PromptGAR: Flexible Promptive Group Activity Recognition

Zhangyu Jin[1] Andrew Feng[1] Ankur Chemburkar[1] Celso M. De Melo[2]

[1] University of Southern California, Institute for Creative Technologies
[2] Army Research Laboratory

{zjin,feng,achemburkar}@ict.usc.edu, celso.m.demelo.civ@army.mil

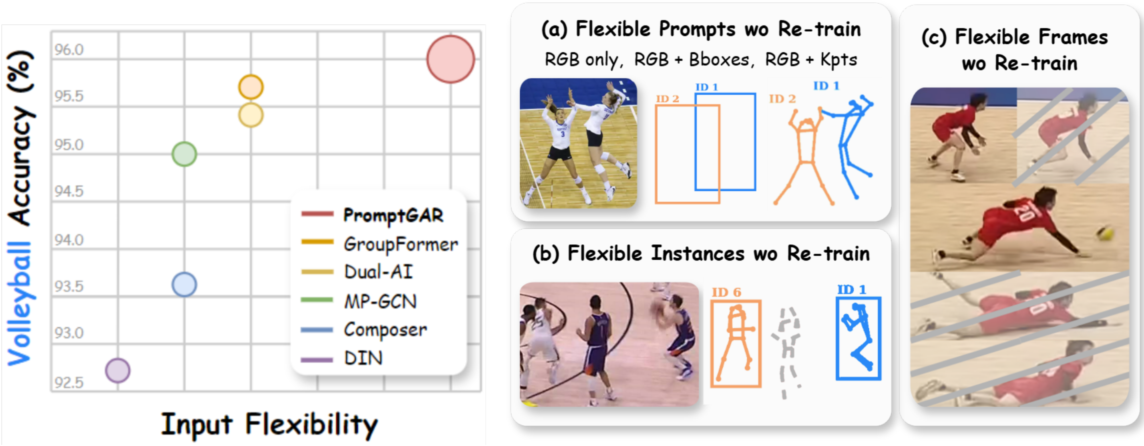

Figure 1. This diagram illustrates PromptGAR's performance and key capabilities. Compared to existing methods, PromptGAR achieves competitive group activity recognition accuracy and superior input flexibility without the need for retraining, including: *(a)* flexible visual prompt inputs, *(b)* flexible instance counts, and *(c)* flexible frame sampling.

Abstract

*We present **PromptGAR**, a novel framework for Group Activity Recognition (GAR) that offering both input flexibility and high recognition accuracy. The existing approaches suffer from limited real-world applicability due to their reliance on full prompt annotations, fixed number of frames and instances, and the lack of actor consistency. To bridge the gap, we proposed PromptGAR, which is the first GAR model to provide input flexibility across prompts, frames, and instances without the need for retraining. We leverage diverse visual prompts—like bounding boxes, skeletal keypoints, and instance identities—by unifying them as point prompts. A recognition decoder then cross-updates class and prompt tokens for enhanced performance. To ensure actor consistency for extended activity durations, we also introduce a relative instance attention mechanism that directly encodes instance identities. Comprehensive evaluations demonstrate that PromptGAR achieves competitive performances both on full prompts and partial prompt inputs, establishing its effectiveness on input flexibility and generalization ability for real-world applications.*

1. Introduction

Group Activity Recognition (GAR) [7] is fundamentally important for video and event understanding, and it is widely used in areas such as video analytics, human computer interaction, and security systems. GAR processes videos and annotations including bounding boxes, skeletons, ball trajectories, and optical flows to determine the group activity label. Building on prior research [13, 16, 24, 26, 51], we aim to not only enhance group activity recognition accuracy but also to create a more flexible architecture handling diverse prompt, frame, and instance inputs.

Although numerous models have been proposed, high-performance and flexible group activity recognition continues to be challenging due to the following difficulties:

Requirement for Full Prompts. Current group activity recognizers [13, 16, 24] rely on full annotations to achieve strong performance at test time. It is widely acknowledged that obtaining accurate annotations is difficult.

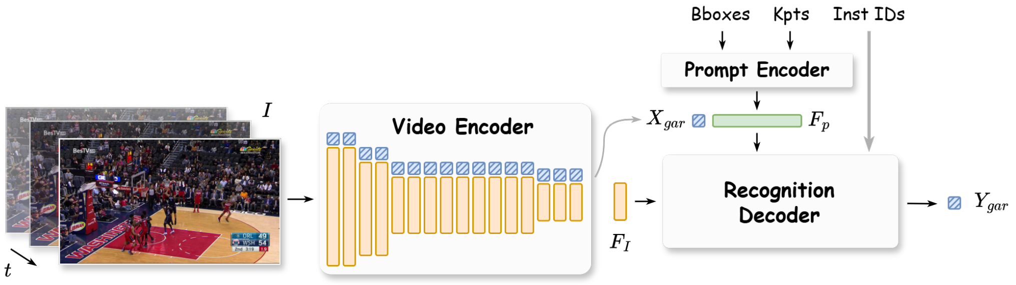

Figure 2. **PromptGAR Architecture**. A sequence of frames I is processed by the video encoder, yielding RGB features $\mathbf{F}_I$ and GAR class token $\mathbf{X}_{gar}$. Prompts, such as bounding boxes and skeletal keypoints, are transformed to prompt tokens $\mathbf{F}_p$ by the prompt encoder. These tokens, along with instance identities, are then fed into the recognition decoder to get group activity prediction $\mathbf{Y}_{gar}$.

Even though using state-of-the-art object detectors [35, 52], trackers [34, 44], and pose estimators [14, 40], the lower quality annotations still exist, such as missed detections, redundant boxes with low confidence scores, player ID switching, player ID reassignment upon reappearance, and so on. In real-world scenarios, manually correcting these annotations at test time is often impractical. Instead of being forced to use potentially inaccurate prompts during test time, users in real-world scenarios want the flexibility to choose from full, partial, or no prompts. However, current GAR architectures [13, 16, 24, 26, 30, 32, 45, 51] either do not support partial prompts, or they demand retraining to perform reasonably. So we design our model for the input flexibility: it is trained with full prompts like prior methods, but it delivers competitive results with full prompts and still performs quite well with fewer or no prompts at test time, without retraining.

Requirement for Fixed Frames and Instances. Most current GAR methods [13, 16, 24] are limited to a fixed number of frames and instances as input. Those rigid requirements lead to significant performance degradation in real-world scenarios: (1) As seen in Fig. 3-a, offense or defense relies on who gets the ball at the end. Such key moment can be anywhere in the video, so previous fixed-frame models may miss it depending on how the frames are sampled from a long sequence. (2) Similarly, in Fig. 3-b and c, real-world data often contains missing actors in annotations or false positive detections of spectators as players. Prior fixed-instances models requires a fixed input shape, causing runtime errors or requiring arbitrary padding/truncation that degrades performance. In contrast, our model is designed to accept a flexible number of frames and instances, and it achieves reliable performance without requiring retraining.

Lacking Actor Consistency. Recent GAR approaches [26, 49, 51] rely on a fixed player order. Their performance degrades when this order changes. Even though player order in the testing set of either Volleyball [18] or NBA [42] is not supposed to be known, that order is still fixed for both validation and testing in these methods. Consequently, among all epochs, they pick the checkpoint with the best performance under this specific player order. However, in real-world scenarios, such player order is entirely unknown. A robust GAR model, therefore, should give identical performance regardless of the input player order. In our work, we address this by introducing a relative instance attention mechanism that only encodes whether two players are the same or not. This design ensures that player order does not affect our model's performance at all.

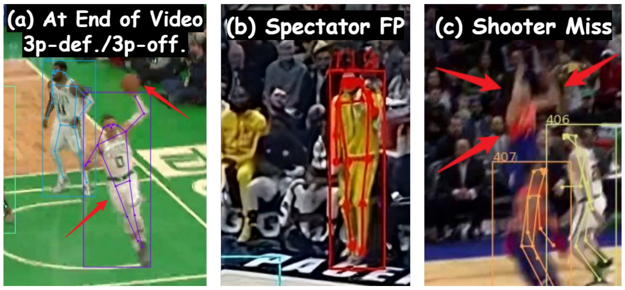

Figure 3. Necessity of flexible frames: (*a*) offense or defense relies on which team gets the ball at the end, and similar key moments can be anywhere in the video. Necessity of flexible instances: (*b*) false-positives and (*c*) false-negatives make player counts unfixed.

Considering the above challenges and motivations, we present **PromptGAR**, a transformer-based group activity recognition framework that leverages diverse visual prompts (i.e., bounding boxes, skeletal keypoints, instance identities) to achieve high group activity recognition accuracy and input flexibility. (a) **Flexible Prompts**. It adapts to varying prompt availability, namely full prompts, partial prompts, and no prompts. When comprehensive annotations are present, the full prompt inputs would maximize performance. If only simpler annotations like bounding boxes and instance identities are available, our model still gives reliable results. Even with no annotations, it can also provide reasonable results using only the raw video. This adaptability makes it suitable for diverse real-world scenarios where annotation quality varies. (b) **Flexible Frames**. The method has temporal flexibility in accepting videos of varying lengths and frame rates. Furthermore, it effectively recognizes group activities regardless of their temporal location within the video, accommodating both instantaneous actions and sequential events. (c) **Flexible Instances**. It is designed to handle varying numbers of actors within a scene. Therefore in an ideal scenario, our model can leverage detailed annotations of all individuals for enhanced accuracy. On the other hand, it can also maintain robust performance even when there are missing annotations for certain actors. (d) **Without Retraining**. After training with full prompts, PromptGAR automatically gains the flexibility for input prompts, frames, and instances during inference. This significantly enhances its practicality and broadens its applicability to diverse scenarios.

Our implementation is inspired by the Segment Anything [21]. We unify bounding boxes and skeletal keypoints as point prompts and employing a two-way decoder for cross-updating class and prompt tokens. **Firstly**, the input flexibility is implemented through several design features: MViTv2's relative positional embedding for variable length of RGB frames, depth-wise prompt pooling to accommodate flexible temporal and prompt dimensions, and a robust head that maintains classification stability even when no prompts are available during inference. **Secondly**, to effectively acquire actor consistency, we introduce relative instance attention, which directly encodes instance IDs. The encoding ensures that the output will remain invariant to instance ID transitions.

Based on the above technical contributions, we evaluate our model under Volleyball [18] and NBA [42] datasets, PromptGAR produces competitive results compared to state-of-the-art GAR methods, showcasing its strong recognition capabilities. Additionally, we demonstrated its flexibility to handle varying prompt inputs, frame sampling, and instance counts while maintaining robust performance without retraining. Our main contributions are as follows.

(*a*) An effective group activity recognition architecture that achieves input flexibility across prompts, frames, and instances without retraining.

(*b*) A relative instance attention module for encoding instance identities and ensuring actor consistency.

2. Related Work

2.1. Group Activity Recognition

GAR [7] has attracted attention due to its massive success in a variety of real-world applications. Earlier techniques relied heavily on handcrafted features [5–8, 15, 22, 28] or AND-OR graphs [1, 2, 33]. Recently, methods based on neural network architectures have been widely studied because of their ability to effectively extract features and fuse various visual prompts.

Various Prompts for GAR. (a) *RGB frames and bounding boxes*. Existing work either directly crops people from scenes [41] or applies ROIAlign [17] to represent actors from extracted feature maps [24, 32, 45]. (b) *Skeletons*. Several works [11, 26, 37, 51] explore using human skeleton joints as inputs to avoid substantial computational resources and discrepancies in background and camera settings in video-based approaches. (c) *Optical flows*. Neighboring RGB frames can produce optical flow images [47]. These flows are then concatenated with RGB frames to create dense inputs to introduce pixel-wise motion information across the temporal dimension [16, 24, 32]. A similar idea is also applied in Composer [51] for skeletal information by computing the temporal difference of coordinates in two consecutive frames. MP-GCN [26] calculated joint motions and bone motions as extra sparse inputs. (d) *Balls*. Group labels in the NBA dataset [42], such as '*2p-succ*' and '*3p-succ*', are closely related to ball positions; therefore, Composer [51] and MP-GCN [26] treat balls as additional inputs. However, it is difficult to leverage these specific model designs for general scenarios such as non-sports activity recognition. (e) *Tracked instance identities*. Recent approaches [24, 45] use optical flows to describe only short-term motions without explicitly encoding instance identities to preserve actor consistency. We introduce a relative instance identities encoding mechanism to handle this long-neglected prompt.

Input Flexibility for GAR. Due to the model design choices, recent works [13, 16, 24, 26, 51] have less input flexibility in group activity recognition. (a) *Flexible prompts*. A common limitation among GAR methods [13, 16, 24] that process RGB frames and bounding boxes is their dependence on ROIAlign [17] for actor feature extraction, making them incompatible with scenarios lacking bounding box prompts. Likewise, GroupFormer's fixed input channel setting [24], achieved through concatenating RGB and skeleton features, prevents its use when skeleton data is unavailable. (b) *Flexible frames*. Dual-AI [16] demonstrates some flexibility by applying different frame sampling strategies during training and testing, such as 3 frames for training and 9 or 20 frames for testing on the Volleyball [18] and NBA [42] datasets, respectively. However, these frame counts remain significantly lower than the total available frames (41 for Volleyball, 72 for NBA) in each clip. This restricted frame sampling flexibility limits its potential for achieving higher performance. (c) *Flexible instances*. Composer [51] relies on normalizing joint coordinates with statistics derived from the complete clip. However, missing instances significantly alter these statistics, shifting the mean and standard deviation far from their expected distribution. Notably, our PromptGAR main-

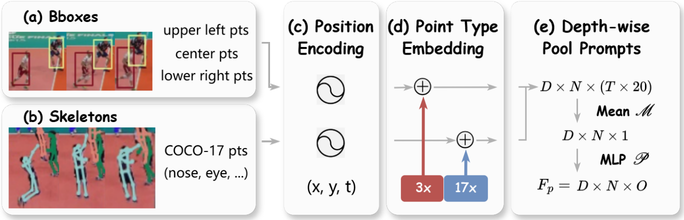

Figure 4. **Prompt Encoder**. (*a*) Bounding boxes are represented by 3 points (upper-left, center, lower-right). (*b*) Skeletal keypoints consist of 17 points. (*c*) Positional encoding captures both spatial and temporal coordinates. (*d*) Point types are distinguished using learnable embeddings. (*e*) Depth-wise prompts pooling reduces temporal and type dimensions to 1, then up-projects to the number of pooled prompts O.

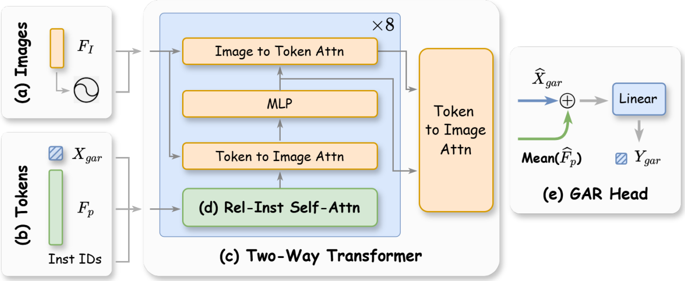

Figure 5. **Recognition Decoder**. The decoder processes (*a*) RGB features $\mathbf{F}_I$ with positional embeddings and (*b*) the GAR class token $\mathbf{X}_{gar}$ and prompt tokens $\mathbf{F}_p$. (*c*) The Two-Way Transformer performs cross-updating between these features. (*d*) Relative instance self-attention, using instance identities, ensures actor consistency. (*e*) The GAR head takes updated GAR class token $\widehat{\mathbf{X}}_{gar}$ and average of updated prompt tokens $\widehat{\mathbf{F}}_p$ to predict the group activity label $\mathbf{Y}_{gar}$.

tains high performance across inputs with flexible prompts, frames, and instances without the needs for retraining.

3. Method

As shown in Fig. 2, PromptGAR is an end-to-end prompt-based framework for group activity recognition. It takes a sequence of frames and corresponding prompts as inputs and outputs the group activity label. In the following subsection, we first introduce how to encode videos and visual prompts in §3.1. Then we illustrate how to fuse various visual prompts and spatial-temporal information in §3.2.

3.1. Visual Inputs Encoding

PromptGAR supports three kinds of inputs: input frames $\mathbf{I}$, bounding box prompts $\mathbf{F}_{box}$, and skeleton prompts $\mathbf{F}_{kpt}$.

Video Encoding. While existing works [16, 45] have shown promises in their power to model RGB features, they focus on images rather than videos. Other studies [24, 32] chose I3D [4], but they require optical flows as extra dense guidance. To address those issues, the sequence of frames $\mathbf{I}$ is processed through MViTv2 [25], to extract multi-scale feature tokens $\mathbf{F}_I$ and the GAR class token $\mathbf{X}_{gar}$.

Point Prompts. As shown in Fig. 4, all of the aforementioned prompts are formulated as point prompts: (a) Bounding boxes are formatted in three points, including upper-left, center, and lower-right points, similarly as in [21]. (b) Human skeletons follow the definition of COCO keypoints [27]. In general, one point is uniquely described by five attributes:

$$(x, y, t, p, ID)$$

where x, y, t are the positions along the width, height, and temporal axes, p is the point type, and *ID* denotes the instance identity that the point belongs to. We encode these attributes in the following ways. *Firstly*, while the original Fourier embedding [36] only maps the spatial coordinate (x, y) to the corresponding feature dimensions, we extend its capability to incorporate temporal location t.

$$\gamma(\mathbf{v}) = \Big[\cos\Big(2\pi\mathbf{B}(2\mathbf{v}-1)\Big), \sin\Big(2\pi\mathbf{B}(2\mathbf{v}-1)\Big)\Big]^T$$

where $\mathbf{v} := [x, y, t]^T \in [0, 1]^3$ is the spatial-temporal coordinate, and $\mathbf{B} \in \mathbb{R}^{\lfloor D/2 \rfloor \times 3}$ is sampled from a Gaussian distribution $\mathcal{N}(0, 1)$. The same positional encoding is also applied to RGB features $\mathbf{F}_I$. *Secondly*, we employ learned embeddings for each prompt type p, similar to [21]. *Thirdly*, *ID* is encoded by relative embeddings, with details in § 3.2.

Depth-wise Prompts Pooling. As described in Fig. 4-(e), after obtaining prompt features $\mathbf{F}_{box}$ and $\mathbf{F}_{kpt}$, we employ a mean pooling operator $\mathcal{M}$ and a MLP projector $\mathcal{P}$. They perform along temporal and type dimensions.

$$\mathbf{F}_p = (\mathcal{P} \circ \mathcal{M})([\mathbf{F}_{box}; \mathbf{F}_{kpt}])$$

where $\mathbf{F}_p \in \mathbb{R}^{D \times N \times O}$, $\mathbf{F}_{box} \in \mathbb{R}^{D \times N \times (T \times 3)}$ and $\mathbf{F}_{kpt} \in \mathbb{R}^{D \times N \times (T \times 17)}$. And D, N, T, O are the embedding size, number of instances, number of frames, and number of pooled prompts, respectively.

The pooling mechanism reduces the computational complexity and avoids out-of-memory (OOM) in the recognition decoder. The depth-wise mechanism, namely the mean operator $\mathcal{M}$ that reduces channels from $T \times 20$ to 1, is specifically designed for input flexibility across temporal and prompt dimensions. Consequently, a model trained on T_1 frames can perform inference on T_2 frames ($T_2 \neq T_1$) without requiring architectural modifications or weight changes. Similarly, the model is able to infer with either bounding box prompts alone or skeleton prompts alone, even when trained on both prompts. Note that the instance dimension N is not pooled and is required by relative identities encoding in §3.2.

3.2. Cross-Visual Decoding

Leveraging the image embedding $\mathbf{F}_I$, the prompt features $\mathbf{F}_p$ and associated instance identities, PromptGAR takes the GAR class token $\mathbf{X}_{gar}$ to decode the GAR logits $\mathbf{Y}_{gar}$.

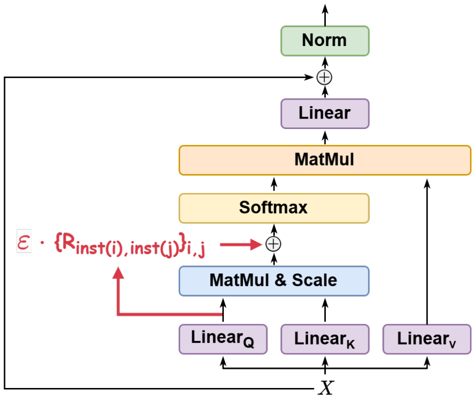

Figure 6. The **Relative Instance Attention** mechanism that incorporating a constant scale ϵ and relative instance identity embeddings $\mathcal{R}_{\text{inst}(i),\text{inst}(j)}$ in the attention block.

Two-Way Transformer. For simplicity, we refer to these embeddings (not including the image embedding) collectively as "tokens".

$$\texttt{images} = \mathbf{F}_I, \quad \texttt{tokens} = [\mathbf{X}_{gar}; \mathbf{F}_p]$$
$$\widehat{\mathbf{F}}_I, [\widehat{\mathbf{X}}_{gar}; \widehat{\mathbf{F}}_p] = \text{Two-Way}(\texttt{images}, \texttt{tokens})$$

Our two-way transformer is shown in Fig. 5-(c), inspired by [21], each layer performs 4 steps: (1) relative instance self-attention on tokens, (2) cross-attention from tokens to the image embedding, (3) an MLP updates each token, and (4) cross-attention from the image embedding to tokens. The next layer takes the updated tokens and the updated image embedding from the previous layer. Input flexibility is well-achieved through the attention mechanism, which naturally handles inputs of different lengths. This allows our two-way transformer to process varying numbers of frames, types, or instances at inference time, regardless of those in training.

Relative Instance Attention. Existing research has demonstrated potential in modeling short-term temporal consistency via either optical flows [16, 24, 32] or joint motions [26, 51]. As these methods primarily focus on capturing motion between immediate time steps, the long-term consistency in the movement of objects or individuals is not well handled. Also, the use of absolute instance *ID* encoding to ensure consistency violates the principle of shift invariance [23]. Namely, the interaction between two prompts becomes dependent on their arbitrary instance *ID*s, even if they refer to the same underlying object or entity. Inspired by relative positional embedding, we introduce the relative instance identity embedding to address this issue by focusing on whether two tokens belong to the same instance.

As illustrated in Fig. 6, we encode the relative instance information between the two input elements, i and j, into embedding $\{R_{\text{inst}(i),\,\text{inst}(j)}\}_{i,j} \in \mathbb{R}^D$, where $\text{inst}(i)$ and $\text{inst}(j)$ denote the instance *ID* of element i and j. Notice that the GAR class token $\mathbf{X}_{gar}$ is not included in $E^{(rel)}$.

$$\text{Attn}(Q, K, V) = \text{Softmax}\Big((QK^T)/\sqrt{d} + \epsilon \cdot E^{(rel)}\Big)V,$$

$$\text{where} \quad E^{(rel)}_{i,j} = Q_i \cdot \{R_{\text{inst}(i),\,\text{inst}(j)}\}_{i,j}$$

Here, the number of token pairs belonging to different instances significantly outnumbers those belonging to the same instance. To address the imbalance problem, we assign a learnable embedding R to pairs of tokens within the same instance, while setting the embedding to 0 for pairs from different instances.

$$\{R_{\text{inst}(i),\,\text{inst}(j)}\}_{i,j} = \begin{cases} 0, & \text{inst}(i) \neq \text{inst}(j) \\ R, & \text{inst}(i) = \text{inst}(j) \end{cases}$$

Due to the sparsity of $E^{(rel)}$, the scaling factor ϵ is a constant determined by experiments, with details in §4.5. Note that relative instance embeddings cannot be applied to cross-attention between image embeddings and tokens, as image embeddings lack associated instance *ID* information.

GAR Head. In contrast to most approaches [10, 21, 25, 43], where `[class]` tokens are directly fed to MLP layers for logits prediction, we observe a unique challenge in GAR. While traditional methods align different prompts with distinct ground truth labels [21], in GAR, various prompts correspond to the same ground truth label. As illustrated in Fig. 5-(e), to encourage the network to utilize the information provided by the prompts, we add the class tokens with the averaged prompt features. Then a linear projection is employed to generate final logits for prediction.

$$\mathbf{Y}_{gar} = \text{Linear}\Big(\widehat{\mathbf{X}}_{gar} + \text{Mean}(\widehat{\mathbf{F}}_p)\Big)$$

where $\text{Mean}(\widehat{\mathbf{F}}_p) \in \mathbb{R}^D$ means taking average over $T \times O$. Our approach is also carefully designed to ensure input flexibility, even when the prompts present during the training time are absent during the inference time. During back-propagation, the gradients of $\widehat{\mathbf{X}}_{gar}$ and $\text{Mean}(\widehat{\mathbf{F}}_p)$ are identical, indicating they are optimized in the same direction and towards the same representation. Consequently, a well-trained model can function effectively using either $\widehat{\mathbf{X}}_{gar}$ or $\text{Mean}(\widehat{\mathbf{F}}_p)$ independently.

4. Experiments

4.1. Experimental Setup

Volleyball Dataset [18] contains 3,493 clips for training and 1,337 clips for testing. Each clip has 41 frames and is labeled with one of eight group activities. Bounding box and skeleton annotations, provided separately by [18, 51], are available only for the central 16 frames of each clip.

Method	Prompt Types					Top1	Mean
	RGB	Bbox	Kpt	Flow	Ball	Acc	Acc
ARG [38]	✔	✔				90.7	91.0
HiGCIN [41]	✔	✔				91.4	92.0
DIN [45]	✔	✔				92.7⋆	92.8⋆
POGARS [37]			✔		✔	93.9	-
Composer [51]			✔		✔	93.6⋆	-
SkeleTR [11]			✔			94.4	-
MP-GCN [26]			✔		✔	95.0†	95.0†
Bi-Causal [49]			✔		✔	95.8⋆	-
CRM [3]	✔	✔		✔		93.0	-
ActorFormer [13]		✔	✔	✔		94.4	-
GIRN [31]	✔	✔	✔	✔		94.0	-
SACRF [32]	✔	✔	✔	✔		95.0	-
GroupFormer [24]	✔	✔	✔	✔		95.7	-
Dual-AI [16]	✔	✔		✔		95.4	-
KRGFormer [30]	✔	✔	✔			94.6	94.8
PromptGAR	✔	✔	✔			**96.0**	**96.3**

Table 1. **Quantitative Comparisons in Volleyball Dataset**. (−) denotes not reported in the paper, (∗) reproduced from released codes, (†) reproduced with keypoint-ball results (late-fusion parts un-released), and unmarked other methods are not open-sourced.

Method	Prompt Types					Top1	Mean
	RGB	Bbox	Kpt	Flow	Ball	Acc	Acc
ARG [38]	✔	✔				59.0	56.8
ActorFormer [13]	✔	✔		✔		47.1	41.5
SAM [42]	✔	✔				49.1	47.5
DIN [45]	✔	✔				61.6	56.0
Dual-AI [16]	✔	✔		✔		58.1	50.2
KRGFormer [30]	✔	✔				72.4	67.1
MP-GCN [26]			✔		✔	75.8†	72.0†
DFWSGAR [20]	✔					75.8	71.2
Flaming-Net [29]	✔			✔		79.1	76.0
PromptGAR	✔	✔	✔			**80.6**	**76.9**

Table 2. **Quantitative Comparisons in NBA Dataset**. (†) reproduced with keypoint-ball results (late-fusion and multi-ensemble parts un-released), and unmarked methods are not open-sourced.

Consistent with prior works, we limit our analysis for central 16 frames to avoid potential interference from additional group activities present in the remaining half of clips.

NBA Dataset [42] includes 7,624 training clips and 1,548 testing clips. Each clip includes 72 frames and is categorized into nine group activity labels. Due to its increasing complexity, the NBA dataset demands special design and processing compared to the Volleyball dataset [18]. Unlike Volleyball [18], we use all 72 frames in NBA because key event frames are not centrally located. Also, NBA activities are much longer than Volleyball's. For example, '*3p-fail-offensive*' requires recognizing shooter location, ball trajectory, and possession.

Implementation details. We utilize the MViTv2-Base [25] video encoder with a 224×224 input resolution. For the Volleyball dataset [18], training and testing use the center 16 frames. The NBA dataset [42] is trained on 56 uniformly sampled frames and tested on all 72. The recognition decoder consists of an eight-layer stack of two-way transformers. All models are trained on 4 A100-80GB GPUs, where the Volleyball dataset use a batch size of 64, and the NBA dataset employ 24. Training runs for 200 epochs, using an AdamW optimizer and a Cosine Annealing learning rate scheduler. The initial learning rates are 2×10^{-4} for Volleyball and 7.5×10^{-5} for NBA.

4.2. Group Activity Recognition

Volleyball Dataset. In Tab. 1, we compare our method to the group activity recognition methods that rely on: (a) RGB frames, like DIN [45], which suffers from limited performance; (b) skeletons and ball trajectories, like Composer [51] and MP-GCN [26], which requires ball trajectories as extra inputs; (c) combined visual prompts, like GroupFormer [24] and Dual-AI [16], which takes computationally expensive optical flows as inputs and thus sensitive to input frame rates. In contrast, our PromptGAR, without the drawbacks of those other visual prompt inputs, still produces 96.0% top1 accuracy and 96.3% mean accuracy, achieving considerable improvements across different baseline models. Also, ACCG [39] can not be evaluated due to unreleased codes.

NBA Dataset. Tab. 2 shows that PromptGAR gets competitive performance over previous methods on the challenging NBA dataset [42]. Prior approaches often face limitations. Older models like SAM [42] and Dual-AI [16] accept only raw videos and bounding boxes, which restricts their performance; Other methods, such as MP-GCN [26], require specialized inputs like ball trajectories, thereby limiting their applicability to broader, non-sports GAR tasks. Additionally, weakly-supervised techniques like DFWSGAR [20] and Flaming-Net [29] introduce considerable training complexity. In contrast, our PromptGAR, without these drawbacks, still produces 80.6% top1 accuracy and 76.9% mean accuracy, reflecting substantial advancements over various baseline models.

4.3. Input Flexibility

Flexible Prompts. Tab. 3 demonstrates PromptGAR's resilience to reduced prompt information. Using only skeleton data, it loses a negligible 0.3% in accuracy. When relying solely on bounding boxes, it achieves the same accuracy as GroupFormer [24], but without retraining. Even with only RGB input, the model's accuracy remains reasonable, only 2.1% lower than with full prompts. This shows PromptGAR's robustness to maintain high performance even with significantly reduced prompt inputs without retraining.

Method	Prompt Types				Re-train	Top1	Mean
	RGB	Bbox	Kpt	Flow		Acc	Acc
GroupFormer [24]	✓	✓	✓	✓	✓	95.7	-
	✓	✓		✓		94.9	-
	✓	✓				94.1	-
PromptGAR	✓	✓	✓		×	**96.0**	**96.3**
	✓		✓			95.7	95.8
	✓	✓				94.1	94.3
	✓					93.9	94.4

Table 3. **Flexible Prompts**. In the Volleyball dataset, we achieve remarkable performance under diverse prompts without retraining.

Method	Train		Test		Re-train	Top1	Mean
	T	Sampling	T	Sampling		Acc	Acc
MP-GCN [26]	18	stride 1	18	stride 1	✓	73.3	68.4
	72	stride 1	72	stride 1		75.8	72.0
PromptGAR	56	uniform	36	stride 2	×	75.4	71.9
			56	stride 1		75.0	71.0
			64	stride 1		78.4	74.1
			72	stride 1		**80.6**	**76.9**

Table 4. **Flexible Frames**. For the NBA dataset, frames are all sampled at the center of each video clip. PromptGAR's validation process also uses $T = 72$ to select the optimal checkpoint.

Method	# Instances	Re-train	Top1 Acc	Mean Acc
PromptGAR	12	×	**96.0**	**96.3**
	10		95.4	95.6
	5		94.5	94.8
	3		94.3	94.6

Table 5. **Flexible Instances**. In the Volleyball dataset, instances are randomly deleted, and results are averaged over three trials.

Flexible Frames. Tab. 4 exhibits PromptGAR's adaptability to various frame sampling configurations. Although trained on 56 frames with the NBA dataset, it can effectively handle both shorter and longer sequences at test time. Specifically, it achieves 75.4% accuracy when tested with 36 frames (stride 2), nearly matching MP-GCN's performance, which needs to be trained and tested on the full 72 frames specifically. Moreoever, PromptGAR also outperforms MP-GCN when tested on all 72 frames, even though MP-GCN is optimized for this exact frame count. Furthermore, the effective rollout length during testing is crucial. When testing with 56 frames and a stride of 1, we see a 0.4% accuracy decrease compared to 36 frames with a stride of 2, due to the shorter rollout length (56 vs. 72). Those results demonstrate PromptGAR's robust ability to handle diverse frame inputs without the need for retraining, a clear advantage over methods requiring retraining.

Flexible Instances. Tab. 5 showcases that PromptGAR can maintain high performance even when tested with fewer instances than it was trained on, without requiring retraining. To ensure a fair comparison, instances were randomly selected, and results were averaged over three experiments. Reducing from 12 instances to 10 results in only a 0.6% accuracy drop, and even when reduced to just 3 instances, the drop is still manageable at 1.7%. This highlights PromptGAR's ability to function with reduced input, without the needs for retraining.

4.4. Actor Consistency

Actor Order Invariance. Prompt inputs for the testing dataset are formatted as a tensor of shape (N, T, M, J, C), representing the number of videos, frames, actors, point types, and coordinates, respectively. Bounding boxes and skeletal keypoints belonging to the same instance *ID* share the same index along the M (actor) axis. The actor order along this M axis is usually fixed in conventional evaluations, but we introduce a rigorous test: we randomly shuffle the M axis and evaluate the same model. This procedure is repeated five times, and the averaged performance is presented in Tab. 6. Our results demonstrate that after shuffling, MP-GCN [26] shows a slight performance degradation, but our model maintains identical performance numbers. This is because current GAR methods [13, 16, 24, 26, 51] fix that order for both validation and testing, choosing the best checkpoint based on performance with that specific player order. In contrast, our model incorporates a novel relative instance attention mechanism that encodes only whether two actors are the same or not. This design ensures our model remains unaffected by actor order in real-world scenarios, thereby enhancing its robustness.

Necessity of Instance Identities. As shown in Tab. 7, there is a performance drop without instance identities. This is because, without instance identities, the model struggles to maintain actor consistency and reliably associate indi-

Method	Shuffle Input Actor Order	Re-train	Top1 Acc	Mean Acc
MP-GCN [26]	×	×	75.8	72.0
	✓		74.7	71.2
PromptGAR	×	×	80.6	76.9
	✓		**80.6**	**76.9**

Table 6. **Actor Order Invariance**. For NBA dataset, input actor order is randomly shuffled, and results are averaged over five trials.

Method	Relative Instance Identity Embeddings	Re-train	Top1 Acc	Mean Acc
PromptGAR	✓	×	**96.0**	**96.3**
	×		93.8	94.1

Table 7. **Necessity of Instance Identities**. In Volleyball dataset, relative instance identity embeddings are included or omitted, using prompts from bounding boxes, skeletons, and RGB videos.

Method	Re-train	Top1 Acc	Mean Acc
PromptGAR	✓	**95.8**	**96.0**
wo inst IDs	✓	95.4	95.6
wo prompt tokens in head	✓	95.1	95.4
no prompts	✓	94.3	94.7

Table 8. **Effectiveness of Novel Modules**. Experiments are conducted using smaller models with $O = 16$ in Volleyball. Ablation studies included: '*wo inst IDs*', where relative instance attention is replaced with regular self-attention; '*wo prompt tokens in head*', where only the GAR class token $\mathbf{X}_{gar}$ is used for the GAR head; and '*no prompts*', where the recognition decoder receives only the GAR class token $\mathbf{X}_{gar}$ and RGB features $\mathbf{F}_I$ as inputs.

O	Memory	Top1 Acc	Mean Acc
16	40 *GB*	95.8	96.0
48	72 *GB*	**96.0**	**96.3**
56	*OOM*	-	-

Table 9. **# of Pooled Prompts** O.

ϵ	Top1 Acc	Mean Acc
0	95.4	95.6
5	95.6	95.9
10	**95.8**	**96.0**

Table 10. **Relative Instance Scale** ϵ.

vidual features across different frames. This also proves the necessity of our instance identities.

4.5. Ablation Studies

Tab. 8 details the impact of our novel modules on performance, starting with a baseline of 96.0% top-1 accuracy on the Volleyball dataset. (a) **Relative instance attention**. Eliminating instance *ID*s during training and testing resulted in a 0.6% drop in top-1 accuracy. This confirms that instance *ID*s are vital for encoding long-term consistency, enhancing performance. Furthermore, relative instance attention also accelerated training convergence. (b) **Prompt tokens in head**. Removing prompt tokens $\widehat{\mathbf{F}}_p$ from the head, leaving only the GAR class token $\widehat{\mathbf{X}}_{gar}$, led to a 0.3% accuracy decrease. This highlights the necessity of explicitly incorporating prompt tokens. Without them, the model struggles to integrate diverse visual prompt features, as the GAR class token alone sufficiently captures RGB information. (c) **No prompts**. Disabling the entire prompt encoder, using only the GAR class token $\widehat{\mathbf{X}}_{gar}$ as input for the recognition decoder and the head, further reduced accuracy by 0.8%. This underscores the contribution of visual prompt inputs in PromptGAR. Crucially, even without any prompts during training and testing, our model still compares favorably with other prompt-based baseline methods. Specifically, it achieves a 1.6% higher accuracy than DIN, which utilizes both RGB frames and bounding boxes, and a 0.7% higher accuracy than Composer, which uses skeletons and ball trajectories as additional information. This demonstrates the superior ability of our MViTv2 architecture to process spatial-temporal information within only the RGB features compared to other backbones that process the video frame-by-frame separately.

Number of Pooled Prompts. Tab. 9 explores the impact of the number of pooled prompt channels (O) on performance. The prompt encoder's output has the shape $D \times N \times O$, where D, N, O are the feature dimension, the number of instances, and the pooled prompt channels after pooling along temporal and prompt type dimensions, repectively. A smaller O value reduces the representativeness and expressiveness of each actor's key information. As shown in the table, increasing O from 16 to 48 improves top1 accuracy by 0.2%. However, this increase quadratically raises the computational cost and CUDA memory usage of the subsequent recognition decoder module. Specifically, the CUDA memory usage for $O = 48$ is 32 GB higher than for $O = 16$. Further increasing O to 56 would exceed the 80 GB limit of an A100 GPU's CUDA memory, resulting in an out-of-memory error. Therefore, while a higher O value improves performance, a balance must be struck to avoid excessive computational demands and memory limitations.

Relative Instance Scale. Tab. 10 details the process of determining the optimal relative instance scale, denoted as ϵ. This value serves as a coefficient for the relative instance weights, which are added to the standard attention weights. Due to the inherent sparsity of the relative instance weight matrix $E^{(rel)}$, a small ϵ value would diminish the impact of instance *ID*s during training. Conversely, an excessively large ϵ would overpower the standard attention weights, hindering the GAR class token's ability to learn effectively from the prompt tokens. The table reveals that setting ϵ to 10 yields a 0.4% improvement in top-1 accuracy compared to setting it to 0 (effectively disabling relative instance attention) and a 0.2% improvement compared to setting it to 5. Therefore, we selected 10 as the final ϵ value. Notice that these experiments are conducted with a pooled prompt channel count (O) of 16.

5. Limitations

Despite its competitive GAR accuracy and input flexibility, PromptGAR faces limitations with input resolution. The input resolution is restricted to 224 × 224, which is significantly lower than 1080 × 720 in other methods [16, 24, 45].

6. Conclusion

In this work, we present PromptGAR, a novel framework that addresses the limitations of current Group Activity Recognition (GAR) approaches by leveraging diverse visual prompts to achieve both input flexibility and high recognition accuracy. Extensive experiments show that PromptGAR allows the input flexibility across prompts, frames, and instances without the need for retraining. Furthermore, we introduce a novel relative instance attention mechanism that directly encodes instance identities, ensuring actor consistency. Extensive evaluation validates the effectiveness and applicability of our model.

References

[1] Mohamed R Amer, Dan Xie, Mingtian Zhao, Sinisa Todorovic, and Song-Chun Zhu. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12*, pages 187–200. Springer, 2012. 3

[2] Mohamed R Amer, Sinisa Todorovic, Alan Fern, and Song-Chun Zhu. Monte carlo tree search for scheduling activity recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1353–1360, 2013. 3

[3] Sina Mokhtarzadeh Azar, Mina Ghadimi Atigh, Ahmad Nickabadi, and Alexandre Alahi. Convolutional relational machine for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2019. 6

[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. 4

[5] Zhongwei Cheng, Lei Qin, Qingming Huang, Shuqiang Jiang, and Qi Tian. Group activity recognition by gaussian processes estimation. In *2010 20th International Conference on Pattern Recognition*, pages 3228–3231. IEEE, 2010. 3

[6] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12*, pages 215–230. Springer, 2012.

[7] Wongun Choi, Khuram Shahid, and Silvio Savarese. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops*, pages 1282–1289. IEEE, 2009. 1, 3

[8] Wongun Choi, Khuram Shahid, and Silvio Savarese. Learning context for collective activity recognition. In *CVPR 2011*, pages 3273–3280. IEEE, 2011. 3

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 1

[10] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5

[11] Haodong Duan, Mingze Xu, Bing Shuai, Davide Modolo, Zhuowen Tu, Joseph Tighe, and Alessandro Bergamo. Skeletr: Towards skeleton-based action recognition in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13634–13644, 2023. 3, 6

[12] Ruopeng Gao, Yijun Zhang, and Limin Wang. Multiple object tracking as id prediction. *arXiv preprint arXiv:2403.16848*, 2024. 1

[13] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees GM Snoek. Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 839–848, 2020. 1, 2, 3, 6, 7

[14] Zigang Geng, Chunyu Wang, Yixuan Wei, Ze Liu, Houqiang Li, and Han Hu. Human pose as compositional tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 660–671, 2023. 2

[15] Hossein Hajimirsadeghi, Wang Yan, Arash Vahdat, and Greg Mori. Visual recognition by counting instances: A multi-instance cardinality potential kernel. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2596–2605, 2015. 3

[16] 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. 1, 2, 3, 4, 5, 6, 7, 8

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[18] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1980, 2016. 2, 3, 5, 6, 1

[19] Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose. *arXiv preprint arXiv:2303.07399*, 2023. 1

[20] Dongkeun Kim, Jinsung Lee, Minsu Cho, and Suha Kwak. Detector-free weakly supervised group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20083–20093, 2022. 6

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 3, 4, 5

[22] Tian Lan, Yang Wang, Weilong Yang, Stephen N Robinovitch, and Greg Mori. Discriminative latent models for recognizing contextual group activities. *IEEE transactions on pattern analysis and machine intelligence*, 34(8):1549–1562, 2011. 3

[23] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989. 5

[24] Shuaicheng Li, Qianggang Cao, Lingbo Liu, Kunlin Yang, Shinan Liu, Jun Hou, and Shuai Yi. Groupformer: Group

activity recognition with clustered spatial-temporal transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13668–13677, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[25] 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. 4, 5, 6, 1

[26] Zhengcen Li, Xinle Chang, Yueran Li, and Jingyong Su. Skeleton-based group activity recognition via spatial-temporal panoramic graph. In *European Conference on Computer Vision*, pages 252–269. Springer, 2025. 1, 2, 3, 5, 6, 7

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 4

[28] Moin Nabi, Alessio Bue, and Vittorio Murino. Temporal poselets for collective activity detection and recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 500–507, 2013. 3

[29] Muhammad Adi Nugroho, Sangmin Woo, Sumin Lee, Jinyoung Park, Yooseung Wang, Donguk Kim, and Changick Kim. Flow-assisted motion learning network for weakly-supervised group activity recognition. *arXiv preprint arXiv:2405.18012*, 2024. 6

[30] Duoxuan Pei, Di Huang, Longteng Kong, and Yunhong Wang. Key role guided transformer for group activity recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(12):7803–7818, 2023. 2, 6

[31] Mauricio Perez, Jun Liu, and Alex C Kot. Skeleton-based relational reasoning for group activity analysis. *Pattern Recognition*, 122:108360, 2022. 6

[32] Rizard Renanda Adhi Pramono, Yie Tarng Chen, and Wen Hsien Fang. Empowering relational network by self-attention augmented conditional random fields for group activity recognition. In *European Conference on Computer Vision*, pages 71–90. Springer, 2020. 2, 3, 4, 5, 6

[33] Tianmin Shu, Dan Xie, Brandon Rothrock, Sinisa Todorovic, and Song Chun Zhu. Joint inference of groups, events and human roles in aerial videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4576–4584, 2015. 3

[34] Vukašin Stanojević and Branimir Todorović. Boosttrack++: using tracklet information to detect more objects in multiple object tracking. *arXiv preprint arXiv:2408.13003*, 2024. 2

[35] Weijie Su, Xizhou Zhu, Chenxin Tao, Lewei Lu, Bin Li, Gao Huang, Yu Qiao, Xiaogang Wang, Jie Zhou, and Jifeng Dai. Towards all-in-one pre-training via maximizing multi-modal mutual information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15888–15899, 2023. 2

[36] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020. 4

[37] Haritha Thilakarathne, Aiden Nibali, Zhen He, and Stuart Morgan. Pose is all you need: The pose only group activity recognition system (pogars). *Machine Vision and Applications*, 33(6):95, 2022. 3, 6

[38] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu. Learning actor relation graphs for group activity recognition. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 9964–9974, 2019. 6

[39] Zhao Xie, Tian Gao, Kewei Wu, and Jiao Chang. An actor-centric causality graph for asynchronous temporal inference in group activity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6652–6661, 2023. 6

[40] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in neural information processing systems*, 35:38571–38584, 2022. 2

[41] Rui Yan, Lingxi Xie, Jinhui Tang, Xiangbo Shu, and Qi Tian. Higcin: Hierarchical graph-based cross inference network for group activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, 45(6):6955–6968, 2020. 3, 6

[42] Rui Yan, Lingxi Xie, Jinhui Tang, Xiangbo Shu, and Qi Tian. Social adaptive module for weakly-supervised group activity recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 208–224. Springer, 2020. 2, 3, 6, 1

[43] Jie Yang, Ailing Zeng, Ruimao Zhang, and Lei Zhang. X-pose: Detecting any keypoints. In *European Conference on Computer Vision*, pages 249–268. Springer, 2025. 5

[44] Kefu Yi, Kai Luo, Xiaolei Luo, Jiangui Huang, Hao Wu, Rongdong Hu, and Wei Hao. Ucmctrack: Multi-object tracking with uniform camera motion compensation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 6702–6710, 2024. 2

[45] Hangjie Yuan, Dong Ni, and Mang Wang. Spatio-temporal dynamic inference network for group activity recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7476–7485, 2021. 2, 3, 4, 6, 8

[46] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 1

[47] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition: 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007. Proceedings 29*, pages 214–223. Springer, 2007. 3

[48] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1

[49] Youliang Zhang, Wenxuan Liu, Danni Xu, Zhuo Zhou, and Zheng Wang. Bi-causal: Group activity recognition via bidirectional causality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1450–1459, 2024. 2, 6

[50] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13001–13008, 2020. 1

[51] Honglu Zhou, Asim Kadav, Aviv Shamsian, Shijie Geng, Farley Lai, Long Zhao, Ting Liu, Mubbasir Kapadia, and Hans Peter Graf. Composer: compositional reasoning of group activity in videos with keypoint-only modality. In *European Conference on Computer Vision*, pages 249–266. Springer, 2022. 1, 2, 3, 5, 6, 7

[52] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023. 2

PromptGAR: Flexible Promptive Group Activity Recognition

Supplementary Material

Method	Prompt Types				**Head Inputs**		Re-train	Top1	Mean
	RGB	Bbox	Kpt	InstID	$\widehat{\boldsymbol{X}}_{gar}$	$\widehat{\boldsymbol{F}}_p$			
PromptGAR	✓	✓	✓	✓	✓	✓	×	**96.0**	**96.3**
	✓		✓	✓	✓	✓		95.7	95.8
	✓	✓		✓	✓	✓		94.1	94.3
	✓	✓	✓	✓	✓			95.8	96.0
	✓		✓	✓	✓			95.5	95.7
	✓	✓		✓	✓			94.0	94.2
	✓				✓			93.9	94.4
	✓	✓	✓	✓		✓		95.2	95.6
	✓		✓	✓		✓		95.2	95.6
	✓	✓		✓		✓		93.4	94.0

Table 11. **Ablation on Head Input Tokens**.

Method	Train				Test				Top1	Mean
	RGB	Bbox	Kpt	InstID	RGB	Bbox	Kpt	InstID	Acc	Acc
PromptGAR	✓	✓	✓	✓	✓	✓	✓	✓	**96.0**	**96.3**
					✓		✓	✓	95.7	95.8
	✓	✓		✓	✓	✓		✓	95.4	95.8
	✓				✓				94.3	94.7

Table 12. **Low Reliance on Skeletons**.

1. More on Implementation Details

Data Augmentations. Following MViTv2 in action recognition [25], we apply a comprehensive set of augmentations for group activity recognition (GAR): Random Augment [9], Random Resized Crop, Flip, Random Erasing [50], CutMix [46], and MixUp [48].

Volleyball Dataset [18]. Since Volleyball only contains annotations with central 16 frames, we replace the depth-wise prompt pooling with a single MLP layer projecting channels from $T \times 20$ to the number of pooled prompts O.

NBA Dataset [42]. We regenerated annotations due to inaccuracies in SAM [42] and MP-GCN [26], such as background audience false positives, player ID switching, and ID reassignment upon reappearance. We used MOTIP [12] for robust player tracking, optimized for sports data, and RTMPose [19] for accurate skeleton generation, handling occlusions effectively.

2. Additional Ablation Studies

Head Inputs. Tab. 11 indicates demonstrates the significance of prompt token inputs within the head, even when dealing with diverse visual prompts. We compared performance using a model trained on full prompts, varying the head inputs during testing. Under full prompt inputs, using only the GAR class token $\widehat{\mathbf{X}}_{gar}$ as the head input resulted in a minor 0.2% performance drop, while using only the prompt tokens $\widehat{\mathbf{F}}_p$ led to a 0.8% drop. This shows that both inputs are crucial for PromptGAR's high performance. Even with reduced prompt inputs, both $\widehat{\mathbf{X}}_{gar}$ only and $\widehat{\mathbf{F}}_p$ only still yielded reasonable results. Notably, when using only prompt tokens $\widehat{\mathbf{F}}_p$, the performance of full prompt inputs was similar to the one with skeleton-only inputs. This suggests that when RGB features don't directly influence the final prediction, bounding box information is effectively captured within the skeleton data. In summary, this table showcases the flexibility of head inputs across various visual prompts and underscores the importance of both the RGB feature representation $\widehat{\mathbf{X}}_{gar}$ and the prompt feature representation $\widehat{\mathbf{F}}_p$ for optimal performance.

Low Reliance on Skeletons. Tab. 12, our model has competitive performance even when trained without skeletal data. This outcome indicates that our model does not heavily rely on skeletal information.

3. Analysis

Quantitative Results. Fig. 7 - 8 shows the confusion matrix of Volleyball and NBA, respectively. (a) For Volleyball, while PromptGAR demonstrates high accuracy on the Volleyball dataset, a common misclassification occurs where '*right-set*' is mistaken for '*right-pass*'. This error is attributed to the inherent similarity in player actions and positions, echoing findings in previous research [51]. (b) For NBA, errors highlight the challenge of distinguishing between highly similar actions. Specifically, differentiating '*offensive*' and '*defensive*' requires nuanced visual analysis of player uniform differences and rebound outcomes. Similarly, distinguishing '*2p-fail*' from '*2p-layup-fail*' demands attention to shooting position and posture near the rim.

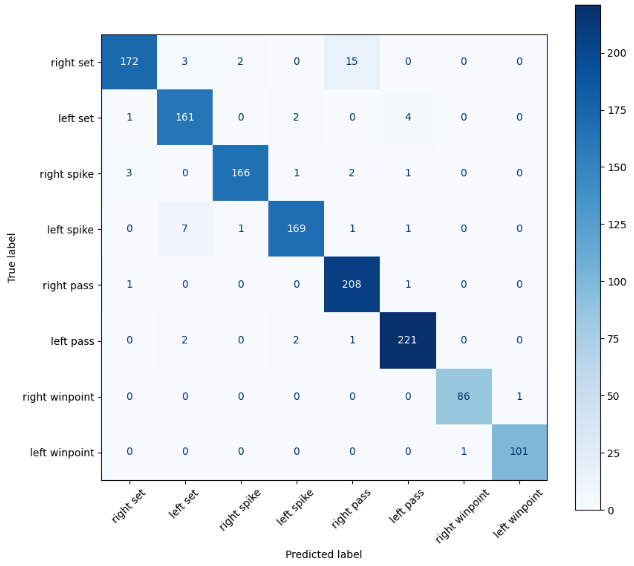

Figure 7. **Volleyball Confusion Matrix**

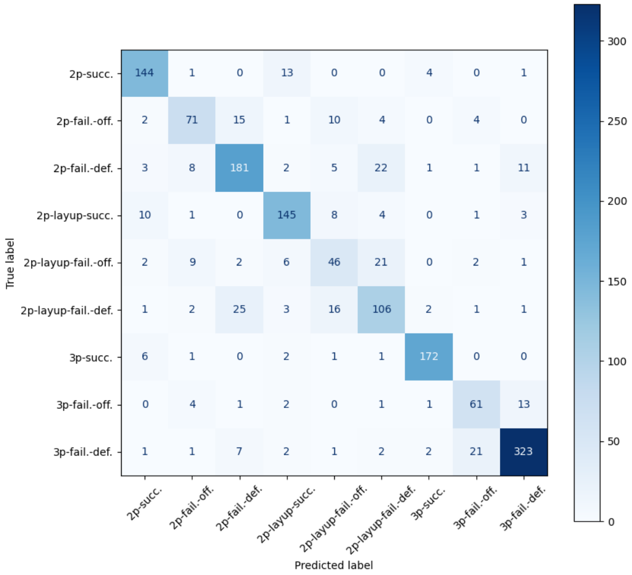

Figure 8. **NBA Confusion Matrix**