Prompt-guided Disentangled Representation for Action Recognition

Tianci Wu, Guangming Zhu,* **Jiang Lu, Siyuan Wang, Ning Wang,**
Nuoye Xiong, Zhang Liang
School of Computer Science and Technology, Xidian University
{tianciwu, lujiang, siyuanwang, ningwang, nyx}@stu.xidian.edu.cn
{gmzhu, liangzhang}@xidian.edu.cn

Abstract

Action recognition is a fundamental task in video understanding. Existing methods typically extract unified features to process all actions in one video, which makes it challenging to model the interactions between different objects in multi-action scenarios. To alleviate this issue, we explore disentangling any specified actions from complex scenes as an effective solution. In this paper, we propose Prompt-guided Disentangled Representation for Action Recognition (ProDA), a novel framework that disentangles any specified actions from a multi-action scene. ProDA leverages Spatio-temporal Scene Graphs (SSGs) and introduces Dynamic Prompt Module (DPM) to guide a Graph Parsing Neural Network (GPNN) in generating action-specific representations. Furthermore, we design a video-adapted GPNN that aggregates information using dynamic weights. Extensive experiments on two complex video action datasets, Charades and SportsHHI, demonstrate the effectiveness of our approach against state-of-the-art methods. Our code can be found in `https://github.com/iamsnaping/ProDA.git`.

1 Introduction

Action recognition is one of the fundamental tasks in the field of video understanding [16], and it remains an active topic in the vision research community. Current mainstream methods can be categorized into two types based on their data processing approach. The first type is based on complete video data. These methods typically input the untrimmed video into 2D Convolution Neural Network (CNN), 3D CNN, or Vision Transformer (ViT) [7], using the network to extract features and then performing classification using the extracted features [10, 42, 4, 45, 27, 30, 27]. The second type is object-centric, where object detectors first crop objects from the video and then input them into ViT or Graph Neural Network (GNN) [53, 33, 57, 48, 54, 55, 48, 53]. Some methods [59, 20, 47, 35] also construct Spatio-temporal Scene Graph (SSG) from these objects and then feed them into the network. However, these methods are essentially information extraction processes that handle all actions in the video at once, making it difficult to model the interactions between objects in complex scenes [48, 55]. In this paper, we aim to analyze actions with a disentangled approach. For instance, in a video with multiple actions, we want the network to focus on analyzing some specified action(s). Moreover, we construct SSGs for this video and aim to disentangle the subgraph, which consist of the specified action-relevant nodes, enabling better understanding of the actions (as shown in Figure 1). Existing Disentangled Representaion Learning (DRL) methods typically approach disentanglement from different perspectives. Some methods focus on attribute disentanglement, aiming to separate content from background [22, 52, 17]. Others adopt an information perspective by disentangling representations into structural invariants and local variations (e.g., shared information
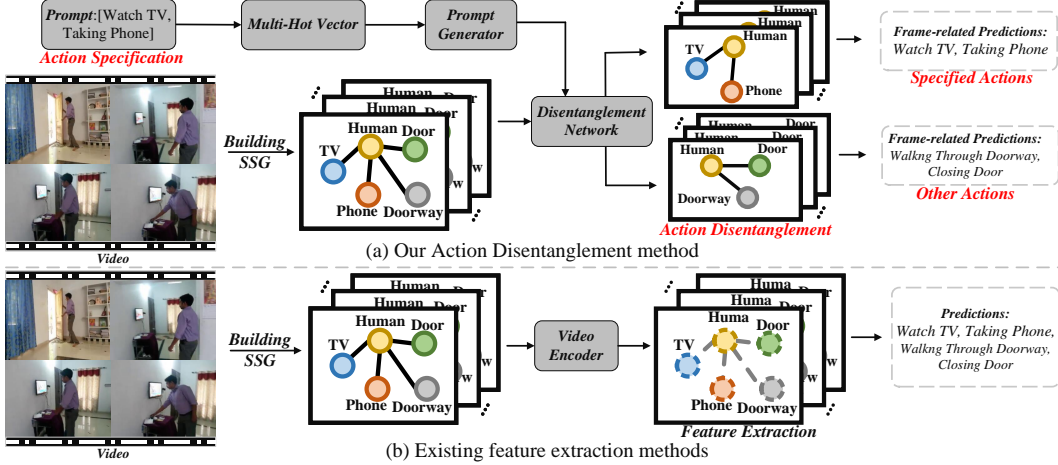
---

*Corresponding Author

Figure 1: Comparison between (a) our proposed Prompt-guided Representation Disentanglement method and (b) existing methods using SSG. Existing methods treat action modeling as a feature extraction process, handling all actions in one video. In contrast, our method disentangles the action representation into two parts: one for the specified action and the other for the remaining actions, enabling more focused analysis and better action understanding.

across nodes and private, node-specific features) [6, 58, 12, 34]. From a structural perspective, some approaches disentangle graphs into a fixed number components(e.g., latent representation, subgraphs) [26, 25, 24, 6, 58, 12]. A task perspective has also been explored, where complex tasks are divided into multiple subtasks to facilitate more structured representation learning [29]. In action recognition, **we argue that disentanglement should be approached from the action perspective**, enabling the dynamic separation of semantic factors related to *any specified actions*. Specifically, we expect the network to analyze any actions we specify and to disentangle the corresponding subgraph from the whole SSG, enabling more targeted and interpretable action understanding.

To this end, in order to enable the model to disentangle representations conditioned on any specified action, we draw inspiration from recent advances in prompt tuning [31, 8, 39, 21, 56, 2] and propose Prompt-guided Disentangled Representation for Action Recognition (ProDA). As depict in Figure 1 (a), **ProDA disentangles the SSG into two complementary parts: one corresponding to the specified actions, and the other to the remaining actions.** To achieve this, we construct an action-aware disentanglement network that can disentangle any specified actions from SSG based on prompt guidance. Specifically, we introduce the Dynamic Prompt Module (DPM), which generates action-aware prompts by combining the multi-hot vector of specified actions with the SSG features. These prompts guide the Graph Parsing Neural Network (GPNN) to perform action-aware disentanglement. Furthermore, we propose a Video Graph Parsing Neural Network (VGPNN), a GPNN variant tailored for SSG, to perform action disentanglement. To ensure temporal consistency and preserve the differences across SSGs from different videos, we introduce a Video Graph Normalization (VGNorm) module, which is applied after each VGPNN layer. Finally, we introduce an Action Disentanglement Loss (AD Loss) to supervise the separated representations. To ensure that the disentangled features retain essential information, we incorporate a reconstruction term, promoting true representation disentanglement rather than mere feature extraction.

In summary, the major contributions of this work are as follows: 1) We present a novel framework ProDA, which disentangles any specified action from multi-action videos, enhancing the interpretability of action understanding; 2) Our method achieves State-of-The-Art performance on multi-label action recognition benchmark. 3) Without requiring action localization supervision, our method demonstrates strong action localization capabilities, further validating its effectiveness and interpretability in understanding complex actions.

2 Related works

Existing action recognition methods can be broadly categorized into three types. First type is CNN-based methods. To compensate for the lack of temporal modeling in 2D CNNs, two-stream networks introduce an additional temporal stream (e.g., optical flow)[10, 9, 41, 44, 49]. By jointly modeling
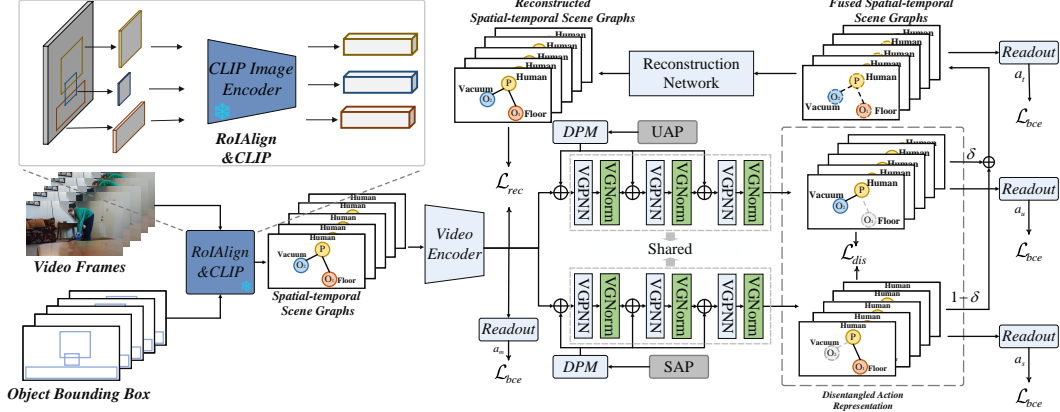
Figure 2: An overview of our ProDA. We first convert the input video into a sequence of scene graphs, which are then processed by a Video Encoder. Next, we generate two complementary prompts from the Action Set (SAP and UAP) using DPM to guide the VGPNN in disentangling the actions into two complementary SSGs, each containing distinct parts of the action. These disentangled SSGs, along with their fused version, are fed into corresponding readout blocks to obtain the global representations of partial and complete actions, respectively.

spatial and temporal features, 3D CNNs are able to better capture spatio-temporal information and thus yield improved performance [42, 4, 51]. The second type is ViT-based methods. Most methods either treat entire frames as input tokens or divide each frame into patches before feeding them into the model [13, 45, 27, 30, 27]. These methods treat the video as a whole, which makes it difficult to model interactions among objects, and often introduces substantial noise. Our work alleviate this by disentangling actions from the SSG, enabling more precise modeling of object interactions. The third type is Objected-centric methods. Object-centric methods construct more compact and efficient video representations by focusing on regions of interest (e.g., objects in the video) [53], and enabling generalization of unseen interaction relationships through structured representations [53, 33, 57, 48, 59]. Nevertheless, these methods remain limited in their ability to effectively model interactions among objects. The incorporation of SSG [59, 20] effectively addresses the difficulty in modeling object interactions. In scene graphs, videos are represented as spatio-temporal graph structures, with objects serving as nodes (e.g., human, cup) and their interactions or spatial-relationship (e.g., holding, above) modeled as edges. [35] processes the scene graph of each frame using a Graph Neural Network (GNN). On the other hand, [47] flattens the scene graphs and feeds them into a ViT [7], leveraging the long-range modeling capability of self-attention [43] to effectively capture interaction transitions. These SSG-based methods still adopt feature extraction approaches that process all actions in a video, which makes it challenging to model interactions between different objects in multi-action scenarios. In contrast, our method disentangles actions from complex scenes, enabling better understanding of the actions

3 Methodology

Our method aims to disentangle any specified actions from multi-action scenarios. This disentanglement provides more concise feature representation for analyzing the specified actions, leading to improved action understanding. The overall architecture of our method is illustrated in Figure 2, we first convert the input video into a SSG. This graph is then processed by a video encoder (specifically, Dual-AI [15]) to model temporal and spatial interactions among nodes within the SSG. Next, guided by the DPM, the video features are fed into VGPNN for representation disentanglement. Finally, to further enhance the disentanglement, in addition to standard classification supervision, we introduce both a reconstruction loss and a disentanglement loss as constraints. We generate the SSG [38] using existing methods and adopt a off-the-shelf video encoder [15]; the following focuses on the remaining components of the network architecture illustrated in Figure 2.

3.1 Formulation

In SSG, each frame of a video is processed into a graph, where nodes represent the objects present in that frame, and edges represent the semantic relationships between these objects (e.g., spatial or

interaction relations). Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ denote the complete SSG constructed from the video, where $\mathcal{V}$ is the set of all object nodes across all frames, $\mathcal{E}$ is the set of all directed edges, and $\mathcal{R}$ is the set of all relation types. Each node $v \in \mathcal{V}$ is uniquely indexed from $\{1, \ldots, |\mathcal{V}|\}$. Each edge $e = (v, w, r) \in \mathcal{V} \times \mathcal{V}$ represents a directed semantic relation of type $r \in \mathcal{R}$ from node v to node w. Let $g_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{R}_i)$ denote the scene graph of the i-th frame, where $\mathcal{V}_i \subseteq \mathcal{V}$, $\mathcal{E}_i \subseteq \mathcal{E}$ and $\mathcal{R}_i \subseteq \mathcal{R}$ denote the frame-specific object nodes, their edges and relations. We denote v_i^j as the j-th object node in frame i, and $e_i^{j,k}$ as the edge from v_i^j to v_i^k with relation type $r_i^{j,k} \in \mathcal{R}_i$. The node features can be denoted as $\mathcal{F}_{\text{node}} = \{f_1^1, f_1^2, \ldots, f_N^M\} \in \mathbb{R}^{N \times M \times D}$, where $f_i^j \in \mathbb{R}^D$ represents the feature of node v_i^j in the j-th frame. Here, N, M, and D denote the number of frames in the video, the number of nodes per frame, and the dimensionality of each node feature, respectively.

3.2 Dynamic Prompt Module (DPM)

The DPM is a key component of our method. Since action disentanglement is driven by prompts, ambiguous prompts can make it hard for the model to focus on the specified action-relevant regions, leading to failure in disentangling the specified actions. we generate dynamic prompts based on the node features of the SSG, which guide the model to attend to action-relevant nodes and achieve better disentanglement. In this section, we introduce how prompts are generated based on arbitrary action specifications. We first describe the design of action specifications, followed by their construction strategy. Finally, we show how prompts are generated from the specifications, which guide the VGPNN to disentangle actions. Inspired by [8, 31], we inject learnable prompts into SSG feature space to accomplish guidance.

Action Specification Design. We first represent each action category as a one-hot vector. To specify a combination of actions to disentangle, we construct a multi-hot vector (e.g., $[1, 0, \ldots, 1]$), where each 1 indicates a hot element, i.e., a specified action. By aggregating several one-hot vectors, forming the Action Specification (AS). In designing the AS, we consider two components: one for the actions we intend to disentangle, called the Specified Action Prompt (SAP), and another for the actions we aim to exclude, called the Unspecified Action Prompt (UAP). For the design of SAP, a straightforward approach is to directly input the label-based SAP into the model. However, this can lead to shortcut learning, where the model relies on label cues rather than genuinely parsing the video content. Additionally, in many real-world scenarios, the exact set of actions in a video is not always known. To address these limitations, we further design an alternative version of SAP that includes both the ground-truth actions and distractor actions (i.e., actions not present in the video). This encourages the model to focus on learning discriminative features rather than simply following label hints. For the distractor-injected SAP design, we keep the number of hot elements in the SAP multi-hot vector to K, only some of which correspond to ground-truth actions, while the rest are distractors. To prevent the model from exploiting shortcuts, we construct the UAP by inverting the SAP. In this way, the combination of SAP and UAP covers the entire label set, ensuring a balanced and comprehensive specification of both present and absent actions (For better clarity, we include illustrative examples in Sec. C.1). In certain cases, SAP and UAP may contain only distractors (i.e., no true actions). For such scenarios, we introduce an auxiliary label called "no-action" to explicitly supervise the model.

Action Specification Construction. The AS consists of two types: distractor-injected and non-distractor-injected. The distractor-injected AS includes a combination of present labels (i.e., labels of some actions present in the video) and absent labels (i.e., labels of some actions absent in the video). Enumerating all possible label combinations is computationally infeasible. To address this, we adopt a random sampling strategy. For a video with L ground-truth labels, the distractor-injected SAPs, we generate $L + 1$ different multi-hot vectors with a length K. Among them, i-th SAP contains i random sampled present labels, and $K - i$ random sampled absent labels. In contrast, the non-distractor-injected SAPs, we generate L different multi-hot vectors, only contains random sampled present labels. UAPs are then obtained by inverting each SAP. This results in $2L + 1$ different pairs of SAP and UAP. This design enables the model to learn from a diverse set of action combinations, promoting robust disentanglement by exposing it to both clean and noisy prompts while maintaining efficiency. For better clarity, we include illustrative examples in Sec. C.2.

Prompt Generation. A possible way to incorporate AS (symbolized as $\mathbf{v}$) is to project it into the same dimension space as the node features using a linear layer and add it directly to the features, we refer to this method as SimplePrompt. However, due to the static essence of such prompts, this

approach may struggle to capture contextual variations and risks encouraging shortcut learning, particularly in our dynamic disentanglement setting, where guidance need adapt to varying action contexts. To address this, we propose the Dynamic Prompt Module, which generates an adaptive prompt p_i^j to each node feature f_i^j. The prompt p_i^j is constructed as a linear combination of T candidate prompts $\{q_1, q_2, \ldots, q_T\}$, which are generated from $\mathbf{v}$. The combination weights $\Delta \mathcal{P}^{i,j} = \{\alpha_1^{i,j}, \alpha_2^{i,j}, \ldots, \alpha_T^{i,j}\}$ are computed based on both $\mathbf{v}$ and f_i^j. It can be expressed as:

$$\Delta \mathcal{P}^{i,j} = \sigma(W^w \cdot (W^y \cdot \mathbf{v} + f_i^j)) \tag{1}$$

The resulting prompts is then added to $\mathcal{F}_{node}$ to generate prompted feature $\mathcal{F}_{node}^\star$, which replaces $\mathcal{F}_{node}$ in subsequent processing. It can be expressed as:

$$\mathcal{F}_{node}^\star = \{f_1^1 + p_1^1, f_1^2 + p_1^2, \ldots, f_N^M + p_N^M\} \qquad p_i^j = \sum_{t=1}^{T} \alpha_t^{i,j} \cdot q_t \tag{2}$$

3.3 Video Graph Parsing Neural Network (VGPNN)

We aim to disentangle subgraphs corresponding to specified actions. To achieve this, we require a network that, under the guidance of prompts, can parse action-relevant subgraphs by directionally propagating action-specific information. Inspired by GPNN [37], we design a variant for SSG, called VGPNN. GPNN parses action-relevant nodes by generating dynamic weights that control the direction of information flow. It consists of three key steps: Link, Message, and Update. In the Link step, it constructs edges between the same object nodes across different frames and learns aggregation weights for each timestamp. In the Message step, it multiplies these weights with messages, where each message is composed of a node's features and its edge features. In the Update step, it selects nodes from a specific frame as the initial hidden states, and updates them sequentially using the weighted messages from different time steps. However, GPNN has several limitations when applied to SSGs. 1)It only updates a node from a single frame, ignoring joint modeling of nodes across the entire video. 2) It performs shallow temporal modeling, as the learned weights are used only once for aggregation and are not further refined or propagated through deeper temporal structures.

Video Graph Parsing Neural Network (VGPNN). VGPNN is designed to disentangle features by decomposing a SSG into two subgraphs, guided by a given AS. To achieve selective information aggregation, with the same to GPNN [37], each VGPNN layer processes the entire SSG in three sequential steps. In Link step, we perform temporal modeling of edges $e^{j,k} = \{e_1^{j,k}, e_i^{j,k}, \ldots, e_N^{j,k}\}$ via self-attention [43] over the same interaction across time. In Message step, we generate dynamic weights from the edge features, which are then multiplied with the messages, where each message is composed of a node's features and its edge features. In Update step, we update all nodes by adding the weighted messages to the corresponding nodes directly. A MLP is then used to fuse the updated and original node features. More details please refer to Sec. B.

Video Graph Normalization (VGNorm). SSG exhibits characteristics of both graph structure and spatio-temporal properties of videos. However, many commonly used normalization methods tend to focus on only one aspect. For instance, BatchNorm [18] primarily normalizes across the temporal batch dimension, while LayerNorm [1] operates along the node (spatial) dimension to enforce spatial consistency. This one-sided focus limits network performance. To ensure temporal consistency and preserve the differences across SSGs from different videos, our method focuses on two key aspects. First, we use the frame-wise global mean $\mu_g \in \mathbb{R}^N$ to ensure temporal consistency across frames. During training, we compute the mean $\mu \in \mathbb{R}^N$ for each batch in real-time and update the $\mu_g \in \mathbb{R}^N$ using a momentum-based approach. Specifically, for batchsize B nodes features $\mathcal{F}^{node} = \{\mathcal{F}_1^{node}, \mathcal{F}_2^{node}, \ldots, \mathcal{F}_B^{node}\} \in \mathbb{R}^{B \times N \times M \times D}$, where $\mathcal{F}_b^{node} \in \mathbb{R}^{N \times M \times D}$ is the b-th video nodes features, the global mean is computed as:

$$\mu_g := \alpha_m \cdot \mu_g + (1 - \alpha_m) \cdot \mu \qquad \mu = \frac{1}{B \cdot M \cdot D} \sum_{b=0}^{B} \sum_{j=0}^{M} \sum_{d=0}^{D} \mathcal{F}^{node} \tag{3}$$

where α_m represent the momentum. This global mean is used to normalize the features at each frame, ensuring temporal consistency.

Second, we introduce a mean scale shift $\alpha \in \mathbb{R}^N$ to account for the heterogeneity across different video SSGs. For the node features $\mathcal{F}_b^{node}$ of the b-th video, the normalization is defined as:

$$\mathcal{F}_b^* = \gamma \cdot \frac{\mathcal{F}_b^{node} - \alpha \cdot \mu_g}{\hat{\sigma}} + \beta \qquad \hat{\sigma}^2 = \frac{\sum_{j=0}^{M} \sum_{d=0}^{D} (\mathcal{F}_b^{node} - \alpha \cdot \mu_g)^2}{M \cdot D - 1} \tag{4}$$

Here, $\gamma, \beta \in \mathbb{R}^N$ are learnable affine parameters, and $\mathcal{F}_b^*$ is the normalized feature that replaces $\mathcal{F}_b$ in subsequent processing.

Readout. First, we compute dynamic weights for intra-frame nodes to aggregate them into frame-level features. Then, we calculate frame-level weights s for each frame feature to perform temporal aggregation, yielding a global representation of the SSG, which is used for subsequent action classification supervision.

3.4 Optimization

In this section, we present the loss functions used to optimize our framework. Our design includes two complementary objectives: an **Action Classification Loss**, which guides the model to correctly disentangle features as specified by the AS, and an **Action Disentanglement Loss**, which ensures that the resulting representations are well separated. Together, these losses encourage both accurate action recognition and effective disentanglement in multi-action video scenes.

Action Disentanglement Loss (AD Loss). AD Loss includes a **reconstruction** loss and a **disentanglement** loss. For disentanglement, we treat the two disentangled representations as random variables and minimize their Pearson's correlation coefficient to enforce disentanglement. However, complete separation may discard shared semantics (e.g., common objects or interactions). To preserve these, we introduce a margin-based relaxation using a ReLU, allowing reasonable overlap while penalizing only excessive correlation. For a mini-batch with batchsize B, disentanglement loss $\mathcal{L}_{dis}$ can be expressed as:

$$\mathcal{L}_{dis} = \frac{1}{B} \cdot \sum_{b=0}^{B} \text{ReLU}\left(\frac{|\text{Cov}(\phi_b(\mathcal{F}_b^u), \psi_b(\mathcal{F}_b^s))|}{\sqrt{\text{Var}(\phi_b(\mathcal{F}_b^u))} \cdot \sqrt{\text{Var}(\psi_b(\mathcal{F}_b^s))}} - m_1 \right) \tag{5}$$

where $\mathcal{F}_b^u, \mathcal{F}_b^s \in \mathbb{R}^{N \times M \times D}$ represent the action representations disentangled by UAP and SAP of b-th video, respectively. ϕ_b, ψ_b are measurable functions [14]. $\text{Cor}(\cdot), \text{Var}(\cdot), m_1$ represent covariance, variance and margin, respectively.

To ensure that the model performs disentanglement rather than feature extraction. We reconstruct the original feature $\mathcal{F}_b^o \in \mathbb{R}^{N \times M \times D}$, which is obtained from the video encoder before disentanglement. And we fuse the features $\mathcal{F}_b^u, \mathcal{F}_b^s$ and then obtain the reconstructed representations $\mathcal{F}_b^r \in \mathbb{R}^{N \times M \times D}$ with the reconstruction network. The reconstruction process can be expressed as

$$\mathcal{F}_b^r = \text{Net}_1(\delta \cdot \mathcal{F}_b^u + (1 - \delta) \cdot \mathcal{F}_b^s) \qquad \delta = \text{Net}_2([\mathcal{F}_b^u, \mathcal{F}_b^w]) \tag{6}$$

where $\text{Net}_1, \text{Net}_2$ represent reconstruction and fusion weight δ generation functions. And we apply a Mean Squared Error to enforce this constraint. Since explicit guidance is added. If we were to strictly align $\mathcal{F}_b^o$ with $\mathcal{F}_b^r$, the video encoder might focus on aligning with the prompts instead of the video. This could result in the video encoder extracting irrelevant information, causing the model to output results based on the prompts instead of the video. To prevent this, we reserve space by applying a margin-based ReLU relaxation m_2 to tolerate overlaps, reconstruction loss $\mathcal{L}_{rec}$ can be expressed as:

$$\mathcal{L}_{rec} = \frac{1}{B} \cdot \sum_{b=0}^{B} \text{ReLU}(\| \mathcal{F}_b^o - \mathcal{F}_b^r \|_2^2 - m_2) \tag{7}$$

Overall. In ProDA, we predict the labels of the disentangled actions specified by UAP and SAP, denoted as a_u, a_s as well as the fused action label a_t (i.e., the video label). The first two predictions ensure that the model correctly disentangles representations according to the AS, while the fused prediction encourages information completeness and promotes interaction between the disentangled components. During training, we have access to the ground-truth labels for UAP and SAP, denoted as y_u, y_w, along with the video ground-truth label y_t. Therefore, we can compute the total loss $\mathcal{L}$ as:

$$\mathcal{L} = \underbrace{\lambda_1 \cdot \mathcal{L}_{bce}(a_u, y_u) + \lambda_2 \cdot \mathcal{L}_{bce}(a_w, y_w) + \lambda_3 \cdot \mathcal{L}_{bce}(a_t, y_t)}_{\text{actions classification loss}} + \underbrace{\lambda_4 \cdot (\mathcal{L}_{dis} + \mathcal{L}_{rec})}_{\text{action disentanglement loss}} \tag{8}$$

where, $\mathcal{L}_{bce}$ is binary cross-entropy loss.

4 Experiments

4.1 Experiment Setup

Dataset. (1) The Charades dataset [40] consists of 9,848 videos, each with an average duration of 30 seconds. It covers 157 action categories, with each video containing an average of 6.8

Table 1: Multi-label action recognition performance comparison on the Charades's validation set in term of mAP. SSG:ground truth SSG. Bbox: bounding box from ground truth SSG.

model	pretrain	Modality	mAP	model	pretrain	Modality	mAP
SGFB+NL [20]	K400	RGB	44.3	SGFB+NL [20]	K400	RGB+SSG	60.3
I3D+NL [4]	K400	RGB	37.5	LaIAR [47]	K400	RGB+Bbox	63.6
OR2G+NL [35]	K400	RGB	44.9	OR2G [35]	K400	RGB+SSG	63.3
LaIAR+NL [47]	K400	RGB	45.1	LaIAR+NL [47]	K400	RGB+Bbox	67.4
VicTR(B/16) [23]	CLIP	RGB	50.1	OR2G+NL [35]	K400	RGB+SSG	67.5
VicTR(L/14) [23]	CLIP	RGB	57.6	**Ours(B/16)**	**CLIP**	**RGB+SSG**	**71.1**

Table 2: Human-Human Interaction Classification results in SportsHHI. SportsHHI* is our reimplementation based on the official SportsHHI code, with minor modifications (e.g., parameter size) to match the capacity of ProDA for fair comparison.

method	mAP	R@150	R@100	R@50	R@20	Params
STTran [5]	3.31	71.29	62.91	42.67	22.14	-
HORT [19]	3.75	78.57	67.78	50.33	26.96	-
SlowFast [11]	5.00	81.66	74.00	52.74	26.82	-
ACARN [36]	5.44	82.85	75.22	56.53	31.77	-
SportHHI [50] (baseline)	10.69	89.25	82.93	68.13	43.72	111.18M
SportsHHI* (reimpletmented)	10.16 ($\downarrow$0.53)	90.07 ($\uparrow$0.82)	84.83 ($\uparrow$1.9)	68.40 ($\uparrow$0.37)	40.16 ($\downarrow$ 3.56)	123.11M
Ours (baseline+ProDA)	**13.55 ($\uparrow$2.86)**	**92.54 ($\uparrow$3.29)**	**87.91 ($\uparrow$4.98)**	**72.69 ($\uparrow$4.56)**	**47.81 ($\uparrow$4.09)**	123.19M

distinct actions. Multiple actions can occur simultaneously, making the recognition task particularly challenging. Based on Charades, the Action Genome dataset [20] provides fine-grained annotations by decomposing actions and focusing on video clips where the actions take place. It contains 234K keyframes, with annotations for 476K object bounding boxes and 1.72 million object relationships. (2) The SportsHHI dataset [50] is a specially designed dataset focusing on human-human interaction in sports. It covers basketball and volleyball games, and video data are selected from MultiSports [28]. This dataset labels 34 interaction classes with 118,075 human bounding boxes and 50,649 interaction instances. Following [47, 50], we conduct evaluations on Charades for multi-label action recognition and on SportsHHI for Human-Human interaction Classification (HHICls), adopting mean Average Precision (mAP) as the metric.

Experiment Setup. (1) On Charades, all experiments are conducted under the same design. For each video, the network takes N frames as input. During training, frames are randomly sampled from the video, while during validation, the frames are extracted evenly over the whole video. In the first stage, the entire network is trained end-to-end, and the resulting weights are used to initialize the second stage. In the second stage, only the classifiers for a_u, a_s, a_t are trained. Additionally, to enable deeper analysis, we introduce an auxiliary prediction a_m, which is inferred from the features before disentanglement (features from Video Encoder as illustrated in Figure 2). Its corresponding classifier is also trained during the second stage. (2) On SportsHHI, we adopt the official open-source model and training code without altering any training settings. For fair evaluation, we conduct two experiments: 1) extending the official baseline with our proposed modules, namely the DPM and AD Loss; and 2) constructing a parameter-matched baseline by enlarging its model size to match ours, without incorporating any proposed components.

4.2 Compared with State-of-The Art Methods

Charades. We compare the action recognition accuracy of the proposed method and the State-of-The-Art methods (SoTA) on the Charades. Table 1 presents our results on the Charades dataset. ProDA achieves state-of-the-art performance. Our method outperforms several CNN- and ViT-based approaches, including I3D [4] and VicTR [23]. This demonstrates that our approach can better capture object interactions within structured data (e.g., SSG), enabling a deeper understanding of video. Notably, our method with a CLIP-B/16 [38] still surpasses VicTR (L/14) [23], even though we only use an image encoder, while VicTR [23] additionally employs a text encoder. Furthermore, our method also outperforms several approaches that leverage SSG [35, 47]. This demonstrates that explicitly disentangling and processing actions separately leads to a more accurate understanding of complex actions. It is also worth noting that our method achieves this performance without requiring any additional external models.

Table 3: Ablation of prompt

prompt type	a_tmAP (%)	a_mmAP (%)
SimplePrompt	68.07	57.80
DPM	**71.10**	**59.04**

Table 4: Ablation of normalization

norm type	a_tmAP (%)	a_mmAP (%)
LayerNorm [1]	68.12	57.96
BatchNorm [18]	65.39	58.20
GraphNorm [3]	68.61	58.32
VGNorm	**71.10**	**59.04**

Table 5: Ablation study of loss function on Charades dataset. ❻ denotes the baseline training only the video encoder without ProDA; ❶ denotes using ProDA with only the action classification loss.

	$\mathcal{L}_{res}$ w/o margin	$\mathcal{L}_{res}$	$\mathcal{L}_{dis}$ w/o margin	$\mathcal{L}_{dis}$	a_t mAP on Charades (%)	a_m mAP on Charades (%)
❶	-	-	-	-	67.45	57.38
❷	✓	-	✓	-	68.43	8.03
❸	-	-	-	✓	67.78	58.28
❹	-	✓	-	-	64.89	57.90
❺	-	✓	-	✓	**71.10**	**59.04**
❻	-	-	-	-	-	58.01

SportsHHI. We compare our method with the SoTA baseline and its parameter-increased variant, as reported in Table 2. Enlarging the baseline leads to a 10.8% increase in parameters, whereas our proposed DPM introduces only 1.46M additional parameters (1.18% of the total). Despite such a small overhead, our method achieves notable improvements across all metrics. Even under comparable parameter settings, it consistently outperforms the baseline. These results demonstrate the strong generalization capability of our approach and confirm that the performance gain primarily arises from the proposed AD Loss rather than from increased model capacity.

4.3 Ablation Study

Impact of DPM. DPM is a key component of our framework. To verify its effectiveness, we conduct comparisons with both Simple Prompt and DPM. In addition to evaluating the final fused output a_t, we also train a separate classifier on a_m, the features before disentanglement, for a more comprehensive analysis. The results in Table 3 show that using DPM improves a_t by 3.03% mAP and a_m by 1.24% mAP compared to the SimplePrompt. These findings indicate that our dynamically generated prompts, conditioned on the input, better capture the evolving nature of actions in videos, enabling more effective representation disentanglement. Furthermore, the clearer disentangled features also enhance the performance of the video encoder network.

Impact of VGNorm. We design VGNorm to adapt to SSG, which integrate both video and graph structures. We compare it with several widely used normalization methods, and observe that VGNorm achieves the best performance as shown in Table 4. Compared to BatchNorm [18], VGNorm better preserves the spatial structure of graphs while maintaining temporal consistency, leading to a significant improvement of +5.71% mAP. Compared to LayerNorm [1], which normalizes features independently and ignores the graph structure, VGNorm incorporates spatially consistent mean scaling to preserve temporal consistency and capture the heterogeneity across SSGs from different videos, resulting in a +2.98% mAP improvement. Finally, compared to GraphNorm [3], VGNorm enhances temporal consistency, enabling the model to better capture temporal dynamics, which contributes to a +2.49% mAP improvement. Moreover, by jointly modeling both temporal consistency and the heterogeneity across SSGs, VGNorm produces more stable and adaptive representations. This leads to the best performance on a_m, further demonstrating the effectiveness of our normalization strategy.

Impact of AD Loss. We design ADLoss to enable effective feature disentanglement by introducing a margin-based reconstruction loss and a disentanglement loss. As shown in Table 5, (❶ *v.s.* ❷) removing the margin constraint results in higher a_t performance, but the accuracy of a_m drops sharply to 8.03% mAP, indicating that the video encoder fails to retain meaningful representations. This suggests that, without the margin, the model learns to take a shortcut by relying solely on the prompt to generate features, bypassing true video understanding. In contrast, when only disentanglement is applied (❶ *v.s.* ❸), the model still achieves notable improvements, demonstrating that action disentanglement alone can help capture object states and interaction transitions more effectively. Comparison between (❶ *v.s.* ❹) further shows that treating all actions as a whole leads to suboptimal results, as the model cannot preserve the independence of different actions. However, the disentanglement structure under reconstruction constraints pushes the video encoder to extract richer features, thereby improving its performance. Moreover, (❸ *v.s.* ❺) shows that adding reconstruction transforms the process from mere feature extraction to genuine disentanglement, preserving more interaction cues. Finally, (❹ *v.s.* ❺) confirms that combining reconstruction with disentanglement ensures the independence of action representations, allowing the model to more accurately capture the dynamic interactions within video, ultimately achieving superior performance.

Figure 3: Frame-level weights s predicted by the readout function for different non-distractor-injected SAPs within the same video. The heatmaps (*colored bars*) visualize the predicted weights, reflecting the relevance of each frame to the target action. The *red bars* indicate the ground-truth action segments, while the *green bars* denote our localization results obtained by thresholding s at 0.7.

Table 6: Non-distractor-injected SAP. label nums indicates the length of the ground-truth label set for each video

label nums	a_s mAP	label nums	a_s mAP
1	100.00 ±0.0	9	99.80 ±0.3
2	100.00 ±0.0	10	99.45 ±0.6
3	100.00 ±0.0	11	99.62 ±0.4
4	99.66 ±0.4	12	99.10 ±1.0
5	99.80 ±0.2	13	99.56 ±0.6
6	99.82 ±0.3	14	98.98 ±1.1
7	99.98 ±0.0	15	99.19 ±1.2
8	99.88 ±0.1	16	99.80 ±0.4

Table 7: Distractor-injected SAP. label nums indicates the length of the ground-truth label set for each video

label nums	a_s mAP	label nums	a_s mAP
1	100.0 ±0.0	9	85.63 ±6.6
2	90.52 ±3.5	10	85.87 ±6.7
3	92.61 ±1.1	11	86.56 ±5.0
4	93.16 ±0.7	12	84.85 ±6.5
5	87.17 ±4.8	13	87.42 ±6.4
6	87.69 ±2.9	14	87.73 ±6.0
7	86.65 ±6.1	15	88.58 ±6.0
8	86.32 ±6.1	16	92.10 ±4.9

4.4 Further Analysis

Disentanglement Robustness Analysis. To evaluate the robustness of our action disentanglement, we analyze the a_s accuracy of videos with different numbers of ground-truth labels (denoted as label nums in Table 6 7) when exposed to varying numbers of present labels in the SAP, under both distractor-injected and non-distractor settings. As shown in Table 7, in the distractor-injected case (SAP contains the labels not present in video), we observe a non-monotonic trend: accuracy drops when the number of present labels increases from 5 to 12, but rises again as present labels dominate the input. We attribute the initial drop to the increasing complexity of the prediction task due to more competing present labels under noisy input. The subsequent performance increase is attributed to reduced noise as more present labels are included. Despite the precision varying between 85% and 100% under distractor-injected, our model demonstrates robust performance across different input conditions. Even in the worst-case scenario, where the precision is at its lowest (85%), the model still maintains a relatively high accuracy. This indicates its ability to handle incomplete or noisy data effectively. Moreover, in the non-distractor-injected case (SAPs only contain labels present in video), as shown in Table 6, the model achieves nearly perfect accuracy, reaching close to 100%. This shows that in ideal conditions, the model is able to fully exploit the available information. The robustness in noisy conditions and strong performance on clean data indicate that our model generalizes well and benefits from mutual reinforcement between the two settings.

Temporal Action Localization. Although our method does not explicitly design a localization module, the disentangled representation naturally provides cues for action localization. Specifically, within our readout function, we aggregate temporal features using learned frame-level weights s. These weights reflect the relevance of each frame to the target action, and can thus be interpreted as frame-level localization scores. By normalizing these weights, we obtain a per-frame prediction that serves as our localization output. To obtain frame-level ground truth for localization, we project the action segment annotations onto frames: all frames within the annotated action duration are labeled as positive (1), and the rest as negative (0). We binarize the normalized weights s using a threshold θ, assigning 1 to frames with $s > \theta$ and 0 otherwise, and evaluate the resulting predictions using frame-level mAP based on a_s. As shown in Table 8 9, we observe that the variation in average precision across different θ values is larger than the variation across different IoU thresholds, and the gap between distractor-injected and non-distractor-injected settings narrows as θ increases. This
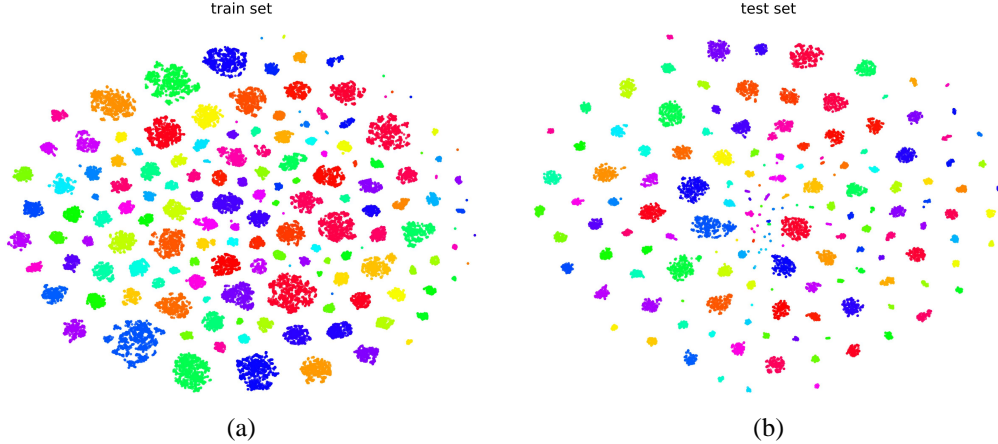
Figure 4: t-SNE visualization of disentangled action features across 157 classes on Charades [40]. (a) Training set. (b) Test set.

<table>
<tr><td colspan="6" align="center">Table 8: Distractor-injected SAP</td></tr>
<tr><td></td><td></td><td colspan="3">IoU threshold</td><td></td></tr>
<tr><td></td><td></td><td>0.2</td><td>0.5</td><td>0.7</td><td>Avg.</td></tr>
<tr><td></td><td>0.2</td><td>77.97</td><td>43.68</td><td>26.08</td><td>49.24</td></tr>
<tr><td>θ</td><td>0.5</td><td>64.25</td><td>33.70</td><td>20.78</td><td>39.57</td></tr>
<tr><td></td><td>0.7</td><td>53.80</td><td>22.98</td><td>12.99</td><td>29.92</td></tr>
<tr><td></td><td>Avg.</td><td>65.34</td><td>33.45</td><td>19.95</td><td>-</td></tr>
</table>

<table>
<tr><td colspan="6" align="center">Table 9: Non-distractor-injected SAP</td></tr>
<tr><td></td><td></td><td colspan="3">IoU threshold</td><td></td></tr>
<tr><td></td><td></td><td>0.2</td><td>0.5</td><td>0.7</td><td>Avg.</td></tr>
<tr><td></td><td>0.2</td><td>85.56</td><td>57.44</td><td>40.18</td><td>61.06</td></tr>
<tr><td>θ</td><td>0.5</td><td>74.44</td><td>43.43</td><td>27.75</td><td>48.53</td></tr>
<tr><td></td><td>0.7</td><td>59.27</td><td>24.26</td><td>13.14</td><td>32.23</td></tr>
<tr><td></td><td>Avg.</td><td>73.09</td><td>41.71</td><td>27.03</td><td>-</td></tr>
</table>

indicates that the introduced noise (i.e., absent labels) mainly adds irrelevant information without significantly disrupting the separation of true actions, demonstrating the robustness of our method. In the non-distractor-injected setting, the model achieves notably better localization performance, confirming that the high recognition accuracy is not due to shortcut learning, but rather due to effective utilization of SAP information for representation disentanglement under clean inputs.

4.5   Visualization

We computed the inter-class and intra-class distances after applying t-SNE [32] for dimensionality reduction on Charades [40], with the visualization results shown in Figure 4. Specifically, for each video with L labels, we disentangle L action features, covering a total of 157 action classes. We further calculate the inter-class and intra-class distances on both the training and test sets. For the training set, the intra-class and inter-class distances are 2.08 ± 1.57 and 58.58 ± 27.99, respectively, while for the test set they are 1.79 ± 1.45 and 74.94 ± 36.98. Notably, the intra-class distances of disentangled features are mostly within 1, whereas the inter-class distances remain substantially larger. This demonstrates that our feature disentanglement effectively increases the separability between action categories while enhancing the compactness of features within the same category, thereby providing more discriminative representations for downstream action recognition and understanding. We provide more visualization results in Sec E.

5   Conclusion

In this paper, we propose Prompt-guided Representation Disentanglement for Action Recognition (ProDA), which is capable of disentangling any complete action from Spatio-temporal Scene Graphs (SSGs). ProDA leverages SSGs and introduces Dynamic Prompt Module (DPM) to guide a Graph Parsing Neural Network (GPNN) in extracting action-specific representations. We further design a video-adapted GPNN that aggregates information using dynamic weights. To enhance performance, we introduce VGNorm and AD Loss, which ensure the spatio-temporal consistency across SSGs of different videos and the completeness of feature disentanglement, respectively. Experimental results demonstrate the superiority of our method.

Acknowledgments and Disclosure of Funding

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.

[3] Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-yan Liu, and Liwei Wang. Graphnorm: A principled approach to accelerating graph neural network training. In *International Conference on Machine Learning*, pages 1204–1215. PMLR, 2021.

[4] Joao Carreira and Andrew Zisserman. 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, 2017.

[5] Y. Cong, W. Liao, H. Ackermann, et al. Spatial-temporal transformer for dynamic scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16372–16382, 2021.

[6] Emily L Denton et al. Unsupervised learning of disentangled representations from video. *Advances in neural information processing systems*, 30, 2017.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 36:52464–52489, 2023.

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[10] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4768–4777, 2017.

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, et al. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[12] Lianqiang Gan, Junyu Lai, Jingze Ju, Lianli Gao, and Yi Bin. Dfdnet: Disentangling and filtering dynamics for enhanced video prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3059–3067, 2025.

[13] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253, 2019.

[14] Arthur Gretton, Alexander Smola, Olivier Bousquet, Ralf Herbrich, Andrei Belitski, Mark Augath, Yusuke Murayama, Jon Pauls, Bernhard Schölkopf, and Nikos Logothetis. Kernel constrained covariance for dependence measurement. In *International Workshop on Artificial Intelligence and Statistics*, pages 112–119. PMLR, 2005.

[15] Mingfei Han, David Junhao Zhang, Yali Wang, Rui Yan, Lina Yao, Xiaojun Chang, and Yu Qiao. Dual-ai: Dual-path actor interaction learning for group activity recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2990–2999, 2022.

[16] Tal Hassner. A critical review of action recognition benchmarks. In *proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 245–250, 2013.

[17] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. *Advances in neural information processing systems*, 31, 2018.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[19] Ji Ji, Rishabh Desai, and Juan Carlos Niebles. Detecting human-object relationships in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8106–8116, 2021.

[20] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10236–10247, 2020.

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

[22] Wentao Jiang, Si Liu, Chen Gao, Jie Cao, Ran He, Jiashi Feng, and Shuicheng Yan. Psgan: Pose and expression robust spatial-aware gan for customizable makeup transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5194–5202, 2020.

[23] Kumara Kahatapitiya, Anurag Arnab, Arsha Nagrani, and Michael S Ryoo. Victr: Video-conditioned text representations for activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18547–18558, 2024.

[24] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. Disentangled contrastive learning on graphs. *Advances in Neural Information Processing Systems*, 34:21872–21884, 2021.

[25] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Disentangled graph contrastive learning with independence promotion. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):7856–7869, 2022.

[26] Haoyang Li, Xin Wang, Zeyang Zhang, Haibo Chen, Ziwei Zhang, and Wenwu Zhu. Disentangled graph self-supervised learning for out-of-distribution generalization. In *Forty-first International Conference on Machine Learning*, 2024.

[27] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4804–4814, 2022.

[28] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. Multisports: A multi-person video dataset of spatio-temporally localized sports actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13536–13545, 2021.

[29] Zhuolong Li, Xingao Li, Changxing Ding, and Xiangmin Xu. Disentangled pre-training for human-object interaction detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 28191–28201, 2024.

[30] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

[31] Yihong Ma, Ning Yan, Jiayu Li, Masood Mortazavi, and Nitesh V Chawla. Hetgpt: Harnessing the power of prompt tuning in pre-trained heterogeneous graph neural networks. In *Proceedings of the ACM Web Conference 2024*, pages 1015–1023, 2024.

[32] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[33] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1049–1059, 2020.

[34] Yujie Mo, Yajie Lei, Jialie Shen, Xiaoshuang Shi, Heng Tao Shen, and Xiaofeng Zhu. Disentangled multiplex graph representation learning. In *International conference on machine learning*, pages 24983–25005. PMLR, 2023.

[35] Yangjun Ou, Li Mi, and Zhenzhong Chen. Object-relation reasoning graph for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20133–20142, 2022.

[36] Junting Pan, Siyu Chen, Mike Zheng Shou, et al. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 464–474, 2021.

[37] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 401–417, 2018.

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

[39] Zhengxiang Shi and Aldo Lipani. Dept: Decomposed prompt tuning for parameter-efficient fine-tuning. *arXiv preprint arXiv:2309.05173*, 2023.

[40] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer, 2016.

[41] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

[42] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[44] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[45] Junke Wang, Dongdong Chen, Chong Luo, Bo He, Lu Yuan, Zuxuan Wu, and Yu-Gang Jiang. Omnivid: A generative framework for universal video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18209–18220, 2024.

[46] Ning Wang, Guangming Zhu, Hongsheng Li, Mingtao Feng, Xia Zhao, Lan Ni, Peiyi Shen, Lin Mei, and Liang Zhang. Exploring spatio–temporal graph convolution for video-based human–object interaction recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(10):5814–5827, 2023.

[47] Ning Wang, Guangming Zhu, HS Li, Liang Zhang, Syed Afaq Ali Shah, and Mohammed Bennamoun. Language model guided interpretable video action reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18878–18887, 2024.

[48] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–417, 2018.

[49] Yunbo Wang, Mingsheng Long, Jianmin Wang, and Philip S Yu. Spatiotemporal pyramid network for video action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1529–1538, 2017.

[50] T. Wu, R. He, G. Wu, et al. Sportshhi: A dataset for human-human interaction detection in sports videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18537–18546, 2024.

[51] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018.

[52] Boqiang Zhang, Hongtao Xie, Zuan Gao, and Yuxin Wang. Choose what you need: Disentangled representation learning for scene text recognition removal and editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 28358–28368, 2024.

[53] Ce Zhang, Changcheng Fu, Shijie Wang, Nakul Agarwal, Kwonjoon Lee, Chiho Choi, and Chen Sun. Object-centric video representation for long-term action anticipation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6751–6761, 2024.

[54] Jingran Zhang, Fumin Shen, Xing Xu, and Heng Tao Shen. Temporal reasoning graph for activity recognition. *IEEE Transactions on Image Processing*, 29:5491–5506, 2020.

[55] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9975–9984, 2019.

[56] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022.

[57] Xingyi Zhou, Anurag Arnab, Chen Sun, and Cordelia Schmid. How can objects help action recognition? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2353–2362, 2023.

[58] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6538–6547, 2020.

[59] Tao Zhuo, Zhiyong Cheng, Peng Zhang, Yongkang Wong, and Mohan Kankanhalli. Explainable video action reasoning via prior knowledge and state transitions. In *Proceedings of the 27th acm international conference on multimedia*, pages 521–529, 2019.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately describe the proposed framework Prompt-guided Disentangled Representation for Action Recognition for multi-action scene, and achieve the state-of-the-art results on multi-label benchmark.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide limitations in the appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: No theoretical results are provided in our method.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, we use publicly available datasets, and detailed code along with implementation details will be provided in the supplementary material to ensure full reproducibility of the main experimental results and to support the claims and conclusions of the pap

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Detailed code for reproducing the main experimental results are provided in the supplementary material. We also use publicly available datasets.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [No]

   Justification: The full details can be provided either with the code, in supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: Since some of the compared methods do not provide complete computational resource information (and some do not release their code), it is challenging for us to perform a fair comparison regarding computational cost.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Yes, I followed.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: We propose a prompt-guided disentangled representation for action recognition, which offers a novel way to disentangle actions. We believe it is natural for peers to build upon our work to improve it or apply it to other directions, such as temporal action localization. Therefore, we did not dedicate a separate section for discussion in the paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: Our paper does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

   Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

   Answer: [NA]

   Justification: Our paper does not release new assets.

   Guidelines:

   - The answer NA means that the paper does not release new assets.
   - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
   - The paper should discuss whether and how consent was obtained from people whose asset is used.
   - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

   Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

   Answer: [NA]

   Justification: Our paper does not involve crowdsourcing nor research with human subjects.

   Guidelines:

   - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
   - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
   - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

   Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

   Answer: [NA]

   Justification: Our paper does not involve crowdsourcing nor research with human subjects.

   Guidelines:

   - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
   - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Video Encoder

We adopt Dual-AI [15] as our video encoder, which is a two-stream architecture composed of interleaved temporal and spatial Transformers [7]. This design facilitates effective spatio-temporal information exchange through alternate modeling along temporal and spatial dimensions. On top of the original architecture, we further introduce additional spatial-to-spatial and temporal-to-temporal branches to enhance the flow of information across both dimensions. The overall architecture is illustrated in Figure 5. After the video encoder, an Multi-Layer Perceptron (MLP) fuses the two branches, and a Feed Forward Network (FFN) [43] further integrates the features to produce the input $\mathcal{F}_o^f$ for Video Graph Parsing Neural Network (VGPNN).

Both the Spatial Transformer and the Temporal Transformer adopt the standard self-attention encoder architecture [43] (as shown in Figure 6), with the primary difference lying in the dimension along which the attention is applied in spatial or temporal dimension, as shown in Figure 7.
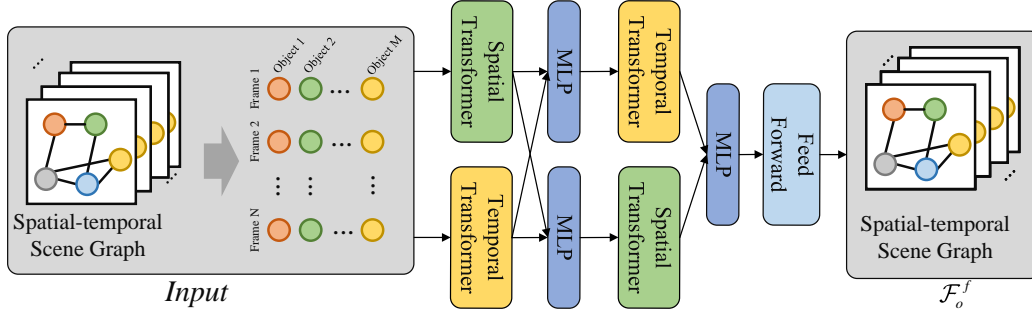

Figure 5: Our video encoder builds upon Dual-AI [15] by introducing additional spatial-spatial and temporal-temporal streams.
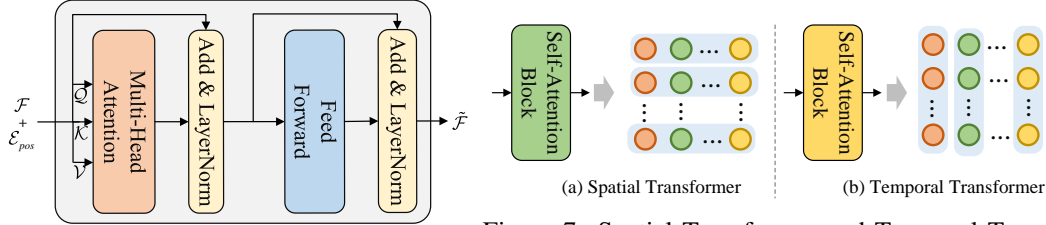

Figure 6: Architecture of self-attention encoder.

Figure 7: Spatial Transformer and Temporal Transformer

(a) Spatial Transformer

(b) Temporal Transformer

B Video Graph Parsing Neural Network

B.1 Formulation

In SSG, each frame of a video is processed into a graph, where nodes represent the objects present in that frame, and edges represent the semantic relationships between these objects (e.g., spatial or interaction relations). Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ denote the complete SSG constructed from the video, where $\mathcal{V}$ is the set of all object nodes across all frames, $\mathcal{E}$ is the set of all directed edges, and $\mathcal{R}$ is the set of all relation types. Each node $v \in \mathcal{V}$ is uniquely indexed from $\{1, \ldots, |\mathcal{V}|\}$. Each edge $e = (v, w, r) \in \mathcal{V} \times \mathcal{V}$ represents a directed semantic relation of type $r \in \mathcal{R}$ from node v to node w. Let $g_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{R}_i)$ denote the scene graph of the i-th frame, where $\mathcal{V}_i \subseteq \mathcal{V}$, $\mathcal{E}_i \subseteq \mathcal{E}$ and $\mathcal{R}_i \subseteq \mathcal{R}$ denote the frame-specific object nodes, their edges and relations. We denote v_i^j as the j-th object node in frame i, and $e_i^{j,k}$ as the edge from v_i^j to v_i^k with relation type $r_i^{j,k} \in \mathcal{R}_i$. The node features can be denoted as $\mathcal{F}_{\text{node}} = \{f_1^1, f_1^2, \ldots, f_N^M\} \in \mathbb{R}^{N \times M \times D}$, where $f_i^j \in \mathbb{R}^D$ represents the feature of node v_i^j in the j-th frame. Here, N, M, and D denote the number of frames in the video, the number of nodes per frame, and the dimensionality of each node feature, respectively.

It is important to note that the number of nodes in each frame is the same. The i-th node in every frame corresponds to the same object, though its visual features may vary across frames. This ensures that each node represents the same object across frames, even though its appearance changes. However, not every object appears in every frame, leading to padding where necessary.

Link. For edge $e_i^{j,k}$, we have N triples across frames, denoted as $\{(v_1^j, v_1^k, r_1^{j,k}), \ldots, (v_N^j, v_N^k, r_N^{j,k})\}$. To model the transition of object interactions over time, we first extract interaction features from these edge triples. Specifically, we concatenate the components of each edge triple and feed them into an edge function $\texttt{Edge}(\cdot)$ (implemented using a Multi-Layer Perceptron, MLPs; unless otherwise specified, all functions in subsequent modules are implemented as MLPs) to obtain frame-wise relation features $f^{j,k} = \{f_1^{j,k}, f_2^{j,k}, \ldots, f_N^{j,k}\}$. We then apply a self-attention encoder [43] to model the dependencies and transitions among these interaction features across frames to obtain interaction feature $\mathcal{F}^{j,k} = \{\mathcal{F}_1^{j,k}, \mathcal{F}_2^{j,k}, \ldots, \mathcal{F}_N^{j,k}\}$ as follow:

$$\mathcal{F}^{j,k} = \texttt{SelfAttEncoder}(\underbrace{\texttt{Edge}([v_1^j, v_1^k, r_1^{j,k}]), \ldots, ([v_N^j, v_N^k, r_N^{j,k}])}_{\text{relation features } f^{j,k}}) \tag{9}$$

where $[\cdot]$ represent concatenation, and $\texttt{SelfAttEncoder}(\cdot)$ is the self-attention encoder with L standard self-attention encoder blocks (as shown in Figure 6).

Message & Update. To facilitate better disentanglement, we employ dynamic weights to modulate the information propagated through different edges and nodes. This not only enables selective information flow, but also provides interpretability by highlighting the importance of each interaction. We through interaction feature $\mathcal{F}^{j,k}$ to obtain weight feature $\mathcal{F}_w^{j,k}$ as follows:

$$\mathcal{F}_w^{j,k} = \sigma(W^m \cdot \mathcal{F}^{j,k}) \tag{10}$$

where W^m denotes a linear transformation projecting to a scalar. We then combine the interaction features and node features to compute the messages $\mathcal{F}_m^j$ to be passed to node $v^j = \{v_1^j, v_2^j, \ldots, v_N^j\}$ in each frame, which are dynamically aggregated using the weight features $\mathcal{F}_w^j$, as follows:

$$\mathcal{F}_m^j = \sum_{k=0}^{K} \left(\mathcal{F}_w^{j,k} \cdot \texttt{Msg}([v^j, \mathcal{F}^{j,k}]) \right) \tag{11}$$

where K denotes the number of neighbors connected to node v^j, and $\texttt{Msg}(\cdot)$ represents the message function. After computing the message features $\mathcal{F}_m^j$, we use an aggregation function to incorporate them into the updated representation of node v_j, as follows:

$$\hat{v^j} = \texttt{Agg}(v^j \oplus \mathcal{F}_m^j) \tag{12}$$

where $\hat{v^j}$, $\texttt{Agg}(\cdot)$, and $\oplus$ denote updated node representation, the aggregation function, and element-wise addition, respectively. $\hat{v^j}$ replaces v^j in following processing.

Compared to previous work Graph Parsing Neural Network (GPNN) [37], VPGNN introduces stronger temporal modeling capabilities by modeling edges over time, rather than just generating aggregation weights for edges. VGPNN updates node features at each layer, while GPNN primarily updates edges and only updates nodes in the final layer. Additionally, VGPNN retains the full SSG across frames, unlike GPNN, which only preserves nodes from a single frame. This results in VGPNN offering better interpretability and generalization.

C Action Specification

C.1 Design

We assume that the label length of a dataset is 5. For a video labeled with classes $\{0, 1\}$, its multi-hot representation is:

$$[1, 1, 0, 0, 0] \tag{13}$$

Then, there will be three Specified Action Prompts (SAPs), corresponding to $\{0, 1\}$, $\{0\}$, and $\{1\}$, respectively (As shown in Eq. 14).

$$\begin{aligned} \{0, 1\} &\rightarrow [1, 1, 0, 0, 0], \\ \{0\} &\rightarrow [1, 0, 0, 0, 0], \\ \{1\} &\rightarrow [0, 1, 0, 0, 0] \end{aligned} \tag{14}$$

However, using only the SAP may lead the model to take shortcuts, focusing solely on the SAP rather than the video content. Therefore, we augment the SAP with distractor classes that are not associated with the video. In this case, the distractor set is $\{2, 3, 4\}$. Assuming **K=3**, we take the second example in Equation 14 (i.e., with label $\{0\}$)to illustrate: it will generate three **distractor-injected SAPs** (As shown in Eq. 15).

$$
\begin{aligned}
&[1, 0, 1, 1, 0], \\
&[1, 0, 1, 0, 1], \\
&[1, 0, 0, 1, 1]
\end{aligned}
\tag{15}
$$

Similarly, taking the first example in Eq 14 (i.e., with label $\{0, 1\}$), with label, it will also generate three **distractor-injected SAPs** (As shown in Eq 16).

$$
\begin{aligned}
&[1, 1, 1, 0, 0], \\
&[1, 1, 0, 1, 0], \\
&[1, 1, 0, 0, 1]
\end{aligned}
\tag{16}
$$

In both examples, the number of ones in the multi-hot vector is 3.

We divide the features into two parts: one corresponding to the specified action and the other to the remaining actions. If we feed the SAP into the part that is intended for unrelated actions, it will lead to information leakage, since the SAP contains the ground-truth labels. To avoid this, we instead feed the full label space of the dataset. Specifically, we obtain the Unspecified Action Prompts (UAP) by taking the complement of the SAP with respect to the entire label space, such that the combination of SAP and UAP covers all action categories in the dataset.

Taking Eq. 15 as an example, its SAP is given by:

$$
\begin{aligned}
&[0, 1, 0, 0, 1], \\
&[0, 1, 0, 1, 0], \\
&[0, 1, 1, 0, 0]
\end{aligned}
\tag{17}
$$

For Eq. 16, its SAP is given by:

$$
\begin{aligned}
&[0, 0, 0, 1, 1], \\
&[0, 0, 1, 0, 1], \\
&[0, 0, 1, 1, 0]
\end{aligned}
\tag{18}
$$

C.2 Construction

The AS (SAP and UAP pair) consists of two types: distractor-injected and non-distractor-injected. The distractor-injected AS includes a combination of present labels (i.e., labels of some actions in groundtruth labels) and absent labels (i.e., labels of some actions absent in the groundtruth labels), while non-distractor-injected only contains present labels.

For a video with the ground truth label set {Putting a pillow somewhere, Taking a pillow from somewhere, Throwing a pillow somewhere, Holding a pillow, Tidying up a blanket/s} corresponding to label indices $\{77, 79, 80, 76, 75\}$, and **presents labels are sample from this ground truth label set**. The remaining labels, i.e., all other labels from the full set $\{0, 1, \ldots, 156\}$ excluding the ground truth labels. **Absent labels are sampled from remaining labels.**

Under the non-distractor-injected setting, we generate 5 SAPs:

$$
\begin{aligned}
&[77], \\
&[80, 76], \\
&[77, 80, 75], \\
&[79, 77, 76, 75], \\
&[80, 77, 76, 75, 79]
\end{aligned}
\tag{19}
$$

Each SAP contains only present labels, with 1 to 5 ground truth labels randomly sampled from the ground truth label set. To prevent information leakage, **we use all labels outside each SAP as the**

corresponding UAP, thereby disentangling the rest of actions. Consequently, for the 5 SAPs, we generate 5 corresponding UAPs.

$$
\begin{aligned}
&\text{4 present labels: } [\underbrace{\textbf{80,76,75,79}, 0, 1, \ldots, 156}_{156 \text{ labels}}], \\
&\text{3 present labels: } [\underbrace{\textbf{77,79,75}, 0, 1, \ldots, 156}_{155 \text{ labels}}], \\
&\text{2 present labels: } [\underbrace{\textbf{79,76}, 0, 1, \ldots, 156}_{154 \text{ labels}}], \\
&\text{1 present labels: } [\underbrace{\textbf{80}, 0, 1, \ldots, 156}_{153 \text{ labels}}], \\
&\text{0 present labels: } [\underbrace{0, 1, \ldots, 156}_{152 \text{ labels}}]
\end{aligned}
\tag{20}
$$

Under the distractor-injected setting, **we fix the length of each SAP to 16**, and generate $5 + 1 = 6$ SAPs:

$$
\begin{aligned}
&\text{0 present labels:} [152, 32, 24, 125, 145, 110, 93, 138, 129, 44, 124, 130, 85, 7, 5, 0], \\
&\text{1 present labels:} [\textbf{76}, 143, 111, 19, 104, 23, 112, 12, 78, 146, 56, 136, 116, 9, 24, 125], \\
&\text{2 present labels:} [\textbf{77, 76}, 140, 111, 134, 130, 23, 18, 155, 136, 88, 133, 112, 127, 17, 2], \\
&\text{3 present labels:} [\textbf{76, 79, 75}, 47, 125, 91, 71, 39, 66, 141, 9, 27, 52, 1, 132, 54], \\
&\text{4 present labels:} [\textbf{76, 79, 75, 77}, 56, 88, 120, 22, 128, 62, 52, 105, 142, 124, 153, 35] \\
&\text{5 present labels:} [\textbf{80, 77, 76, 75, 79}, 17, 51, 109, 143, 103, 57, 15, 152, 130, 125, 47]
\end{aligned}
\tag{21}
$$

Each SAP contains both present and absent labels, with 0 to 5 ground truth labels randomly sampled from the original ground truth label set, and the rest 16 to 11 labels sampled from the remaining labels. To prevent information leakage, **we use all labels other than those in each SAP as the corresponding UAP**, thereby disentangling the other actions. **Each UAP has a length of 141**. Consequently, for the five SAPs, we generate 6 corresponding UAPs, each of length 141.

$$
\begin{aligned}
&\text{5 present labels: } [\underbrace{\textbf{80,76,77,75,79}, 1, 2, \ldots, 156}_{141 \text{ labels}}], \\
&\text{4 present labels: } [\underbrace{\textbf{80,77,75,79}, 0, 1, \ldots, 156}_{141 \text{ labels}}], \\
&\text{3 present labels: } [\underbrace{\textbf{80,79,75}, 0, 1, \ldots, 156}_{141 \text{ labels}}], \\
&\text{2 present labels: } [\underbrace{\textbf{80,77}, 0, 2, \ldots, 156}_{141 \text{ labels}}], \\
&\text{1 present labels: } [\underbrace{\textbf{80}, 0, 1, \ldots, 156}_{141 \text{ labels}}], \\
&\text{0 present labels: } [\underbrace{0, 1, \ldots, 156}_{141 \text{ labels}}]
\end{aligned}
\tag{22}
$$

Finally, for a video with 5 labels, we generate $2 \cdot 5 + 1 = 11$ ASs (11 SAP and UAP pairs).

25

Table 10: This table outlines the division of the Charades dataset into five subsets, ensuring that there is no overlap between the scenes used for training and testing

Subdataset	Training Scene	Test Scene
Scenario1	Stairs,Laundry room,Home Office, Hallway,Bedroom,Pantry,Dining room,Entryway	Living room,Closet,Kitchen,Bathroom, Garage,Recreation room,Basement,Other
Scenario2	Laundry room,Bathroom,Pantry,Closet, Entryway,Recreation room,Garage,Other	Bedroom,Living room,Kitchen,Home Office, Hallway,Stairs,Basement,Dining room
Scenario3	Stairs,Laundry room,Bedroom,Basement, Bathroom,Entryway,Recreation room,Other	Living room,Closet,Kitchen,Home Office, Garage,Hallway,Pantry,Dining room
Scenario4	Kitchen,Stairs,Laundry room,Home Office, Bedroom,Bathroom,Pantry,Dining room	Living room,Closet,Garage,Hallway, Recreation room,Entryway,Basement,Other
Scenario5	Kitchen,Laundry room,Hallway,Basement, Dining room,Living room,Closet,Other	Bedroom,Home Office,Bathroom,Garage, Stairs,Recreation room,Entryway,Pantry

Table 11: Domain shift experiment on the Charades dataset

Method	mAP (%)
STIGPN [46]	54.1
LaIAR [47]	57.2
Ours	**63.0**

D   Details and More Results for Our Experiment

D.1   Additional Experiment

We conduct domain shift experiments on the Charades dataset to evaluate the generalization ability of our method. Specifically, we follow the experimental setup of [47], splitting Charades into five disjoint subsets with different action groups. The details are shown in Table 10.

As shown in Table 11, our method achieves the best performance under domain shift settings, indicating strong generalization ability across different scenarios. We attribute this to the effective disentanglement of features, which allows the model to better capture action-relevant information, thereby improving action understanding and enhancing generalization performance in unseen environments.

D.2   More Details

For each video, we uniformly sample 16 frames as input. We adopt a two-stage training strategy for all experiments. The first stage (pre-train) runs for 5 epochs with a learning rate of 2e-4, aiming to learn the representations for a_u and a_s. In the second stage (post-train), we load the checkpoint from the epoch with the best performance on a_u and a_s in the pre-training stage, and only train the classifier for a_t. This stage is also trained for 5 epochs with a learning rate of 1e-4. We report the best-performing epoch from the second stage. The detailed results are shown in Table 12 13. Except for our method, other approaches generally show a certain degree of performance improvement across different settings. Notably, our method consistently achieves the best performance in all scenarios, demonstrating its robustness and superior generalization capability.

E   Visualization

Figure 8 shows the action disentanglement localization results for two different videos (with 3 and 4 action labels respectively). Both videos contain concurrent or individual interactions. Figure 9 illustrates the process by which our method disentangles the specified actions from the concurrent interactions.

Table 12: Ablation of normalization

norm type	Post-train a_tmAP (%)	Pre-train a_tmAP (%)
LayerNorm [1]	68.12	65.44
BatchNorm [18]	65.39	63.05
GraphNorm [3]	68.61	65.32
VGNorm	**71.10**	**71.32**

Table 13: Ablation study of loss function on Charades dataset. ❶ denotes using ProDA with only the action classification loss.

	$\mathcal{L}_{res}$ w/o margin	$\mathcal{L}_{res}$	$\mathcal{L}_{dis}$ w/o margin	$\mathcal{L}_{dis}$	Post-train a_tmAP (%)	Pre-train a_tmAP (%)
❶	-	-	-	-	67.45	65.83
❷	✓	-	✓	-	68.43	18.27
❸	-	-	-	✓	67.78	66.92
❹	-	✓	-	-	64.89	61.19
❺	-	✓	-	✓	**71.10**	**71.32**

Action: *Holding a vacuum*


Action: *Turning off a light*

Figure 8: Frame-level weights s predicted by the readout function for different non-distractor-injected SAPs within the same video. The heatmaps (*colored bars*) visualize the predicted weights, reflecting the relevance of each frame to the target action. The *red bars* indicate the ground-truth action segments, while the *green bars* denote our localization results obtained by thresholding s at 0.7.

In Figure 10, we show a more complex example from a video with 15 labels, containing multiple concurrent interactions. In this example, the person interacts simultaneously with a table, chair, cup, book, and food. Using prompts, we disentangle the interaction between the person and the shoe from these overlapping interactions. Additionally, the subgraphs relevant to the specified action are successfully parsed from the full SSG. In Figure 11, we present an example from a video with 10 labels, which also contains multiple concurrent interactions. In this example, the person interacts simultaneously with a sofa, blanket, TV, and food. Using prompts, we disentangle the interaction between the person and the shoe from these overlapping interactions. Additionally, the subgraphs relevant to the specified action are successfully parsed from the full SSG.

F Limitation

Our method operates solely on visual features and the derived Spatio-Temporal Scene Graphs (SSGs), without incorporating other modalities such as audio, text, or human pose. This reliance on a single modality may limit the flexibility and robustness of the model, especially in scenarios where multimodal cues are essential for disambiguating actions. In future work, we plan to extend our framework to integrate multimodal information, enabling more comprehensive and robust action disentanglement in complex environments.
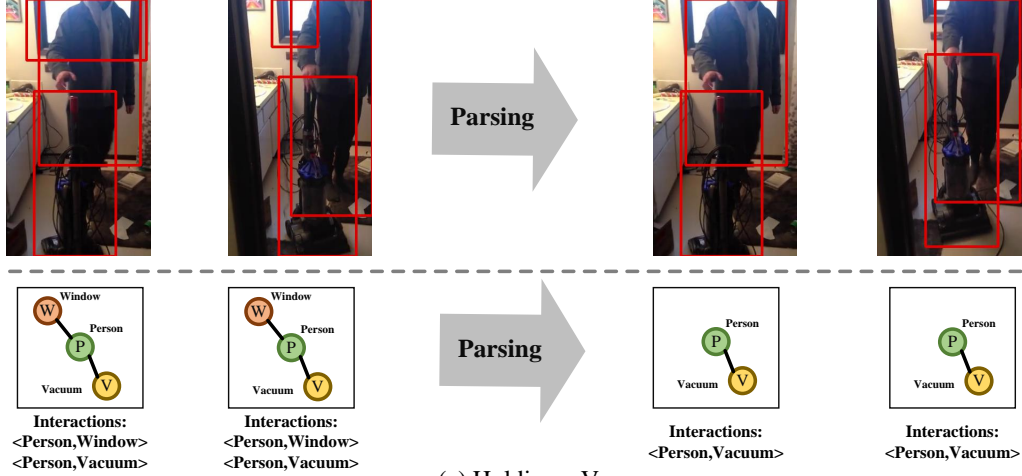
(a) Holding a Vacuum
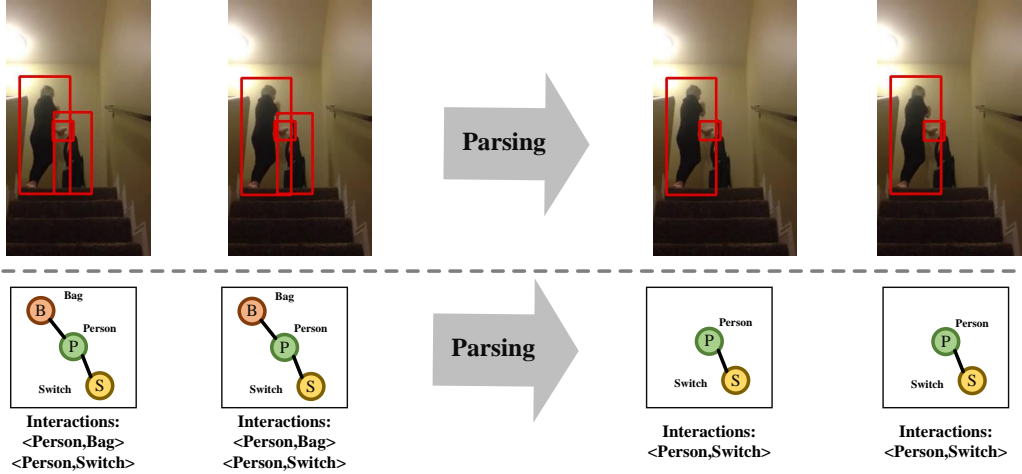
(a) Turn off a Light

Figure 9: Key parsing steps for the two examples presented in Figure 8. For each case, the specified action is disentangled from multiple co-occurring interactions. The red boxes indicate the objects retained by VGPNN during the parsing process. Initially, the scene graph is fully connected with all nodes; after parsing, only a subset of relevant nodes remains. Below each image is the corresponding scene graph of that frame.
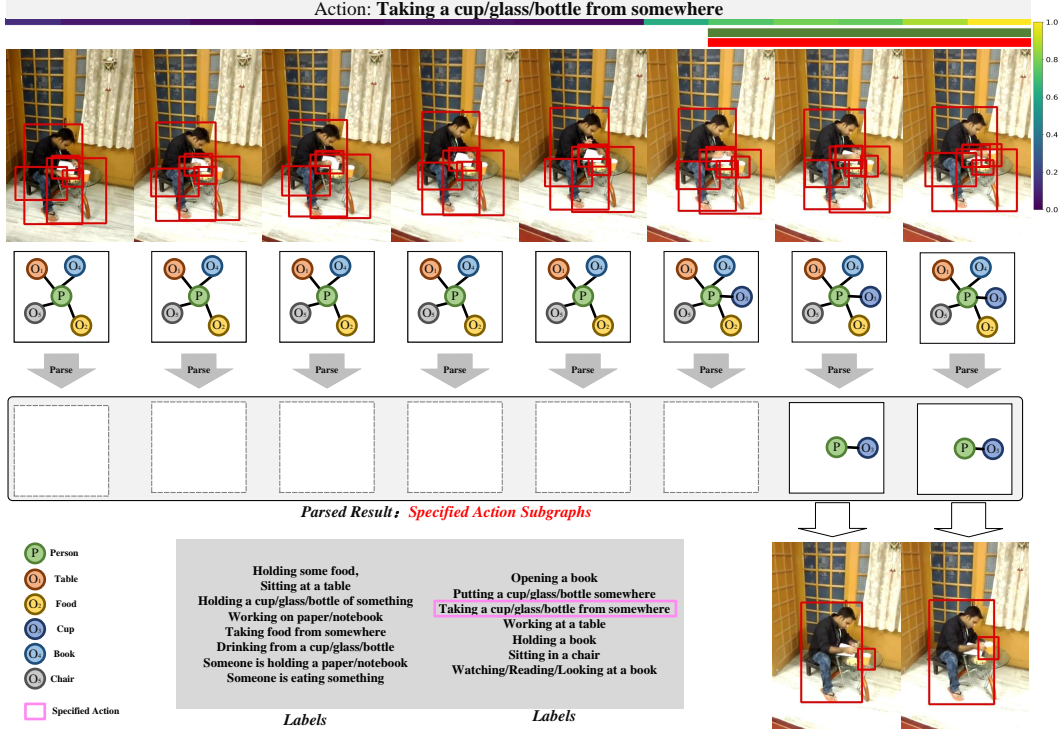
Figure 10: This is the parsing process of a 15 labels video. We disentangle a single interaction from a series of complex interactions in the scene by using prompts. The top part of the image shows our Temporal Action Localization results.
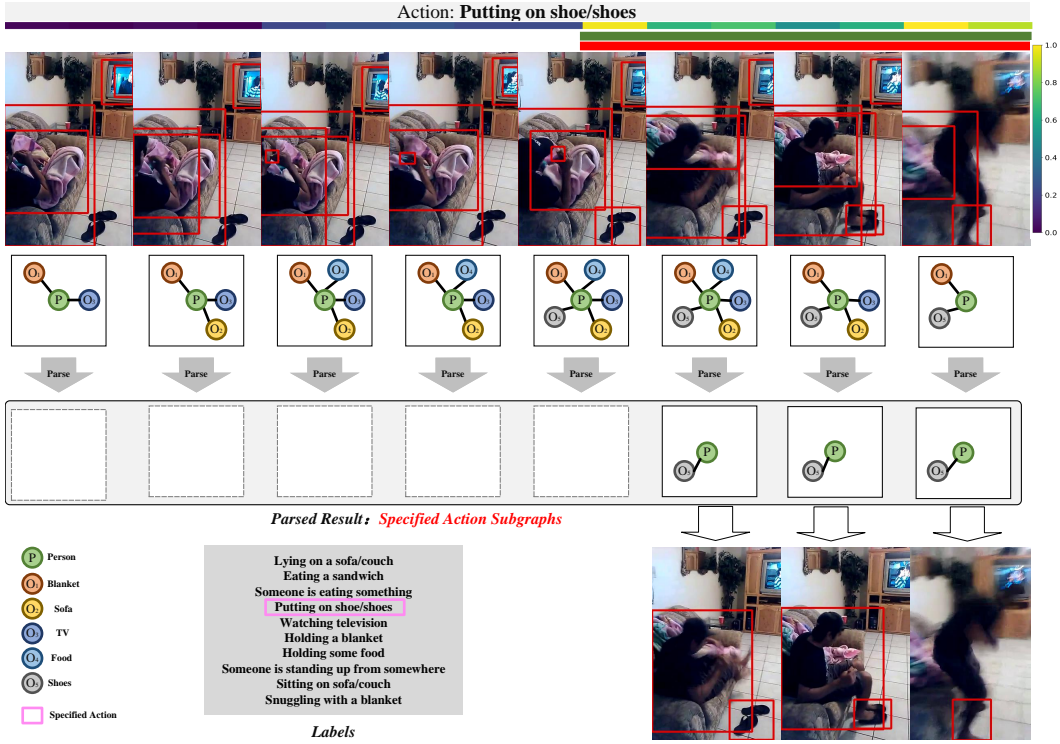

Figure 11: This is the parsing process of a 10 labels video. We disentangle a single interaction from a series of complex interactions in the scene by using prompts. The top part of the image shows our Temporal Action Localization results.