HIG: Hierarchical Interlacement Graph Approach to Scene Graph Generation in Video Understanding
Supplementary

Trong-Thuan Nguyen, Pha Nguyen, Khoa Luu
CVIU Lab, University of Arkansas
{thuann, panguyen, khoaluu}@uark.edu
https://uark-cviu.github.io/ASPIRe/

1. ASPIRe Dataset Annotation Pipeline

We introduce a specialized system for data labeling that uniquely combines the power of visual and linguistic analysis to generate precise and contextually rich labels for image data. By employing advanced techniques like RoIAlign for region-specific data extraction and integrating these with language embeddings, GPT4RoI [5] transcends the conventional labeling approach. This process, augmented by post-processing with Spacy [1] and meticulous human curation, ensures accurate data labels.

GPT4RoI. Our initial step involves using GPT4RoI to generate textual descriptions corresponding to input bounding boxes. GPT4RoI integrates visual and linguistic data and adeptly handles spatial instructions. During processing, GPT4RoI replaces $<region_i>$ tags in these instructions with results from RoIAlign, derived directly from the image"s features. This process creates a unique fusion of region-specific data with language embeddings. For enhanced multimodal understanding, this combination of embeddings is then interpreted by the Vicuna [6] model, a specialized instance of the LLaMA [2]. This allows us to input bounding boxes around objects and prompt the system for detailed descriptions, covering aspects like appearance, situation, positioning, interactions, and relationships. For instance, when we input bounding boxes around objects and ask the system questions, such as determining the relationship between individuals in $<region_1>$ and $<region_2>$, the system responds with detailed, context-rich descriptions.

Post-Processing with Spacy. After generating text with GPT-4RoI, we utilize Spacy, a Python library for natural language processing, to refine the text further. We specifically use Spacy to add grammatical tags to each word in the text. This tagging involves identifying the grammatical role of each word and determining if it is a noun, verb, or adjective, among others. This process is essential for understanding the sentence structure and ensuring that the text is accurate in its content and grammatically coherent.

Human Curation and Filtering. For the final step, we rely on human expertise to ensure the highest quality of our output. Our team carefully reviews the Spacy-processed text using a specially designed filter that helps categorize interactivity types. This human oversight is essential for maintaining the highest standards of accuracy and relevance. It enables us to meticulously confirm and refine the interaction types identified by the LLM, ensuring that our final label is precise.

2. Data format

Our annotations are organized following the below main structure:

```
data[{
    "file_name": str,
    "height": int,
    "width": int,
    "image_id": int,
    "frame_index": int,
    "video_id": int,
```

[1] https://spacy.io/

```
    "segments_info":[{
        "id": int,
        "track_id": int,
        "category_id": int,
        "iscrowd": 0 or 1,
        "isthing": 0 or 1,
        "area": int
    }],
    "annotations":[{
        "bbox": [x, y, width, height],
        "segmentation": [polygon],
        "bbox_mode": 0 or 1,
        "category_id": int
    }]
    "appearances" : [{
        "segment_id": int,
        "app_id": int
    }],
    "situations" : [{
        "segment_id": int,
        "sit_id": int
    }],
    "positions" : [{
        "segment_id": int,
        "segment_id": int,
        "pos_id": int
    }],
    "interactions" : [{
        "segment_id": int,
        "segment_id": int,
        "inter_id": int
    }],
    "relations" : [[
        "segment_id": int,
        "segment_id": int,
        "rel_id": int
    ]]
}],
"thing_classes": [int],
"stuff_situations": [int],
"predicate_apparances": [int],
"predicate_situations": [int],
"predicate_positions": [int],
"predicate_interactions": [int],
"predicate_relations": [int],
```

2.1. Basic Image Information

This section details the fundamental attributes of each image:

- `file_name`: The name of the image file.
- `height`: The height of the image in pixels.
- `width`: The width of the image in pixels.
- `image_id`: A unique identifier for the image.

- `frame_index`: The index of the frame within the video sequence.
- `video_id`: An identifier for the video or image collection to which this image belongs.

2.2. Segment Information

This section includes the `segments_info` key, which is a list of segments within the image. Each segment contains:
- `id`: Unique identifier for the segment.
- `track_id`: Identifier to track the segment across different frames.
- `category_id`: Identifier for the category of the object in the segment.
- `iscrowd`: A binary value indicating if the segment represents a crowd.
- `isthing`: A binary value indicating if the segment represents a "thing" (as opposed to "stuff" like banner, blanket, curtain, pillow, towel).
- `area`: The area covered by the segment in the image.

In addition, for each entry in `segments_info`, we provide the corresponding masks (`segmentation`) and bounding boxes (`bbox`), each tagged with a specific `category_id` in the `annotations`.

2.3. Interactivity Attributes

This section encompasses lists of `predicate_appearances`, `predicate_situations`, `predicate_positions`, `predicate_interactions`, and `predicate_relations` for each segment. For single-actor attributes (i.e., appearances and situations), the structure is as follows:
- `segment_id`: Identifier for the segment.
- `id`: Identifier for the interactivity type.

For double-actor attributes (i.e., positions, interactions, and relations), the structure includes two different `segment_ids` to represent the interactivity between two segments:
- `segment_id_1`: Identifier for the first segment.
- `segment_id_2`: Identifier for the second segment.
- `id`: Identifier for the interactivity type.

These descriptors represent lists of integers, specifying various aspects of the subject, object, and interactivity for each bounding box within the annotations and `segments_info`. For example, *[1, 9, 8]* in a dual-actor scenario indicates that the second segment in `segments_info` is the subject, the ninth segment is the object, and they share a predicate class 8, signifying a position/interaction/relation. Conversely, *[1, 8]* or *[2, 0]* in a single-actor scenario indicates that the second or third segment in `segments_info` is associated with a class 8 or 0 predicate of appearance/situation.

3. Approximation

Table 1. Summary of annotated attributes between two actors in our ASPIRe dataset (with ✓represented as 1 and ✗as 0). *appearance* and contextual *situation* are single-actor attributes

	Position	**Interaction**	**Relation**
Person-Person	✓	✗	✓
Person-Object	✗	✓	✗
Object-Person	✓	✗	✓
Object-Object	✗	✗	✗

We investigate our problem among individual actors and estimate the possible pairs between two actors within these interactivities. When examining a single attribute, two pivotal metrics arise the subject"s *appearance* ($\mathcal{A}$) and *situation* ($\mathcal{S}$). When we identify a set S_t at a particular time t to encompass all subjects, these individual interactivities correspond to the number of subjects, denoted as $|S_t|$. When shifting to the bipartite matching of the dual-actor, three central pillars come into focus: *position* ($\mathcal{PO}$), *interaction* ($\mathcal{IN}$), and *relationship* ($\mathcal{RE}$). To provide further detail, we classify these interactivities into four distinct pairs, as shown in Table 1. Each can be depicted as a pairwise matrix product, effectively capturing the presence or absence of our central attributes.

By leveraging the unique eigenvectors of attributes that span various interactivities, our focus shifts to a set S_t comprising n subjects. Specifically, the pairs are determined by combinations of bipartite subjects, which we denote as C_2^n. We define the possible configurations combined with this combinatorial expression for each pair with the feasible attribute vector specific to that particular interactivity. We symbol r_{PP}, r_{PO}, r_{OP}, and r_{OO} that is the feasible attribute eigenvalues equivalented to each row in the binary coefficients matrix, resulting in the unified equation that defines the number of pairs across the three interactivities $\#_{pairs} = C_2^n \times (r_{PO} + r_{OP} + r_{OO})$. In our analysis of each actor, we carefully assess both the physical attributes of the subjects and their contextual situations. Consequently, the number of single attributes considered equals the number of subjects, denoted as n. Conversely, interactivities related to position, interaction, and relationships involve dual actors. To elaborate further, we categorize these interactions into four distinct pairs, as presented in Table 1. In each pairing of dual actors, we define their roles to illustrate how various attributes manifest. These subjects are paired together in combinations, denoted as C_2^n. With each pair, we investigate how their interactivities influence each other, employing specific attribute values customized for those specific interactivities.

Importantly, each type of interactivity exhibits unique characteristics that enable us to form pairs. Positions and relations are relevant when both the subject and object are persons or when the object assumes the role of a subject and appears with a person. On the other hand, Interaction exclusively takes place when a person serves as the subject engaging with an object. As a result, to determine the total number of pairs within these interactions, we utilize the following formula: $\#_{pairs} = C_2^n \times (r_{PO} + r_{OP} + r_{OO})$, where the variable r, ranging from 1 to 3, corresponds to the attributes of the subjects within each pair. This formula calculates the number of pairs while interactivities can influence these pairs across three distinct types of interactions.

4. Data Sample

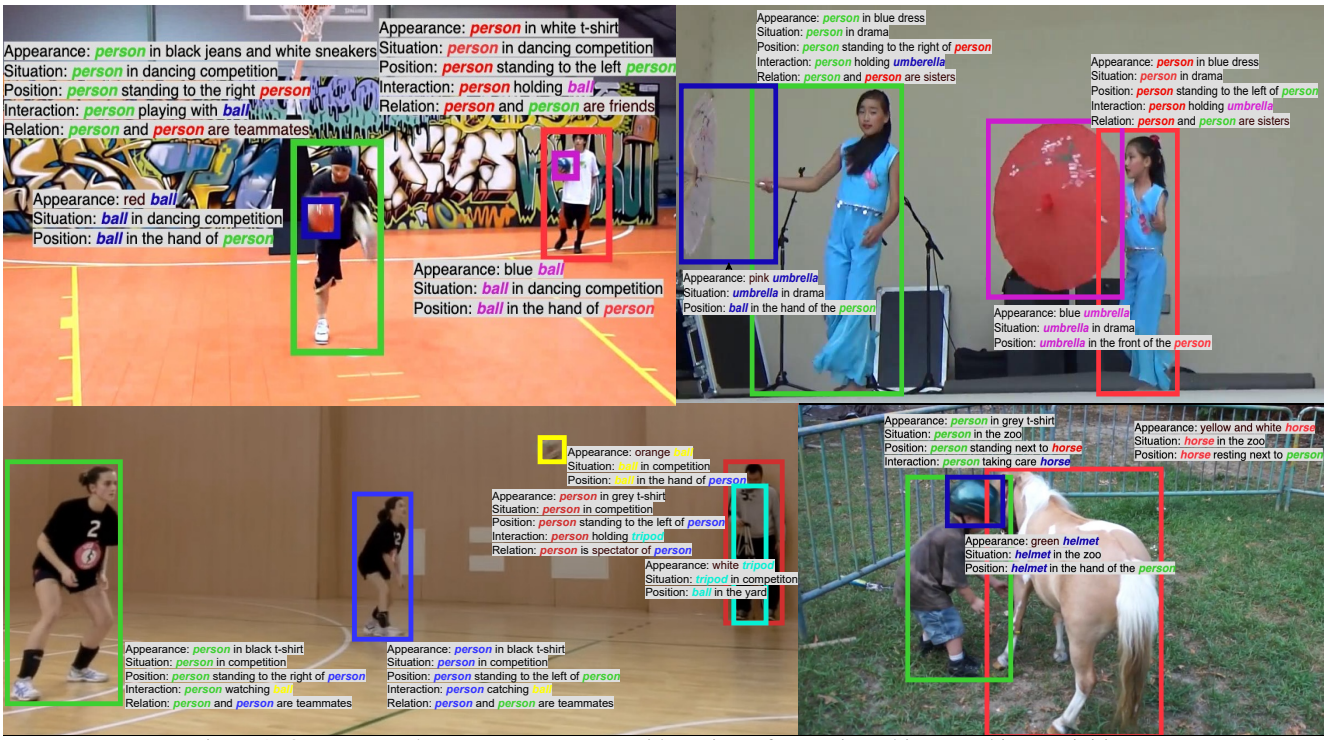

Figure 1. Our *ASPIRe* dataset encompasses a wide variety of scenarios, objects, and interactivities.

Fig. 1 presents selected samples from our *ASPIRe* dataset, notable for its comprehensive range. This dataset includes bounding box annotations and provides detailed descriptions of interactivities across various scenarios. As outlined in Sec. 1, each scene is annotated with precision and contextual relevance, ensuring clarity and circumventing typical ambiguities like generic or overlapping labels found in other datasets. The interactivity within ASPIRe is categorized into five distinct types: appearance, situation, position, interaction, and relation. This multifaceted approach to annotation makes ASPIRe uniquely comprehensive compared to other datasets [1, 3, 4]. Our meticulous annotation process establishes ASPIRe as an invaluable

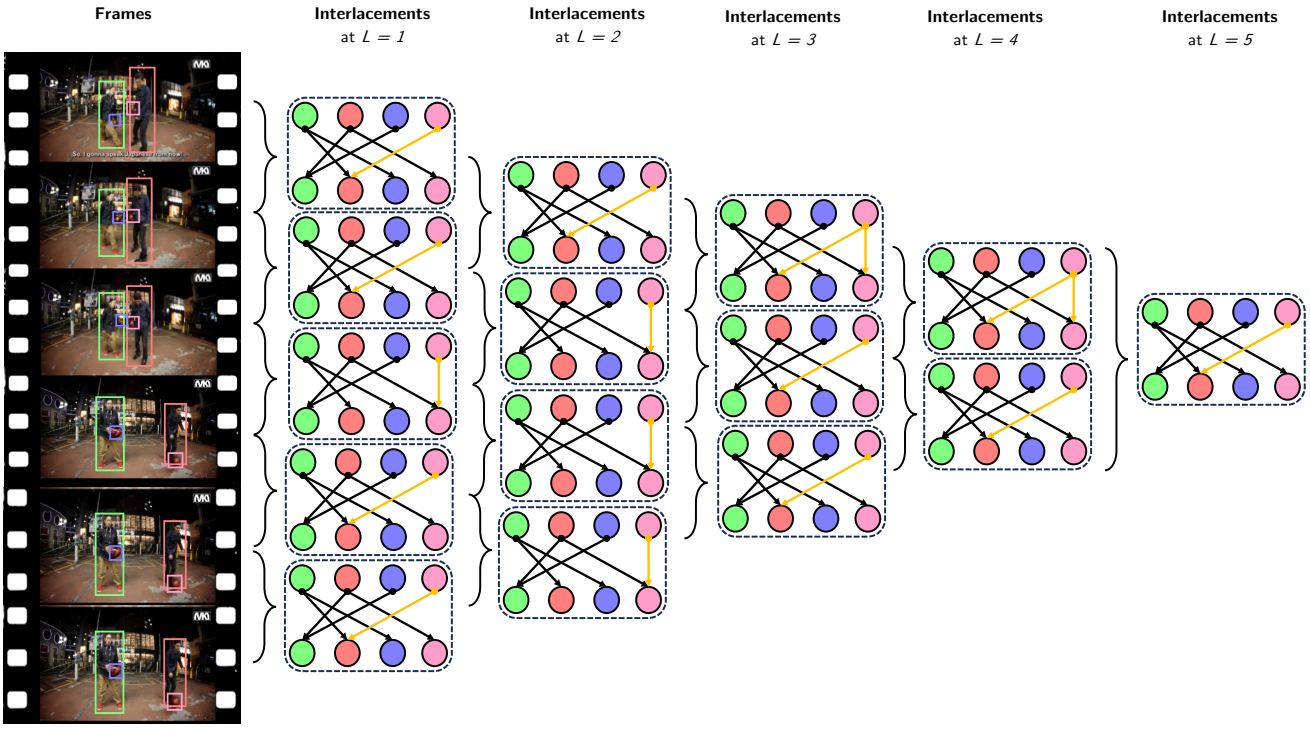

Figure 2. Illustration of Hierarchical Interlacement Graph (HIG).

resource for enhancing the accuracy and efficacy of Visual Interactivity Understanding Algorithms.

5. Methodology

5.1. Baseline Methods

Four basic methods [3] are formulated below to assimilate information from adjacent frames, thereby incorporating temporal information as baseline methods for our problem. At a t^{th} frame, the feature representation of an object i is denoted as q_i^t. Let $Q_i^{(t_1,t_2)}$ be the the set of queries spanning from t_1 to t_2, given by $Q_i^{(t_1,t_2)} = \{q_i^{t_1}, \ldots, q_i^{t_2}\}$. Q also denotes the query tube throughout the entire video if $t_1 = 0$ and $t_2 = |V|$, acting as the feature set for interactivity classification. To this end, we employ pairwise fusion as an initial step to obtain the e_{ij}^t embedding:

$$e_{ij}^t = \text{Concat}(q_i^t, q_j^t) \tag{1}$$

Vanilla Approach involves the fusion of pairwise features:

$$F_{ij}^t = e_{ij}^t \cdot w + b \tag{2}$$

where F_{ij}^t represents the final feature after being transformed via a linear operator.
Handcrafted Filter is a filter g (*i.e.* Gaussian) that convolves with the concatenated feature F_{ij}^t to capture context-specific information. The operation is expressed as:

$$F_{ij}^t = \sum_{k=-W/2}^{W/2} g_k \cdot e_{ij}^{(t+k)} \tag{3}$$

where h_k denotes the values of the handcrafted filter at position k, and W specifies the window size, defining the temporal range of frames considered for contextual analysis.
Convolutional Layer incorporates a trainable 1D-Convolutional layer enhances the feature extraction process. The concatenated embedding e_{ij}^t undergoes convolution with a set of learnable weights w, capturing temporal patterns:

$$F_{ij}^t = \sum_{k=-W/2}^{W/2} w_k \cdot e_{ij}^{(t+k)} \tag{4}$$

Table 2. Comparison at different hierarchical levels of the HIG model.

Hierarchical Level	Interlacement	R/mR@20	R/mR@50	R/mR@100
1	Appearance	7.85 / 0.32	11.47 / 0.38	13.56 / 0.41
	Situation	4.12 / 0.28	5.89 / 0.33	8.43 / 0.37
	Position	8.67 / 0.22	12.34 / 0.27	16.78 / 0.31
	Interaction	5.98 / 0.18	10.76 / 0.23	15.29 / 0.26
	Relation	6.21 / 0.15	10.04 / 0.19	14.67 / 0.24
$n/4$	Appearance	9.43 / 0.39	13.58 / 0.44	15.97 / 0.48
	Situation	4.76 / 0.34	6.22 / 0.39	9.67 / 0.43
	Position	10.89 / 0.29	14.55 / 0.34	19.03 / 0.38
	Interaction	7.34 / 0.24	12.19 / 0.29	17.42 / 0.33
	Relation	7.89 / 0.21	11.76 / 0.26	16.34 / 0.30
$n/2$	Appearance	11.02 / 0.47	15.34 / 0.52	17.89 / 0.56
	Situation	4.83 / 0.40	6.56 / 0.45	11.12 / 0.49
	Position	12.11 / 0.36	16.78 / 0.41	21.45 / 0.45
	Interaction	8.56 / 0.30	14.03 / 0.35	19.67 / 0.39
	Relation	9.02 / 0.27	13.89 / 0.32	18.56 / 0.36
$3n/4$	Appearance	12.76 / 0.53	17.02 / 0.58	19.43 / 0.62
	Situation	4.89 / 0.46	7.01 / 0.51	11.78 / 0.55
	Position	12.45 / **0.42**	18.22 / 0.47	23.67 / 0.51
	Interaction	10.12 / **0.36**	16.47 / **0.41**	22.34 / **0.45**
	Relation	10.16 / **0.33**	15.43 / **0.38**	20.89 / **0.42**
n **(full)**	Appearance	**15.02 / 0.60**	**18.60 / 0.64**	**20.11 / 0.65**
	Situation	**5.01 / 0.56**	**7.02 / 0.55**	**12.01 / 0.63**
	Position	**13.02** / 0.09	**24.52 / 1.33**	**42.33 / 1.12**
	Interaction	**12.02** / 0.11	**24.65** / 0.12	**41.65** / 0.14
	Relation	**10.26** / 0.29	**23.72** / 0.34	**41.47** / 0.39

Here, w represents the weights of the convolutional layer.

Transformers leverage the Transformer architecture, which is to model complex interactivities. Queries are subjected to the cross-attention mechanism to enhance features:

$$F_{ij}^t = \text{Transformer}\left(e_{ij}^t, [e_{ij}^{t-W/2}, \ldots, e_{ij}^{t+W/2}]\right) \tag{5}$$

After transforming the concatenated feature via one of the four baseline approaches above, the resulting output I_{ij}^t between subject i and subject j at the t^{th} frame, within the context of multi-category classification, is represented as:

$$I_{ij}^t = \text{softmax}\left(F_{ij}^t\right) \tag{6}$$

In cases where objects engage in multiple concurrent interactivities, we frame the problem as a multi-category classification task, utilizing binary cross-entropy loss.

Limitations. Despite their simplicity, the limitation of the above methods lies in their ability to capture and represent temporal information in videos. First, these filters have a fixed temporal scale, making it challenging to capture information at multiple temporal resolutions in a single design. Next, they do not inherently capture the spatial or hierarchical relationships between different frames in a video. Therefore, they lack positional information. Those methods typically span their primary operations via a temporal window size. As a result, they have a limited receptive field, which means they can only effectively capture long-range temporal dependencies in videos if they use very deep networks.

5.2. Hierarchical Interlacement Graph

The Limitations of The Monolithic Interlacement Graphs. Monolithic Interlacement Graphs undergo computational bottlenecks when edges span the entirety of a video containing T frames with n objects. Due to their structure, the number of correct edge hypotheses is restricted by the constraint $E_{\text{correct}} \leq 2n$. This implies that each node can correctly associate with at most two other nodes. Hence, the maximum number of potential edges is $E_{\text{potential}} = \frac{n(n-1)}{2}$, indicating a quadratic growth pattern and presenting challenges. Firstly, the computational imposed by this graph limits its scalability, particularly when processing extensive video sequences or handling a significant number of objects. Secondly, while Monolithic Interlacement Graphs efficiently control short-range dependencies, they struggle to capture sparse long-range interactivities. Finally, any temporary occlusions within the frames further strengthen the complexity of understanding, increasing the risk of incorrect interactivities.

Building Temporally-Refined Hierarchical Partitions. The Hierarchical Interlacement Graph (HIG) introduces the Hierarchical Clip Partitioning strategy to address these challenges. Initially, the graph initiates interactivities between consecutive frames, capturing all the activities within two frames. When extending the temporal view, this structure recursively divides the clip into distinct, non-overlapping temporal segments. Progressing through the hierarchy, each interlacement at level L captures interactivities spanning more extended periods. This approach ensures that long-term interactivities are inherited base levels. In addition, the hierarchical structure gains efficiency by converging nodes and edges, considerably reducing the graph"s dimensions. This speeds up processing and improves clarity, particularly for objects that are temporally obscure in the frame. Therefore, building temporally refined hierarchical partitions enables it to navigate the intricate object interactivity, irrespective of their temporal length or complexity.

Number of Hierarchical Levels. We investigates the impact of hierarchical depth on the HIG model"s performance, as depicted in Fig. 2. The model"s standard configuration encompasses n levels, where n equals the total number of video frames minus one. In our ablation study, we explore the model"s performance across reduced hierarchical depths, specifically at levels $L = 1$, $L = n/4$, $L = n/2$, $L = 3n/4$ and $L = n$, corresponding to the configurations shown in Fig. 2. This study aims to ascertain the optimal number of hierarchical levels required for the HIG model to interpret the complex interactions within a video effectively while also determining whether increasing hierarchical levels significantly boosts accuracy or leads to overfitting.

We observe that as the hierarchical level increases, encompassing a more significant portion of the video frame, it ***effectively reduces noise and leads to a higher recall rate***. Tab. 2 reveals that the model can analyze and interpret video content substantially enhanced as it progresses deeper into its hierarchical structure. The HIG model conceptualizes videos as a series of interconnected graphs, each corresponding to a pair of frames, adeptly capturing complex interactions within the video. The key performance indicators, specifically recall and mean recall, evaluated at different thresholds (20, 50, and 100), exhibit a consistent upward trend with the increasing hierarchical depth. This enhancement is particularly notable in figuring out the position at the highest hierarchical level, where there is a significant improvement.

References

[1] 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. 4

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[3] Jingkang Yang, Yi Zhe Ang, Zujin Guo, Kaiyang Zhou, Wayne Zhang, and Ziwei Liu. Panoptic scene graph generation. In *European Conference on Computer Vision*, pages 178–196. Springer, 2022. 4, 5

[4] Jingkang Yang, Wenxuan Peng, Xiangtai Li, Zujin Guo, Liangyu Chen, Bo Li, Zheng Ma, Kaiyang Zhou, Wayne Zhang, Chen Change Loy, et al. Panoptic video scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18675–18685, 2023. 4

[5] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023. 1

[6] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023. 1