^{N_f \times H \times W \times (d_v//2)} \\ F_l^{up} = Up_l(F_l^{temp}) \in \mathbb{R}^{N_f \times (H \times W) \times d_v} \\ F_{l+1}[:, 1:, :] = [F_l[:, 0, :], F_l[:, 1:, :] + F_l^{up}] \end{cases} \tag{8}$$

The output event prompts from the last layer are projected to d dimension to represent multi-granularity event features, denoted as $\widetilde{E} \in \mathbb{R}^{N_e \times d}$.

3.3 Textual Branch

The textual branch builds upon DGL (Yang et al., 2024). To enhance multimodal alignment, two projection layers are utilized to project visual prompts P_l^v to prefix and postfix prompts. These are concatenated with word features for updating:

$$\begin{cases} P_l^{pre/post} = f_{pre/post}(P_l^v) \in \mathbb{R}^{(Nt/2) \times d} \\ [\_\_, T_{l+1}, \_\_] = L_l^t([P_{l,pre}^t, T_l, P_{l,post}^t]) \end{cases} \tag{9}$$

where L_l^t is the l-th layer of the textual branch, $P_l^{pre/post}$ are prefix and postfix prompts, $f_{pre/post}$ are projection layers, T_l are input word features. The query representation $\widetilde{T} \in \mathbb{R}^d$ is obtained from the features of the last word in the final layer.

3.4 Training Objective

Following the MIL paradigm, the highest similarity score between the query $\widetilde{T}$ and event prompts $\widetilde{E}$ is selected:

$$S(T, V) = \max_e \{cos(\widetilde{T}, \widetilde{E})\} \tag{10}$$

The alignment is conducted with pair-wise similarities based on the symmetric InfoNCE loss (Chen et al., 2020; Radford et al., 2021).

4 Experiments

4.1 Experimental Settings

Datasets We evaluate ProPy on four public datasets: TVR (Lei et al., 2020), ActivityNet-Captions (Krishna et al., 2017), Charades-STA (Gao et al., 2017) and QVHighlights (Lei et al., 2021). TVR comprises around 21.8K videos, each paired with 5 descriptions. ActivityNet-Captions

Table 2: Performance comparison on TVR, ActivityNet Captions and Charades-STA dataset. Rows highlighted in gray represent original performance of methods leveraging ResNet152 + I3D + Roberta features. The best, second and third performance are marked in **bold**, underline and wave , respectively .

Method	TVR					ActivityNet Captions					Charades-STA				
	R@1	R@5	R@10	R@100	SumR	R@1	R@5	R@10	R@100	SumR	R@1	R@5	R@10	R@100	SumR
MS-SL	13.5	32.1	43.4	83.4	172.4	7.1	22.5	34.7	75.8	140.1	1.8	7.1	11.8	47.7	68.4
PEAN	13.5	32.8	44.1	83.9	174.2	7.4	23.0	35.5	75.9	141.8	**2.7**	8.1	13.5	50.3	74.7
GMMFormer	13.9	33.3	44.5	84.9	176.6	8.3	24.9	36.7	76.1	146.0	2.1	7.8	12.5	50.6	72.9
DL-DKD	14.4	34.9	45.8	84.9	179.9	8.0	25.0	37.5	77.1	147.6	-	-	-	-	-
Proto	15.4	35.9	47.5	86.4	185.1	7.9	24.9	37.2	77.3	147.4	-	-	-	-	-
ARL	15.6	36.3	47.7	86.3	185.9	8.3	24.6	37.4	78.0	148.3	-	-	-	-	-
GMMFormer-V2	16.2	37.6	48.8	86.4	189.1	8.9	27.1	40.2	78.7	154.9	2.5	8.6	13.9	**53.2**	**78.2**
DGL-MIL	17.5	38.2	51.1	87.5	194.3	10.5	26.4	40.5	77.4	154.8	1.4	5.3	9.2	40.6	56.5
MS-SL	17.2	39.1	51.5	87.4	195.2	9.4	26.1	37.9	77.2	150.6	1.3	4.6	8.2	38.5	52.6(↓15.8)
QASIR	19.0	39.9	50.4	87.2	196.5	14.1	32.9	44.5	79.9	171.4	1.9	5.8	10.1	40.0	57.8
GMMFormer	18.4	39.5	50.8	89.2	197.9	9.4	26.4	38.2	76.2	150.2	0.9	4.5	8.0	39.0	52.4(↓20.5)
GMMFormer-V2	19.2	40.1	50.5	**90.3**	200.1	10.8	28.8	41.1	78.1	158.8	1.3	4.7	9.0	40.4	55.4(↓22.8)
AMDNet	19.7	42.4	54.1	88.9	205.1	12.3	32.5	45.9	82.1	172.8	1.1	4.2	7.2	36.4	48.9
Propy	**22.4**	**45.0**	**55.9**	89.5	**212.8**	**14.9**	**34.9**	**47.5**	**82.7**	**180.0**	2.6	**8.7**	**14.8**	50.4	76.5

Table 3: Performance on QVHighlights *val* split. Rows highlighted in gray are results with CLIP-B/16 features adopted from Proto (Moon et al., 2025).

Model	R@1	R@5	R@10	R@100	SumR
GMMFormer	18.2	43.7	56.7	92.5	211.1
MS-SL	20.4	46.7	60.7	94.6	222.5
Proto	22.6	48.8	61.3	93.9	226.6
GMMFormer	16.3	39.7	52.3	88.4	196.7
AMDNet	17.1	40.8	52.5	88.4	198.8
GMMFormer-V2	15.6	40.2	53.7	88.5	198.0
MS-SL	17.4	43.4	55.2	88.8	204.8
Propy	**37.4**	**65.6**	**76.1**	**96.5**	**275.5**

Table 4: Ablation on pyramid structures.

N_f	$\mathcal{H} = \{(c_k, o_k)\}$	R@5	R@10	R@100
16	{(4,2),(3,2),(3,1)}	8.3	14.1	47.7
16	{(2,1),(3,2),(3,2),(3,1)}	8.5	14.2	48.1
32	{(4,2),(3,2),(3,2),(3,1)}	8.5	14.4	49.3
32	{(2,2),(2,1),(3,2),(3,2),(3,1)}	**8.7**	**14.8**	**50.4**

Table 5: Ablation study on semantic interaction mechanism of event prompts.

Attention Area	Interaction	R@5	R@10	R@100
$\mathcal{A}$	inter-only	8.1	12.6	49.5
$\mathcal{D}$	intra-only	8.4	13.4	48.3
$\mathcal{P}$	inter-only	8.5	13.7	49.1
$\mathcal{C}$	intra-only	8.3	13.3	47.8
$\mathcal{P} \cup \mathcal{C}$	inter-intra	8.6	14.1	49.4
$\mathcal{W}$	unstructured	**8.7**	14.2	48.7
$\mathcal{S}$	none	8.3	14.1	48.6
$\mathcal{A} \cup \mathcal{D}$	inter-intra	**8.7**	**14.8**	**50.4**

(referred to as ActivityNet for simplicity) comprises around 20K YouTube videos, with an average duration of 118 seconds and 3.7 descriptions per video. Charades-STA contains 6.7K videos, annotated with an average of 2.4 descriptions per video. QVHighlights is a dataset for moment retrieval containing over 10K videos. We follow previous works (Dong et al., 2022; Moon et al., 2025) for data partitioning and evaluation metrics.

Implementation Details We select CLIP-B/32 as the backbone. The dimensions d_v, d are set to 768 and 512, respectively. For the default structure hyperparameter, N_f is set to 32 and $\mathcal{H}$ is configured as $\{(2,2),(2,1),(3,2),(3,2),(3,1)\}$. Following DGL (Yang et al., 2024), we use 4 visual prompts (N_v) and 8 textual prompts (N_t) per layer. The learning rates are set to 1e-3 for ActivityNet, 9e-4 for QVHighlights, and 8e-4 for TVR and Charades-STA, with a uniform batch size of 24 across all datasets. Following (Luo et al., 2022; Yang et al., 2024), ProPy is trained for 10 epochs using the AdamW optimizer. All experiments are conducted on a single NVIDIA RTX 3090 GPU.

Baselines ProPy is the first prompt-based model built on CLIP for PRVR, and no existing methods adopt a similar architecture. To ensure a comprehensive comparison, we evaluate 3 types of baseline models. (1) **DGL-MIL** We adapt our base framework, DGL (Yang et al., 2024), to fit the MIL paradigm. Specifically, we expand the number of global prompts to N_e and use the entire set for alignment. (2) **QASIR** (Nishimura et al., 2023). A CLIP-based model that employs super-image construction for feature enhancement. (3) **Other PRVR models** (Dong et al., 2022, 2023; Wang et al., 2024a,b; Jiang et al., 2023; Song et al., 2025; Moon et al., 2025; Cho et al., 2025). We run prior open-source PRVR models using extracted CLIP-B/32 features for fair comparison. Additionally, we include their original performance (trained with ResNet152 (He et al., 2016)+I3D (Carreira and Zisserman, 2017)+RoBERTa (Liu et al., 2019) features) as a reference. Further implementation details are provided in Appendix A.

4.2 Overall Comparison

The performance comparison is shown in Table 2, 3. As shown in Table 2, ProPy significantly outperforms all baselines on TVR and ActivityNet, achieving absolute improvents of 7.7% on TVR

and 7.2% on ActivityNet. These gains can be partially attributed to CLIP's well-aligned multimodal features, as evidenced by the improved performance of other PRVR models when using CLIP features. However, on Charades-STA, most PRVR models exhibit degraded performance with CLIP features, while ProPy remains competitive. This aligns with observations in prior work (Nishimura et al., 2023), where shorter query lengths in Charades-STA (averaging 6.2 words, vs. 12.2 for ActivityNet and 13.6 for TVR) lead to insufficient textual supervision (Moon et al., 2025), demanding deeper video understanding. In this circumstance, CLIP's static image features (from its last layer) underperform dynamic I3D features from 3D-CNNs. ProPy addresses this limitation by aggregating semantically rich features from *all* CLIP layers and integrated temporal adapters, enabling comprehensive video understanding. On the challenging QVHighlights dataset, ProPy achieves remarkable gains (Table 3) – even surpassing methods with CLIP-B/16 (a more powerful backbone with 16x16 grid size) features – with a 37.4% R@1 score, far exceeding competitors. These results validate ProPy's superiority and broad applicability.

4.3 Analysis

Unless otherwise specified, the following experiments are based on the Charades-STA dataset.

Pyramid Structure Settings We investigate the impact of *width* (N_f) and *depth* (number of layers), as shown in Table 4. Results indicate that increasing either parameter improves performance by providing richer information and more event candidates. However, the memory issue should also be considered in model design, particularly for frame feature memory relevant to parameter N_f.

Ancestor-Descendant Interaction Mechanism We analyze the impact of interaction mechanisms. We evaluate 7 alternative mechanisms: 1) $\mathcal{A}$ attends only to ancestors, missing descendant intra-segment information; 2) $\mathcal{D}$ attends only to descendants, blocking inter-segment communication guided by ancestors; 3) $\mathcal{P}$ attends only to parent nodes, providing limited inter-segment interaction; 4) $\mathcal{C}$ attends only to child nodes, offering partial intra-segment interaction; 5) $\mathcal{P} \cup \mathcal{C}$ combines both parent and child attention; 6) $\mathcal{W}$ attends to all nodes without structure; and 7) $\mathcal{S}$ prohibits any interaction. The results in Table 5 demonstrate that: (1) Semantic interactions are essential, as the non-interactive model $\mathcal{S}$ performs poorly; (2) Interaction structure significantly impacts performance, shown by $\mathcal{W}$'s results; (3) Both intra-segment and inter-segment interactions are important ($\mathcal{A} \cup \mathcal{D}$ is better than $\mathcal{A}$ and $\mathcal{D}$, $\mathcal{P} \cup \mathcal{C}$ is better than $\mathcal{P}$ and $\mathcal{C}$). (4) The Ancestor-Descendant Interaction Mechanism achieves optimal performance by effectively integrating both interaction types.

Table 6: Ablation study on frame feature updating mechanism.

Mechanism	R@1	R@5	R@10	R@100	SumR
attn-pyr	1.5	5.9	9.5	37.1	54.0
attn-whole	1.4	6.4	10.4	37.3	55.5
attn-adapter	1.7	6.6	9.8	40.0	58.1
orig	1.9	6.6	11.3	42.5	62.3
adapter	**2.6**	**8.7**	**14.8**	**50.4**	**76.5**

Table 7: Ablation study on event prompts from different layers for MIL learning.

Levels(k)	R@1	R@5	R@10	R@100	SumR
{1,2,3}	2.4	8.1	13.0	48.8	72.3
{3,4,5}	2.3	8.0	13.2	48.5	72.0
{1,2,3,4}	**2.6**	**9.0**	14.3	49.9	75.8
{2,3,4,5}	2.3	8.3	13.6	49.1	73.3
{1,2,3,4,5}	**2.6**	8.7	**14.8**	**50.4**	**76.5**

Updating Mechanism of Frame Feature We perform studies for the frame feature updating mechanism, evaluating 3 other distinct mechanisms: (1) *orig*: vanilla CLIP processing without trainable components; (2) *attn-whole*: DGL-style attention using *all* event prompts as keys/values; (3) *attn-pyr*: constrained attention where frames only interact with governing event prompts (masked by M_f^e); (4) *attn-adapter*: enhanced version of (3) with per-layer adapters. As shown in Table 6, the significant performance improvement demonstrates the effectiveness of temporal adapters.

Operation on Visual Prompt To validate the operation on visual prompts, we further consider two alternative operations: (1) *no-copy*. All event prompts incorporate N_v visual prompts for updating. The corresponding attention mask M_v^e has the shape of $N_e \times N_v$. (2) $C(E)$. Visual prompts are copied N_e times and assigned to event prompts in a one-to-one manner. M_v^e is in the shape of $N_e \times (N_e \times N_v)$. The original design is denoted as $C(\mathcal{L}_1)$, which copies visual prompts n_1 times. Results are shown in Table 8, showing that the $C(\mathcal{L}_1)$ design achieves the best performance. This design considers the inclusion relation of event prompts, preserving beneficial structure information.

Different Levels of Event Prompts We select different levels of event prompts for MIL learning with the pyramid structure fixed to examine the

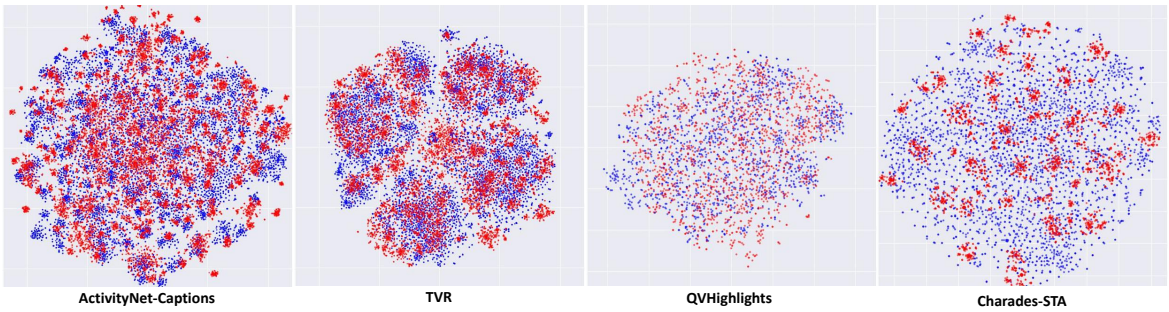

Figure 3: TSNE visualizations. Points in red, blue are text features and segment features, respectively.

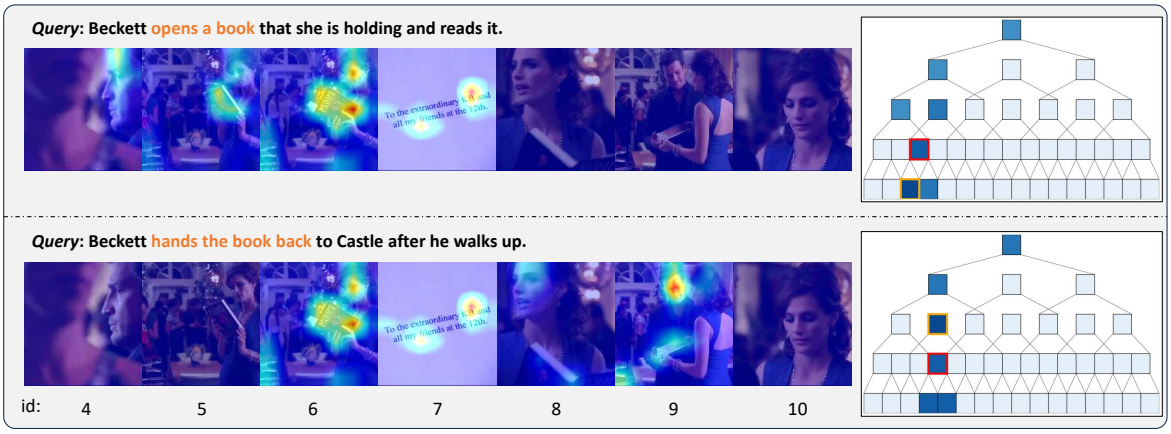

Figure 4: **Left**: Retrieval results of ProPy (samples are selected based on the R@1 metric). The attention maps visualize the scores between selected event prompts and image patches. **Right**: Attention scores between selected event prompts (with red borders) and other prompts. Events with the highest scores are marked by orange borders.

Table 8: Ablation study of operations on visual prompts.

Operation	R@1	R@5	R@10	R@100	SumR
no-copy	2.2	8.2	13.0	48.8	72.2
$C(E)$	2.1	8.1	13.3	48.5	72.0
$C(\mathcal{L}_1)$	**2.6**	**8.7**	**14.8**	**50.4**	**76.5**

Table 9: Ablation on each design. $\mathcal{AD}$, Ω, $C(\mathcal{L}_1)$ mean Ancestor-Descendant Interaction, using adapters and replicating visual prompts n_1 times, respectively. Compared operations are disabling event prompt interaction, vanilla CLIP processing and no replication.

Idx	$\mathcal{AD}$	Ω	$C(\mathcal{L}_1)$	R@1	R@5	R@10	R@100	SumR
a				1.8	5.8	9.3	38.6	55.5
b	✓			1.7	6.8	11.0	41.9	61.4
c		✓		1.5	6.6	10.9	41.5	60.5
d			✓	1.7	5.6	9.4	39.4	56.1
e	✓	✓		2.2	8.2	13.0	48.8	72.2
f	✓		✓	1.9	6.6	11.3	42.5	62.3
g		✓	✓	2.1	8.3	14.1	48.6	73.1
h	✓	✓	✓	**2.6**	**8.7**	**14.8**	**50.4**	**76.5**

roles of different prompt layers. Results in Table 7 show that models with more levels of prompts in learning are superior to those with fewer levels. We also find that layers nearer to the bottom (k=1) contribute more to performance. To uncover the underlying reason, we conduct a statistical experiment on the distribution of selected events during learning, shown in Figure 5. It illustrates that ProPy learns in an easy-to-hard manner: it initially processes shorter segments then progressively extends to longer ones, with learned semantics evolving from low-level to high-level.

Table 10: Performance on the ActivityNet dataset in the Weakly-Supervised VCMR setting. Rows highlighted in gray represent original performance of methods leveraging ResNet152 + I3D + Roberta features.

Method	IoU=0.3		IoU=0.5		IoU=0.7	
	R@10	R@100	R@10	R@100	R@10	R@100
FAWL	11.86	38.98	6.25	21.77	2.88	10.05
JSG	13.27	40.61	8.76	29.98	3.83	15.78
FAWL	23.68	48.02	17.54	43.57	9.35	20.66
JSG	25.62	54.31	19.35	45.15	10.92	26.14
Propy	**28.57**	**57.42**	**20.81**	**46.22**	**12.94**	**31.85**

Table 11: Parameters and inference time (per query) comparison on Charadest-STA. Matching time refers to the time for video-text matching process.

Method	Parameters(MB)		Inference Time(ms)	
	trainable	total	matching	total
MS-SL	4.85	4.85	0.65	3.05
GMMFormer	12.85	12.85	0.44	1.79
ProPy	7.97	159.24	1.16	20.3

Ablation Study We conduct ablation studies on each design. As shown in Table 9, the Ancestor-Descendant Mechanism individually contributes

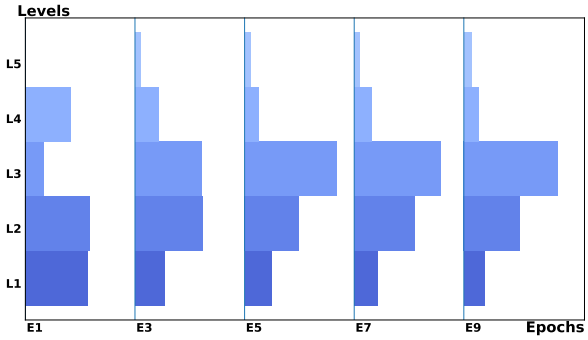

Figure 5: Length distributions of selected events.

the most. When equipped with adapters, the model achieves comparable performance with the Ancestor-Descendant Mechanism or the $C(\mathcal{L}_1)$ operation (e and g). This indicates that structured visual prompts are beneficial for event learning. The full setting, with temporal adapters, Ancestor-Descendant Mechanism and structured visual prompts, achieves the best performance.

Grounding Capability We evaluate ProPy's grounding capability under the Weakly-Supervised Video Corpus Moment Retrieval (Chen et al., 2023) setting. As shown in Table 10, ProPy achieves SOTA performance on ActivityNet thanks to the design of the multi-granularity event pyramid. Notably, compared to previous methods (Chen et al., 2023; Pan et al., 2025), ProPy does not require complex intra-video losses and time-consuming NMS (Lin et al., 2018) operations.

Efficiency Comparison As shown in Table 11, ProPy contains relatively more parameters. However, most parameters are frozen CLIP weights, and there are only 5% trainable parameters. Among these trainable parameters, temporal adapters introduce 7.11M (89%), and prompts only occupy 0.86M (11%). However, as discussed before, incorporating these temporal adapters is crucial for overall performance. ProPy takes longer inference time, yet most time is spent on feature computation through CLIP layers. Once the computation process of features is done, the matching time is comparable with other models. This indicates that in real-world retrieval scenarios, ProPy enjoys high retrieval accuracy without significant increases in latency.

Qualitative Analysis We visualize the t-SNE clustering results in Figure 3. As observed, the cluster distributions vary across datasets, reflecting the differences in their video content and textual annotations. For ActivityNet, which primarily focuses on actions or events, semantically similar actions are located close to each other, resulting in a large number of distinct cluster centers. In TVR, the videos are sourced from 6 different TV shows, where each show features recurring characters and scenes, leading to roughly 4 to 6 clusters. QVHighlights contains videos from news and vlogs, with more diverse visual content and textual descriptions, which results in a more diffuse distribution without clear cluster centers. As previously discussed, the Charades-STA dataset contains short textual queries with limited fine-grained annotations, causing its textual features to collapse around a few dense centers. Except for the Charades dataset, the t-SNE plots of the other datasets generally reveal a reasonable degree of alignment between the video and textual features in the shared feature space.

We further illustrate some retrieval results from TVR in Figure 4. The results show that: (1) ProPy is able to extract high-quality spatial-temporal semantics, such as'opens a book' and 'hands the book back'. (2) ProPy ensures sufficient semantic interactions. For example, the latter event 'hands the book back to Castel' requires the previous context of 'book', 'Castel', and the visualization shows that the selected event gives the highest attention to its parent, which is one of direct information channels for the former event.

5 Conclusion

We propose ProPy, the first in-depth CLIP-based model for PRVR. By considering both intra-event and inter-event relations of video events, we design an Interactive Prompt Pyramid architecture to extract multi-granularity event features and an Ancestor-Descendant Interaction Mechanism to ensure sufficient semantic interactions. Extensive experiments demonstrate the superiority and generalizability of ProPy.

6 Limitations

Though ProPy only requires a small number of trainable parameters, the memory occupied by CLIP features cannot be ignored. Furthermore, the 2^k segment sampling strategy and the structure parameters $\mathcal{H}$ are empirical. Future works will include an adaptive selection method (like (Wang et al., 2024c)) of video frames and pyramid structures.

Acknowledgement

This work was supported by the International Partnership Program of the Chinese Academy of Sciences (Grant No. 104GJHZ2023053FN) and the Young Scientists Fund of the State Key Laboratory of Multimodal Artificial Intelligence Systems (Grant No. ES2P100118).

References

Meng Cao, Haoran Tang, Jinfa Huang, Peng Jin, Can Zhang, Ruyang Liu, Long Chen, Xiaodan Liang, Li Yuan, and Ge Li. 2024. Rap: Efficient text-video retrieval with sparse-and-correlated adapter. *arXiv preprint arXiv:2405.19465*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Zhiguo Chen, Xun Jiang, Xing Xu, Zuo Cao, Yijun Mo, and Heng Tao Shen. 2023. Joint searching and grounding: Multi-granularity video content retrieval. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 975–983.

Dingxin Cheng, Shuhan Kong, Bin Jiang, and Qiang Guo. 2024. Transferable dual multi-granularity semantic excavating for partially relevant video retrieval. *Image and Vision Computing*, 149:105168.

Cheol-Ho Cho, WonJun Moon, Woojin Jun, MinSeok Jung, and Jae-Pil Heo. 2025. Ambiguity-restrained text-video representation learning for partially relevant video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 2500–2508.

Chaorui Deng, Qi Chen, Pengda Qin, Da Chen, and Qi Wu. 2023. Prompt switch: Efficient clip adaptation for text-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15648–15658.

Jianfeng Dong, Xianke Chen, Minsong Zhang, Xun Yang, Shujie Chen, Xirong Li, and Xun Wang. 2022. Partially relevant video retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 246–257.

Jianfeng Dong, Minsong Zhang, Zheng Zhang, Xianke Chen, Daizong Liu, Xiaoye Qu, Xun Wang, and Baolong Liu. 2023. Dual learning with dynamic knowledge distillation for partially relevant video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11302–11312.

Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne Hsu. 2024. Video-of-thought: Step-by-step video reasoning from perception to cognition. *arXiv preprint arXiv:2501.03230*.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*, pages 214–229. Springer.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Siteng Huang, Biao Gong, Yulin Pan, Jianwen Jiang, Yiliang Lv, Yuyuan Li, and Donglin Wang. 2023. Vop: Text-video co-operative prompt tuning for cross-modal retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6565–6574.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer.

Xun Jiang, Zhiguo Chen, Xing Xu, Fumin Shen, Zuo Cao, and Xunliang Cai. 2023. Progressive event alignment network for partial relevant video retrieval. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1973–1978. IEEE.

WooJin Jun, WonJun Moon, Cheol-Ho Cho, MinSeok Jung, and Jae-Pil Heo. 2025. Bridging the semantic granularity gap between text and frame representations for partially relevant video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4166–4174.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *Computer Vision–ECCV 2020:*

16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16, pages 447–463. Springer.

Jun Li, Jinpeng Wang, Chaolei Tan, Niu Lian, Long Chen, Yaowei Wang, Min Zhang, Shu-Tao Xia, and Bin Chen. 2025. Enhancing partially relevant video retrieval with hyperbolic learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019.

Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. 2018. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zichen Liu, Kunlun Xu, Bing Su, Xu Zou, Yuxin Peng, and Jiahuan Zhou. 2025. Stop: Integrated spatial-temporal dynamic prompting for video understanding. *arXiv preprint arXiv:2503.15973*.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2022. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304.

WonJun Moon, Cheol-Ho Cho, Woojin Jun, Minho Shim, Taeoh Kim, Inwoong Lee, Dongyoon Wee, and Jae-Pil Heo. 2025. Prototypes are balanced units for efficient and effective partially relevant video retrieval. *arXiv preprint arXiv:2504.13035*.

Taichi Nishimura, Shota Nakada, and Masayoshi Kondo. 2023. Vision-language models learn super images for efficient partially relevant video retrieval. *ACM Transactions on Multimedia Computing, Communications and Applications*.

Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. 2022. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477.

Yi Pan, Yujia Zhang, and Xiaoguang Zhao. 2025. Fawl: Weakly-supervised video corpus moment retrieval with frame-wise auxiliary alignment and weighted contrastive learning. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Junlong Ren, Gangjian Zhang, Yu Hu, Jian Shu, and Hao Wang. 2025. Exploiting inter-sample correlation and intra-sample redundancy for partially relevant video retrieval. *arXiv preprint arXiv:2504.19637*.

Peipei Song, Long Zhang, Long Lan, Weidong Chen, Dan Guo, Xun Yang, and Meng Wang. 2025. Towards efficient partially relevant video retrieval with active moment discovering. *arXiv preprint arXiv:2504.10920*.

Yuting Wang, Jinpeng Wang, Bin Chen, Tao Dai, Ruisheng Luo, and Shu-Tao Xia. 2024a. Gmmformer v2: An uncertainty-aware framework for partially relevant video retrieval. *arXiv preprint arXiv:2405.13824*.

Yuting Wang, Jinpeng Wang, Bin Chen, Ziyun Zeng, and Shu-Tao Xia. 2024b. Gmmformer: gaussian-mixture-model based transformer for efficient partially relevant video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5767–5775.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. 2024c. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv preprint arXiv:2405.19209*.

Muhammad Waqas, Syed Umaid Ahmed, Muhammad Atif Tahir, Jia Wu, and Rizwan Qureshi. 2024. Exploring multiple instance learning (mil): A brief survey. *Expert Systems with Applications*, page 123893.

Xiangpeng Yang, Linchao Zhu, Xiaohan Wang, and Yi Yang. 2024. Dgl: Dynamic global-local prompt tuning for text-video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6540–6548.

Yang Yang, Yurui Huang, Weili Guo, Baohua Xu, and Dingyin Xia. 2023. Towards global video scene segmentation with context-aware transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 3206–3213.

Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. 2022. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*.

Haonan Zhang, Pengpeng Zeng, Lianli Gao, Jingkuan Song, and Heng Tao Shen. 2024. Mpt: Multi-grained prompt tuning for text-video retrieval. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1206–1214.

Qun Zhang, Chao Yang, Bin Jiang, and Bolin Zhang. 2025. Multi-grained alignment with knowledge distillation for partially relevant video retrieval. *ACM Transactions on Multimedia Computing, Communications and Applications*.

A More Implementation Details

Following (Yang et al., 2024), videos are compressed first to 3fps and 224 × 224 resolution. The model is trained using the AdamW optimizer whose decoupled weight decay is set to 0.2. During training, a warm-up strategy is adopted followed by a cosine learning rate policy. For other PRVR models, we utilize the output sequences from the last layer of CLIP's textual branch as text features, and the [CLS] features from the visual branch's last layer as visual features. To make minimal modifications, the number of frames is set to 128 for other PRVR models, which is much larger than 32 for ProPy. The training process for baseline PRVR models also follows the original process, i.e., 100 max epochs with an early stop strategy.

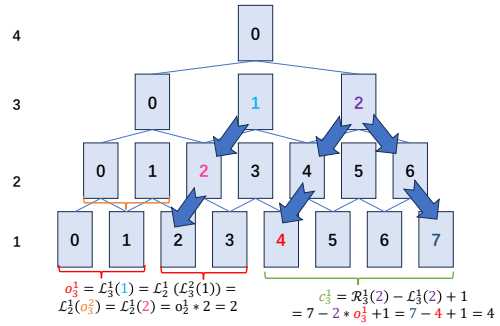

Figure 6: Mathematical relations of ProPy. For each layer, indices start from 0. We focus on the prompt with index 1 and $n_k - 1$ for the k-th layer. The relations of its leftmost and rightmost descendant are transitive.

B Mask Construction Algorithm

The pyramid structure and attention masks are determined by the frame count N_f and the structure hyperparameters $\mathcal{H} = \{(c_k, o_k)\}$ (c_k, o_k are the number and offset of children). We provide one fast algorithm to construct the three attention masks M_e^e, M_f^e and M_v^e used for the Ancestor-Descendant Interaction Mechanism. There are two steps of the algorithm.

Step 1: Cross-layer structure parameters

First, we expand definitions in Equation (3) to cross-layer structure parameters $c_{k_1}^{k_2}$ and $o_{k_1}^{k_2}$ between layer $\mathcal{L}_{k_1}$ and $\mathcal{L}_{k_2}$ ($k_1 > k_2$):

$$\begin{aligned} c_{k_1}^{k_2} &= |\{e_{j_2}^{k_2} | e_{j_2}^{k_2} \in \mathcal{D}(e_{j_1}^{k_1})\}| \\ \mathscr{L}_{k_1}^{k_2}(j_1) &= \arg\min_{j_2} \{e_{j_2}^{k_2} | e_{j_2}^{k_2} \in \mathcal{D}(e_{j_1}^{k_1})\} \\ o_{k_1}^{k_2} &= \mathscr{L}_{k_1}^{k_2}(j_1 + 1) - \mathscr{L}_{k_1}^{k_2}(j_1) \end{aligned} \tag{11}$$

Algorithm 1: Structure parameters

Input: $N_f, \mathcal{H} = \{(c_k, o_k)\}$
Output: $\mathbf{H}$=$\{(k_1, k_2) : (c_{k_1}^{k_2}, o_{k_1}^{k_2})\}$

1 $\mathbf{L} = [N_f]$ `// length of each layer`
2 $K = len(H)$;
3 **for** $k \leftarrow 1$ ***to*** K **do**
4 $\quad \mathbf{H}[(k, k-1)] = (c_k, o_k)$ `// from` $\mathcal{H}$
5 $\quad n_k = (\mathbf{L}[k-1] - c_k)//o_k + 1$;
6 $\quad \mathbf{L}.append(n_k)$;
7 **end**
8 **for** $k_1 \leftarrow K$ ***to*** 1 **do**
9 $\quad$ **for** $k_2 \leftarrow k_1 - 2$ ***to*** 0 **do**
10 $\quad\quad c_{k_1}^{k_2+1}, o_{k_1}^{k_2+1} = \mathbf{H}[(k_1, k_2+1)]$;
11 $\quad\quad c_{k_2+1}^{k_2}, o_{k_2+1}^{k_2} = \mathbf{H}[(k_2+1, k_2)]$;
12 $\quad\quad o_{k_1}^{k_2} = o_{k_1}^{k_2+1} * o_{k_2+1}^{k_2}$;
13 $\quad\quad c_{k_1}^{k_2} = \mathbf{L}[k_2] - o_{k_1}^{k_2} * (\mathbf{L}[k_1] - 1)$;
14 $\quad\quad \mathbf{H}[(k_1, k_2)] = (c_{k_1}^{k_2}, o_{k_1}^{k_2})$;
15 $\quad$ **end**
16 **end**
17 **return** $\mathbf{H}, \mathbf{L}$

We treat frame sequence as $\mathcal{L}_0$ with length $n_0 = N_f$ for convenience. We calculate $c_{k_1}^{k_2}$ and $o_{k_1}^{k_2}$ with Algorithm 1. The core codes are from line 12 and 13 that update $(c_{k_1}^{k_2}, o_{k_1}^{k_2})$ in an iterative manner. Here is a brief proof. We define an additional operation $\mathscr{R}_{k_1}^{k_2}(j)$ similar in Equation (3) to find the index of the rightmost descendant of $e_{j_1}^{k_1}$ in $\mathcal{L}_{k_2}$:

$$\mathscr{R}_{k_1}^{k_2}(j) = \arg\max_{j_2} \{e_{j_2}^{k_2} | e_{j_2}^{k_2} \in \mathcal{D}(e_{j_1}^{k_1})\} \tag{12}$$

It is not difficult to find relations:

$$\begin{aligned} \mathscr{L}_{k_1}^{k_2}(j) &= o_{k_1}^{k_2} \times j \\ \mathscr{R}_{k_1}^{k_2}(j) - \mathscr{L}_{k_1}^{k_2}(j) &= c_{k_1}^{k_2} - 1 \end{aligned} \tag{13}$$

We obtain $\mathscr{L}_{k_1}^{k_2}(1) = o_{k_1}^{k_2}$ when j is set to 1. Furthermore, $\mathscr{L}$ and $\mathscr{R}$ have a transitive property, as shown in Figure 6, which leads to:

$$\begin{aligned} o_{k_1}^{k_2} &= \mathscr{L}_{k_1}^{k_2}(1) = \mathscr{L}_{k_1-1}^{k_2}(\mathscr{L}_{k_1}^{k_1-1}(1)) \\ &= \mathscr{L}_{k_1-1}^{k_2}(o_{k_1}^{k_1-1}) = o_{k_1-1}^{k_2} \times o_{k_1}^{k_1-1} \\ &= ... = \prod_{k=k_1}^{k_2+1} o_k^{k-1} \end{aligned} \tag{14}$$

Similarly, we leverage the transitive property of $\mathscr{R}$ on events located at the rightmost position of each layer:

$$\begin{aligned}\mathscr{R}_{k_1}^{k_2}(n_{k_1}-1) &= \mathscr{R}_{k_1-1}^{k_2}(\mathscr{R}_{k_1}^{k_1-1}(n_{k_1}-1)) \\ &= \mathscr{R}_{k_1-1}^{k_2}(n_{k_1-1}-1) = ... = n_{k_2}-1\end{aligned} \tag{15}$$

Then, by applying equation (13), $c_{k_1}^{k_2}$ can be calculated as :

$$\begin{aligned}c_{k_1}^{k_2} &= \mathscr{R}_{k_1}^{k_2}(n_{k_1}-1) - \mathscr{L}_{k_1}^{k_2}(n_{k_1}-1)+1 \\ &= (n_{k_2}-1) - o_{k_1}^{k_2} \times (n_{k_1}-1) + 1 \\ &= n_{k_2} - o_{k_1}^{k_2} \times (n_{k_1}-1)\end{aligned} \tag{16}$$

Equation (14) and (16) are implemented in an iterative manner in line 12, 13 of Algorithm 1.

Step 2: Mask Construction

Then we construct masks based on these structure parameters produced in Algorithm 2, filling attention areas with positive values layer-by-layer. For the sub-mask $M_{k_1}^{k_2} \in \mathbb{R}^{n_{k_1} \times n_{k_2}}$ from $\mathcal{L}_{k_1}$ to $\mathcal{L}_{k_2}$, $(k_1 > k_2)$, the attention scores are filled conditioning on relation $e_{j_2}^{k_2} \in \mathcal{D}(e_{j_1}^{k_1})$ and positions:

$$M_{k_1}^{k_2}[j_1][j_2] = \begin{cases} 1 & \text{if } 0 \le j_2 - o_{k_1}^{k_2} \times j_1 < c_{k_1}^{k_2} \\ 0 & \text{else} \end{cases} \tag{17}$$

Mask $M_k^k \in \mathbb{R}^{n_k \times n_k}$ on the same layer $\mathcal{L}_k$ is an identity matrix. $\widetilde{M}_f^e$ is constructed on the frame layer $\mathcal{L}_0$. $\widetilde{M}_v^e$ is same as the mask on $\mathcal{L}_1$. $\widetilde{M}_f^e, \widetilde{M}_v^e$ are further expanded to M_f^e, M_v^e with shapes $\mathbb{R}^{N_e \times (N_f \times N_s)}, \mathbb{R}^{N_e \times (n_1 \times N_v)}$.

C More Visualization

We provide additional visualizations in Figure 7 on TVR, Charades-STA, and ActivityNet-Captions. Notably, events in ActivityNet tend to interact with high-level event prompts such as the global prompt. This occurs because many ActivityNet videos revolve around a single theme, contain longer textual annotations, and exhibit more complex dependencies. Consequently, the prompts must interact with higher-level ancestors to capture contextual information across broader temporal spans.

Algorithm 2: Mask Construction

Input: $\mathbf{H}$, $\mathbf{L}$
Output: M_e^e, $\widetilde{M}_f^e$, $\widetilde{M}_v^e$

$N_e = \textbf{sum}(\mathbf{L}[1:])$ // prompt number
$n_1 = \mathbf{L}[1]$ // length of $\mathcal{L}_1$
$N_f = \mathbf{L}[0]$ // frame number
$\mathbf{M} = \textbf{zeros}(N_e + N_f, N_e + N_f)$;
for $k_1 \leftarrow K$ ***to*** 0 **do**
  for $k_2 \leftarrow k_1$ ***to*** 0 **do**
    $u_1 = \textbf{sum}(\mathbf{L}[k_1+1:])$;
    $v_1 = \textbf{sum}(\mathbf{L}[k_1:])$;
    $u_2 = \textbf{sum}(\mathbf{L}[k_2+1:])$;
    $v_2 = \textbf{sum}(\mathbf{L}[k_2:])$;
    if $k_1 = k_2$ **then**
      // same layer
      $\mathbf{M}_{sub} = \mathbf{M}[u_1:v_1, u_2:v_2]$;
      $\mathbf{M}_{sub}.fill\_diagonal(1)$
    end
    else
      $c_{k_1}^{k_2}, o_{k_1}^{k_2} = \mathbf{H}[(k_1, k_2)]$;
      for $i \leftarrow 0$ ***to*** $\mathbf{L}[k_1] - 1$ **do**
        $u_i = u_2 + i * o_{k_1}^{k_2}$;
        $v_i = u_i + c_{k_1}^{k_2}$;
        $\mathbf{M}[u_1 + i][u_i : v_i] = 1$;
        // symmetrical
        $\mathbf{M}[u_i : v_i][u_1 + i] = 1$;
      end
    end
  end
end
$M_e^e = \mathbf{M}[:N_e][:N_e]$ // $\mathbb{R}^{N_e \times N_e}$
$\widetilde{M}_f^e = \mathbf{M}[:N_e][N_e:]$ // $\mathbb{R}^{N_e \times N_f}$
$\widetilde{M}_v^e = \mathbf{M}[:N_e][N_e - n_1 : N_e]$ // $\mathbb{R}^{N_e \times n_1}$
return $M_e^e, \widetilde{M}_f^e, \widetilde{M}_v^e$

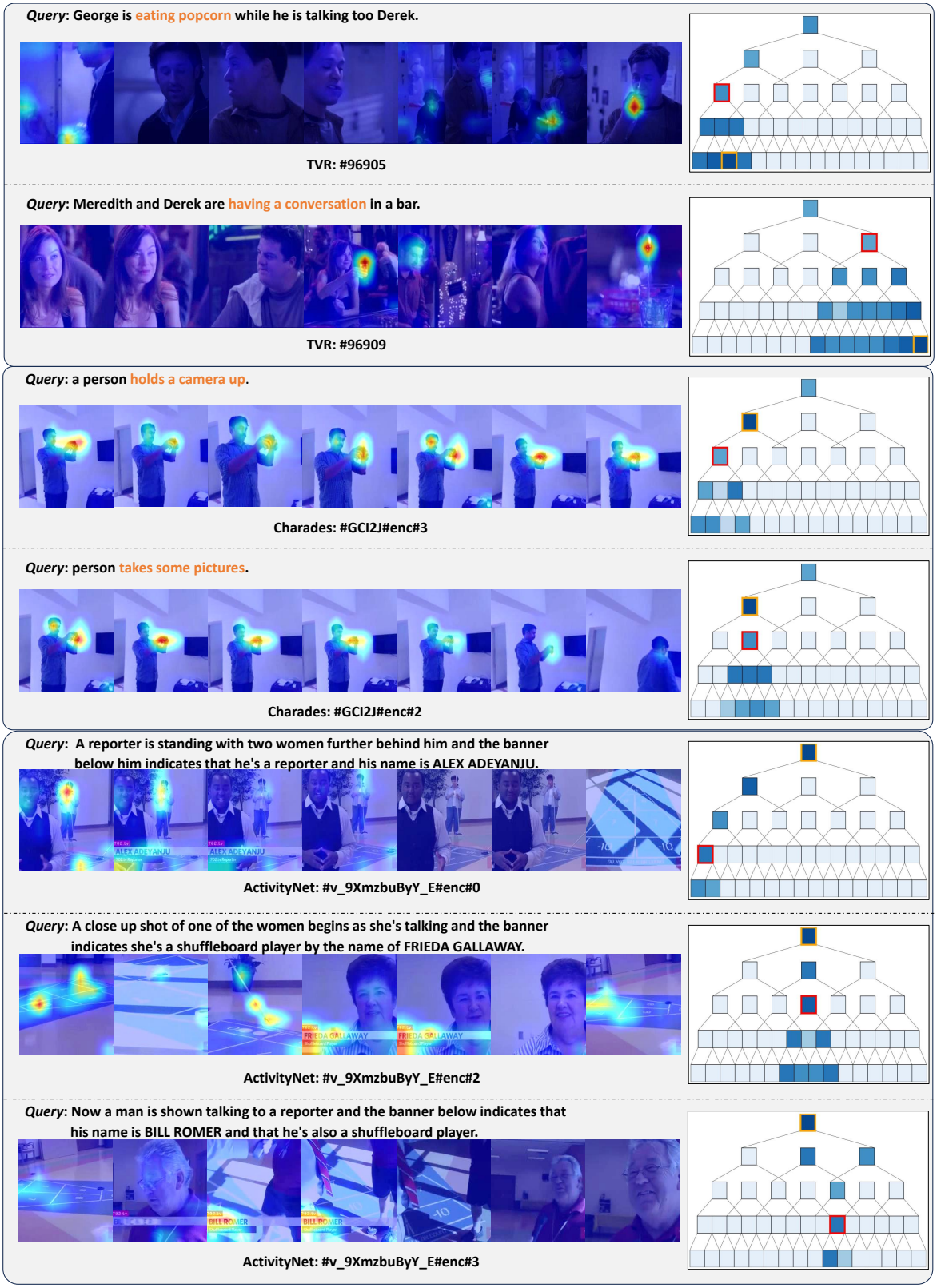

Figure 7: More visualization results. Samples are selected based on the R@1 metric. Selected event prompts are highlighted with red borders. Events with the highest attention scores are marked by orange borders.