Panoptic Video Scene Graph Generation

Jingkang Yang†, Wenxuan Peng‡, Xiangtai Li†, Zujin Guo†, Liangyu Chen†, Bo Li†
Zheng Ma†, Kaiyang Zhou†, Wayne Zhang‡, Chen Change Loy†, Ziwei Liu†
†S-Lab, Nanyang Technological University, Singapore
‡SenseTime Research, Shenzhen, China
https://github.com/Jingkang50/OpenPVSG

Figure 1. An example video from our panoptic video scene graph (PVSG) dataset. The top row shows some keyframes overlaid with the frame-wise panoptic segmentation masks. The timeline tubes underneath the keyframes contain fine, temporal scene graph annotations. The PVSG dataset contains 400 videos (with an average duration of 76.5 seconds), including 289 third-person and 111 egocentric videos.

Abstract

Towards building comprehensive real-world visual perception systems, we propose and study a new problem called panoptic video scene graph generation (PVSG). PVSG is related to the existing video scene graph generation (VidSGG) problem, which focuses on temporal interactions between humans and objects localized with bounding boxes in videos. However, the limitation of bounding boxes in detecting non-rigid objects and backgrounds often causes VidSGG systems to miss key details that are crucial for comprehensive video understanding. In contrast, PVSG requires nodes in scene graphs to be grounded by more precise, pixel-level segmentation masks, which facilitate holistic scene understanding. To advance research in this new area, we contribute a high-quality PVSG dataset, which consists of 400 videos (289 third-person + 111 egocentric videos) with totally 150K frames labeled with panoptic segmentation masks as well as fine, temporal scene graphs. We also provide a variety of baseline methods and share useful design practices for future work.

1. Introduction

In the last several years, scene graph generation has received increasing attention from the computer vision community [15, 16, 24, 48–51]. Compared with object-centric labels like “person” or “bike,” or precise bounding boxes commonly seen in object detection, scene graphs provide far richer information in images, such as “a person riding a bike,” which capture both objects and the pairwise relationships and/or interactions. A recent trend in the scene graph community is the shift from static, image-based scene graphs to temporal, video-level scene graphs [1, 41, 49]. This has marked an important step towards building more comprehensive visual perception systems.

Compared with individual images, videos clearly contain more information due to the additional temporal dimension,
which largely facilitates high-level understanding of temporal events (e.g., actions [14]) and is useful for reasoning [59] and identifying causality [10] as well. However, we argue that current video scene graph representations based on bounding boxes still fall short of human visual perception due to the lack of granularity—which can be addressed with panoptic segmentation masks. This is echoed by the evolutionary path in visual perception research: from image-level labels (i.e., classification) to spatial locations (i.e., object detection) to more fine-grained, pixel-wise masks (i.e., panoptic segmentation [20]).

In this paper, we take scene graphs to the next level by proposing panoptic video scene graph generation (PVSG), a new problem that requires each node in video scene graphs to be grounded by a pixel-level segmentation mask. Panoptic video scene graphs can solve a critical issue exposed in bounding box-based video scene graphs: both things and stuff classes (i.e., amorphous regions containing water, grass, etc.) can be well covered—the latter are crucial for understanding contexts but cannot be localized with bounding boxes. For instance, if we switch from panoptic video scene graphs to bounding box-based scene graphs for the video in Figure 1, some nontrivial relations useful for context understanding like “adult-1 standing on/in ground” and “adult-2 standing on/in water” will be missing. It is also worth noting that bounding box-based video scene graph annotations, at least in current research [15], often miss small but important details, such as the “candles” on cakes.

To help the community progress in this new area, we contribute a high-quality PVSG dataset, which consists of 400 videos among which 289 are third-person videos and 111 are egocentric videos. Each video contains an average length of 76.5 seconds. In total, 152,958 frames are labeled for fine panoptic segmentation and temporal scene graphs. There are 126 object classes and 57 relation classes. A more detailed comparison between our PVSG dataset and some related datasets is shown in Table 1.

To solve the PVSG problem, we propose a two-stage framework: the first stage produces a set of features for each video and the second stage generates rich annotations of panoptic segmentation masks and temporal scene graphs.

Table 1. Comparison between the PVSG dataset and some related datasets. Specifically, we choose three video scene graph generation (VidSGG) datasets, three video panoptic segmentation (VPS) datasets, and two egocentric video datasets—one for short-term action anticipation (STA) while the other for video object segmentation (VOS). Our PVSG dataset is the first long-video dataset with rich annotations of panoptic segmentation masks and temporal scene graphs.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Task</th>
<th>#Video</th>
<th>Video Hours</th>
<th>Avg. Len.</th>
<th>View</th>
<th>#ObjCls</th>
<th>#RelCls</th>
<th>Annotation</th>
<th># Seg</th>
<th>Frame</th>
<th>Year</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cityscapes-VPS [17]</td>
<td>VPS</td>
<td>500</td>
<td>-</td>
<td>-</td>
<td>vehicle</td>
<td>19</td>
<td>-</td>
<td>Panoptic Seg.</td>
<td>3K</td>
<td>2020</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>KITTI-STEP [45]</td>
<td>VPS</td>
<td>50</td>
<td>-</td>
<td>-</td>
<td>vehicle</td>
<td>19</td>
<td>-</td>
<td>Panoptic Seg.</td>
<td>18K</td>
<td>2021</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VIP-Seg [24]</td>
<td>VPS</td>
<td>3,536</td>
<td>5</td>
<td>5s</td>
<td>3rd</td>
<td>124</td>
<td>-</td>
<td>Panoptic Seg.</td>
<td>85K</td>
<td>2022</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PVSG</td>
<td>PVSG</td>
<td>400</td>
<td>9</td>
<td>77s</td>
<td>3rd + ego</td>
<td>126</td>
<td>57</td>
<td>Panoptic Seg</td>
<td>150K</td>
<td>2023</td>
<td>-</td>
<td>VidOR + Ego4D + EPIC-KITCHENS</td>
</tr>
</tbody>
</table>

In summary, we make the following contributions to the scene graph community:

1. A new problem: We identify several issues associated with current research in scene graph generation and propose a new problem, which combines video scene graph generation with panoptic segmentation for holistic video understanding.

2. A new dataset: A high-quality dataset with fine, temporal scene graph annotations and panoptic segmentation masks is proposed to advance the area of PVSG.

3. New methods and a benchmark: We propose a two-stage framework to address the PVSG problem and benchmark a variety of design ideas, from which valuable insights on good design practices are drawn for future work.

2. Related Work

Scene Graph Generation Given an image, the scene graph generation (SGG) task expects the model to output a scene graph representation, where nodes represent objects and edges represent relations between objects. To localize object instances, the nodes should be grounded by the bounding boxes [48]. Classic scene graph generation methods have been dominated by the two-stage pipeline that consists of object detection and pairwise predicate estimation [38, 39, 48, 56, 58]. Recent works on one-stage methods [4, 23, 50] provide simpler models that output semantically diverse relation predictions. Though the prevalent
The PVSG dataset contains 400 third-person and ego-centric videos from diverse environments, as shown in (a). The statistics of object classes and relation classes are shown in (b) and (c).

Video Scene Graph Generation

Shang et al. [35] first proposes Video Scene Graph Generation (VidSGG) and released ImageNet-VidVRD dataset. They generate object tracklet proposals and short term relations on overlapping segments. Subsequently, they greedily associate these relation triplets into video level. Several works follow the track-to-detect paradigm with spatio-temporal graph and graph convolutional neural networks [26, 31], or multiple hypothesis association [37]. MVSGG [49] investigates the spatio-temporal conditional bias problem in VidSGG. They perform a meta training and testing process, constructing the data distribution of each query set w.r.t. the conditional biases. TRACE [41] decouples the context modeling for relation prediction from the complicated low-level entity tracking. [1] raises the issue of domain shift between image and video scene graphs. They exploit external commonsense knowledge to infer the unseen dynamic relationship, and employ hierarchical adversarial learning to adapt from image to video data distributions. Embodied Semantic SGG [24] exploits the embodiment of the intelligent agent to autonomously generate an appropriate path by reinforcement learning [9] to explore an environment.

Video Panoptic Segmentation


3. The PVSG Problem

The goal of the PVSG problem is to describe a given video with a dynamic scene graph, with each node associated with an object and each edge associated with a relation in the temporal space. Formally, the input of the PVSG model is a video clip $V \in \mathbb{R}^{T \times H \times W \times 3}$, where $T$ denotes the number of frames, and the frame size $H \times W$ should be consistent across the video. The output is a dynamic scene graph $G$. The PVSG task can be formulated as follows,

$$Pr(G | V) = Pr(M, O, R | V).$$ (1)
More specifically, \( \mathbf{G} \) comprises the binary mask tubes \( \mathbf{M} = \{ \mathbf{m}_1, \ldots, \mathbf{m}_n \} \) and object labels \( \mathbf{O} = \{ o_1, \ldots, o_n \} \) that correspond to each of the \( n \) objects in the video, and their relations in the set \( \mathbf{R} = \{ r_1, \ldots, r_l \} \). For object \( i \), the mask tube \( \mathbf{m}_i \in \{0, 1\}^{T \times H \times W} \) collects all its tracked masks in each frame, and its object category should be \( o_i \in \mathbb{C}^O \). For all objects in a frame \( t \), the masks do not overlap, i.e., \( \sum_{i=1}^{n} \mathbf{m}_i^t \leq 1^{H \times W} \). The relation \( r_i \in \mathbb{C}^R \) associates a subject and an object with a predicate class and a time period. \( \mathbb{C}^O \) and \( \mathbb{C}^R \) means the object and predicate classes.

**Metric** In practice, the output of the PVSG task is to predict a set of triplets to describe the input video. Take a triplet as an example, which contains a relation \( r_i \) from \( t_1 \) to \( t_2 \), associates the subject with its class category \( o_s \) and mask tube \( \hat{\mathbf{m}}_s(t_1:t_2) \), and an object with \( o_o \) and \( \hat{\mathbf{m}}_o(t_1:t_2) \). \( \hat{\mathbf{m}}(t_1:t_2) \) denotes the mask tube \( \mathbf{m} \) span across the period of \( t_1 \) to \( t_2 \).

To evaluate the PVSG task, we follow the classic SGG and VidSGG paper and use the metrics of the R@K and mR@K, which calculates the triplet recall and mean recall given the top \( K \) triplets from the PVSG model. A successful recall of a ground-truth triplet \((o_s, \hat{\mathbf{m}}_s(t_1:t_2), o_o, \hat{\mathbf{m}}_o(t_1:t_2), \hat{r}(t_1:t_2))\) should meet the following criteria: 1) the correct category labels of the subject, object, and predicate; 2) the volume IOU between the predicted mask tubes \((\mathbf{m}^{(t_1:t_2)}_s, \hat{\mathbf{m}}_s(t_1:t_2)), \mathbf{m}^{(t_1:t_2)}_o, \hat{\mathbf{m}}_o(t_1:t_2)\) and the ground-truth tubes \((\mathbf{m}^{(t_1:t_2)}_s, \hat{\mathbf{m}}_s(t_1:t_2)), \mathbf{m}^{(t_1:t_2)}_o, \hat{\mathbf{m}}_o(t_1:t_2)\) should be individually over 0.5. When the previous two criteria are met, a soft recall score of the time IOU between \((t_1, t_2)\) and \((t_1, t_2)\) is recorded.

Please notice the nuance of the PVSG metrics compared with VidSGG metrics for VidOR [34]. For a case where a child stop-and-go several times in a video, different from VidOR which considers several “child-1 walking on ground” triplets, our PVSG metrics only consider the triplet once, but with a scattered time span. This small change avoids some relations dominating the metrics by repeating.

### 4. The PVSG Dataset

In this section, we first summarize the existing VidSGG datasets and highlight their problems. Then, we introduce the overview and statistics of our PVSG dataset, and its annotation pipeline.

#### 4.1. Connecting Existing Datasets to PVSG

To select candidate video clips for the PVSG dataset, a go-to option is to borrow the videos from other VidSGG datasets. Table 1 lists three classic VidSGG datasets chronologically. While the limited size of their first VidSGG dataset, ImageNet-VidVRD [35], Shang et al. collects 10K videos from the user-uploaded dataset YFCC100M [42] and generate a large-scale VIDOR dataset [34], with dense object and relation annotation. Ji et al. also introduces a large-scale dataset Action Genome (AG) based on a diverse, crowd-sourcing Charades dataset [36]. While Charades provides a novel solution to gather large-scale, less-biased video datasets by asking people to act based on the generated script, the curated scripts usually produce random action series, such as a man running out of the room and running back for no reason. Also, the performance traces turn out to be heavy in the dataset. These shortcomings limit the potential of the community to explore contextual logic and reasoning in videos.

Alternative video datasets that lean toward logic reasoning and video scene understanding are instruction datasets or movie datasets. However, these datasets are either full of close-up shots (e.g., Something-Something [11], HowTo100M [29]) or cut shots (e.g., MOMA [27], HC-STVG [40]). In fact, humans rely on unpolished videos to form an essential understanding of the world. In this sense, we find that the unedited, natural, and diverse VidOR [34] videos are a good candidate for learning the visual essence as well as keeping the potential of contextual logic exploration. While the videos presented above showcase a third-person perspective, egocentric videos have gained popularity due to their practicality in autonomous driving [54], robotic decision-making [37], and in the metaverse [30]. In particular, a subset of the Ego4D dataset [12] is suitable for exploring logical relationships and modeling, as it supports short and long-term action anticipation tasks. Additionally, the Epic-Kitchens [6] dataset is focused on the kitchen scenario and offers rich action data. Its subset, the VISOR dataset, includes video object segmentation (VOS) annotation, which partially matches the PVSG scope, though its relations are not yet annotated.

Another dataset category that is closely related to the PVSG problem is the video panoptic segmentation (VPS) datasets. Popular VPS datasets include Cityscapes-VPS [17] and KITTI-STEP [45]. However, the relations in the self-driving scenarios are limited, which is not suitable for the PVSG task. Although the recent VIP-Seg [28] provides a more diverse VPS dataset, each video only lasts around 5 seconds, which also lacks temporal relations.

With all the rationale above, we eventually decide to combine three video sources to the PVSG dataset, which are VidOR, Ego4D-STA, and Epic-Kitchens-100 (including some videos from VISOR).

#### 4.2. Dataset Statistics

Figure 2 displays the statistics of the PVSG dataset, which consists of 400 videos, including 289 third-person videos from VidOR and 111 egocentric videos from Epic-Kitchens and Ego4D. Among the videos, 62 videos feature birthday celebrations, while 35 videos center around ceremonies, providing rich content for contextual logic and reasoning. Furthermore, the dataset includes numerous videos related to sports and pets, featuring complex and diverse ac-
Figure 3. **PVSG Dataset Annotation Pipeline.** The construction of PVSG dataset can be divided into VPS annotation and relation annotation. For VPS annotation, we select a few key frames and use an off-the-shelf video object segmentation (VOS) model AOT [53] to propagate the annotated objects to the whole video, and then perform frame-level mask fusion using the predefined layer order to obtain a coarse VPS annotation for further revision. The relations are annotated based on the description of the key information in the video.

### 4.3. Dataset Construction Pipeline

Creating the PVSG dataset is never a trivial task considering that both video panoptic segmentation and relation annotations are required. This section describes how the PVSG dataset is collected and annotated.

**Step 1: Video Clip Selection**

To get rid of the drawbacks of the current datasets (i.e., the unnatural videos in AG [15] without logical script, and the static and short videos from the VPS datasets), we carefully select around 300 long, daily, unedited videos with a logical storyline. In addition, to encourage the VidSGG models to be practical on egocentric videos, we also select around 100 videos from Epic-Kitchens and Ego4D with the same criteria. Videos with too many small and trivial objects are also discarded for VPS annotation purposes. We hope the selected videos could greatly encourage the exploration of video recognition, understanding, and reasoning.

**Step 2: VPS Annotation**

Notice that the PVSG videos have more than 300 frames on average and 150K in total, it is impossible to annotate panoptic segmentation for each frame. After iterations and improvements, we finalize a human-machine collaborative VPS annotation pipeline, depicted in Figure 3. In a nutshell, we largely rely on an off-the-shelf VOS model called AOT [53] for the human-machine interactive annotation process.

**Coarse VPS Annotation:** With a few well-annotated object masks in the first frame, the AOT [53] is able to propagate the masks to later frames. With this strong automatic tool, we design a pipeline to obtain coarse VPS annotation. For the example video in Figure 3 (actions 1-3), we first identify several key objects to annotate, and also identify key frames where the selected objects have a clear and whole appearance. To identify key objects, our annotators need to select all objects and background to address “panoptic”, except those messy and unrelated ones. After annotating these key objects on their corresponding frames, we use AOT based on the frames to propagate the mask, both forward and backward. Thus, each frame will yield a whole mask video. To merge those mask videos into one, the layer order should be considered beforehand, i.e., the objects from which layer should be put in front. In fact, the decision of the layer order is made with key frame selection.

**Fine VPS Annotation:** Based on the coarse VPS annotation, we conduct several rounds (more than 5) of the human-machine interactive revision process to obtain the final annotation. We rely on the multi-frame panoptic segmentation propagation mode of the AOT algorithm [53], which interpolates the entire video masks based on several frames with the entire panoptic segmentation. The quality of interpolation increases with more intermediate frames. To accelerate the revision process, we revise the transit frames first, as shown in action 5 in Figure 3. Typical examples of poor masks include incorrect tracking masks and boundaries that deviate significantly from the object.

**Step 3: Relation Annotation**

We annotate temporal re-
The two-stage framework to solve the PVSG task. The goal of the first stage is to obtain the video panoptic segmentation mask for each object, as well as its corresponding video-length feature tube. Two options are provided to achieve the goal. The second stage predicts pairwise relations based on all the feature tubes from the first stage. Four options are provided for a comprehensive comparison.

5. Methodology

In this section, we introduce the two-stage pipeline to address the PVSG problem. We provide two options for the first stage and four options for the second stage.

5.1. Stage One: Video Panoptic Segmentation

Given a video clip input $V \in \mathbb{R}^{T \times H \times W \times 3}$, the goal of VPS is to segment and track each pixel in a non-overlap manner. Specifically, the model predicts a set of video clips $\{y_i\}_{i=1}^N = \{(m_i, p_i(c))\}_{i=1}^N$, where $m_i \in \{0, 1\}^{T \times H \times W}$ denotes the tracked video mask, and $p_i(c)$ denotes the probability of assigning class $c$ to a clip $m_i$. $N$ is the number of entities, which includes thing classes and stuff classes.

We present two strong baselines for the first stage of VPS processing. In particular, we adopt the state-of-the-art image segmentation baseline [3] with an extra tracker and the improved video panoptic segmentation method [25]. For the former, it processes the video frames individually. For the latter, it processes the video frames across the temporal dimension, with a nearby frame as the reference frame.

IPS+T: Image Panoptic Segmentation With Tracker

We adopt strong Mask2Former [3] as our baseline method since it is a mask-based transformer architecture. It contains a transformer encoder-decoder architecture with a set of object queries, where the object queries interact with encoder features via masked cross-attention. Given an image $I$, during the inference, the Mask2Former directly outputs a set of object queries $\{q_i\}_{i=1, \ldots, N}$, where each object query $q_i$ represents one entity. Then, two different multiple layer perceptrons (MLPs) project the queries into two embeddings for mask classification and mask prediction, respectively. During training, each object query is matched to ground truth masks via masked-based bipartite matching.

We first fine-tune the Mask2Former on our dataset. Then, we test the model with an extra tracker [44]. In particular, we first obtain panoptic segmentation results of each frame. Then we link each frame via using UniTrack [44] to obtain the final $N$ tracked video cubes for each clip. Therefore, a query tube is obtained. For the object $i$ at the $t$-th frame, the query is noted as $q^t_i$. We use $Q^t_{\{t_1, t_2\}}$ to denote the set of queries $\{q^t_i\}_{t=t_1}^{t_2}$ and $Q_i$ denotes the query tube in the entire video.

VPS: Video Panoptic Segmentation Baseline

For video baselines, we modify the previous state-of-the-art method Video K-Net [25] into Mask2Former framework. We first replace the backbone and neck in Video K-Net [25] with Mask2Former feature extractor. Then we use the temporal contrastive loss to directly on the output queries from the last layer of the decoder. In particular, given two frames, we first obtained the object queries from both frames, and then we sent them into an embedding layer (a shared MLP) to obtain association embeddings. We adopt the same tracking loss used in [25] to supervise the association embeddings. The embeddings are close if they are the same object, otherwise, they are pulled away.

During the training, the two nearby frames are sent to the model to learn the association embedding. During the inference, the learned association embeddings are used to perform instance-wise tracked cues to match each thing masks frame by frame in an online manner. Compared with the image baseline, our video baseline considers the tempo-
Table 2. Comparison between all two-stage PVSG baselines. We provide two options for the first stage and four options for the second stage, as described in Section 3. The results show that using the basic image-based method in the first stage with the transformer encoder in the second stage can achieve the optimal recall.

<table>
<thead>
<tr>
<th>Method</th>
<th>PVSG Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stage-1</td>
</tr>
<tr>
<td>IPS+T [3, 44]</td>
<td></td>
</tr>
<tr>
<td>Vanilla</td>
<td>2.35 / 1.22</td>
</tr>
<tr>
<td>Handcrafted Window</td>
<td>2.56 / 1.24</td>
</tr>
<tr>
<td>1D Convolution</td>
<td>2.79 / 1.24</td>
</tr>
<tr>
<td>Transformer Encoder</td>
<td>4.02 / 1.75</td>
</tr>
<tr>
<td>VPS [3, 25]</td>
<td></td>
</tr>
<tr>
<td>Vanilla</td>
<td>0.52 / 0.24</td>
</tr>
<tr>
<td>Handcrafted Window</td>
<td>0.54 / 0.27</td>
</tr>
<tr>
<td>1D Convolution</td>
<td>0.60 / 0.27</td>
</tr>
<tr>
<td>Transformer Encoder</td>
<td>0.75 / 0.36</td>
</tr>
</tbody>
</table>

5.2. Stage Two: Relation Classification

The object query (feature) tubes \( \{ Q_i \}_{i=1}^{N} \) serve as a link between the first and second stages. Object tubes are paired with each other in their intersections in the second stage, as in Figure 4. Specifically, as in Figure 4 (b), we first concatenate the query pairs. Next, we mainly introduce four operations to process the relations between feature pairs.

**Vanilla: Fully-Connected Layer** Begin with the most basic version, the pairwise feature fusion is followed by a straightforward fully-connected layer on the fused features. In this scenario, some objects may have several interactions occurring simultaneously, we define the issue as a multi-label classification job with binary cross-entropy loss.

**Handcrafted Filter** To further consider the temporal information, we design a simple kernel to gather the information from the context in nearby frames. By default, the handcrafted filter is a simple vector of \([1/4, 1/2, 1/4, 1/2, 1/4]\) with a window size of 5. Note that the filter is also required during inference.

**1D-Convolutional Layer** To improve the handcrafted filter, we also utilize a learnable 1D-Convolutional layer to capture temporal information. The kernel sizes are set to 5 in 3 layers.

**Transformer Encoder** A transformer encoder [43] is also suitable in this scenario. We utilize a 3-layer transformer block with positional embeddings in the entire fused query feature to capture temporal information via cross-attention between frames.

6. Experiments

In this section, we show the experimental results for the PVSG dataset. We split the dataset with 360 videos for training and 40 videos for testing. For both IPS+T and VPS, we adopt Mask2Former upon the ResNet-50 [13] backbone with 12 training epochs, which takes 12 hours and 48 hours on 8 V-100 GPUs, respectively. The training time of the second stage is shorter than an hour on single V-100 GPU.

The experimental results to compare two stage-one options and four stage-two options are shown in Table 2. We first take a look at the second stage. The transformer encoder obtains the optimal results regardless of the first-stage options, showing the effectiveness of temporal information fusion. Besides, the 1D convolutional layer achieves better results than the handcrafted window, showing that modeling with learning parameters in the second stage is worth exploring. Considering the harsh recall criteria described in Section 3, even the most basic vanilla method can achieve a few recall scores, showing that the PVSG task is solvable with a decent first-stage model. We hope that the second stage alone could advance research efforts on visual temporal predictions.

We then discuss the influence of the first stage. According to Table 2, the end-to-end VPS model seems to underperform the IPS+T baseline. Although the VPS models are shown effective on the existing VPS datasets such as Cityscape-VPS and Kitty-STEP, videos in the PVSG dataset are longer and more dynamic (frequent and large camera view shift), which seems to bring new challenges for the VPS community. According to Figure 5, the end-to-end VPS model fails to achieve a higher tracking performance, which seems severely affect its performance on the PVSG task.

7. Conclusion, Challenges, and Outlook

In this paper, we introduce a new PVSG task, a new PVSG dataset with several baselines to address the new task, in hope of encouraging comprehensive video understanding and trigger more interesting downstream tasks such as visual reasoning. Here we discuss the challenges
Figure 5. The visualization of the top triplets generated by PVSG models. The result shows that the IPS+T method is able to predict a better-quality video panoptic mask. The VPS baseline is shown unable to perform well on tracking (e.g., the tracking of the child switched in the later frames), which leads to its low performance in the PVSG task.

Challenges Real-world data often exhibit long-tailed heteroscedastic distributions across objects and relations, as shown in Figure 2. The PVSG models are expected to predict informative and diverse relations, rather than being obsessed with statistically common relations. Yet another challenge the PVSG models faces is the aleatoric uncertainty in verbal relation descriptions. For example, “playing with” can be overlapping with “chasing” when it describes two kids chasing each other. Such nuances from canonical languages introduce intrinsic label noises in prevailing video event datasets, including PVSG. Another important challenge that the PVSG dataset introduces is video panoptic segmentation. With the video with a large view shift, the VPS models are expected to have a better performance on tracking and segmentation.

Outlook on Video Perception and Reasoning We foresee the potential of PVSG in bridging video scene perception and reasoning. While current video question-answering datasets lack pixel-level segmentation masks that refine (sometimes determine) the relations between object pairs, the inclusion of such dense annotations is critical to video reasoning tasks. PVSG is related to social intelligence, with rich event annotations in human behaviors and dynamics. In the same spirit, it is also related to human-object interaction (HOI) that dense labels are capable to capture very subtle visual differences in the scene.

Potential Negative Societal Impacts This work releases a dataset containing human behaviours, posing possible gender and social biases inherently from data. Potential users are encouraged to consider the risks of overlooking ethical issues in imbalanced data, especially in underrepresented minority classes.

Acknowledgement This study is supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 (MOE-T2EP20221-0012), NTU NAP, and under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s). We are also grateful to SuperAnnotate1 for providing an outstanding annotation platform and excellent customer service. Special thanks to Binzhu Xie and Zitang Zhou from Beijing University of Posts and Telecommunications for their dedicated leadership of the annotation team.

1https://www.superannotate.com/
References

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kala-tidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV, 2017. 3
[22] Lin Li, Long Chen, Yifeng Huang, Zhimeng Zhang, Songyang Zhang, and Jun Xiao. The devil is in the labels: Noisy label correction for robust scene graph generation. In CVPR, 2022. 3
[29] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic.


[34] Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xin Yang, and Tat-Seng Chua. Annotating objects and relations in user-generated videos. In ICMR, 2019.


[59] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In Proceed-
ings of the European conference on computer vision (ECCV),