Learning 4D Panoptic Scene Graph Generation from Rich 2D Visual Scene

Supplementary Material

Overview

The appendix presents more details and additional results not included in the main paper due to page limitation. The list of items included are:

- Potential Limitation and Future Work in §A;
- Extended Task Definition in §B;
- Framework Architecture in §C;
- Detailed Training Procedure in §D;
- Detailed System Inference in §E;
- Dataset Specification in §F;
- Detailed Experimental Implementations in §G;
- Additional Experiments in §H.

A. Potential Limitation and Future Work

A.1. Potential Limitations of 4D-LLM

Despite its promising performance, 4D-LLM faces certain limitations. First, the dataset annotations used for training and evaluation may suffer from incomplete labeling or annotation errors, which may hinder the model's ability to learn accurate representations and achieve optimal performance. These limitations in the dataset may result in missed detections or incorrect relationships within the generated scene graphs, reducing the reliability of the model in practical applications. The proposed approach, chained inference, which leverages the inherent reasoning capabilities of LLMs, has shown the potential to alleviate this issue by improving the consistency and robustness of the model's predictions.

Another potential limitation is the model's capacity to handle extreme long-term 4D scenes. Understanding and processing extended temporal sequences in dynamic 4D environments remains challenging due to the complexity of capturing and reasoning about long-term dependencies and interactions. Current models, including 4D-LLM, may struggle to maintain coherence and accuracy in such scenarios, which is critical for tasks requiring an understanding of prolonged events or activities, such as surveillance or continuous monitoring.

A.2. Future Works

Several directions can be pursued to enhance and expand the applications of 4D-LLM. One significant application area is robotics, where processing and understanding rich 4D scene information could greatly enhance robotic perception, decision-making, and task execution. For instance, by leveraging 4D-LLM, robots could gain a comprehensive understanding of their surrounding environments, enabling them to make informed decisions and adapt their actions in real time. This capability is particularly relevant for tasks such as autonomous navigation in dynamic and complex environments, where accurate scene understanding is essential for planning and executing tasks with high precision and safety.

Beyond robotics, 4D-LLM holds potential in virtual environments, such as serving as an autonomous agent in video games like GTA. Unlike traditional task completion systems that passively respond to predefined inputs, 4D-LLM could actively perceive and interpret its surroundings, dynamically interacting with the environment to achieve objectives. This transition from passive to active perception and decision-making highlights a shift toward greater autonomy and adaptability in AI systems.

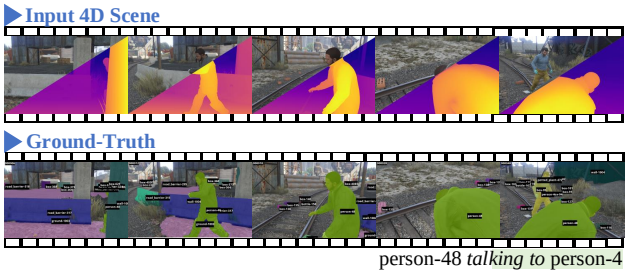

Figure 1. Input and output of the 4D panoptic scene graph (4D-PSG) generation task.

B. Extended Task Definition

As shown in Fig. 1, given a 4D scene, specifically represented as a sequence of RGB-D frames $\mathcal{I} \in \mathbb{R}^{T\times H\times W\times 4}$, where T denotes the number of frames and each frame has dimensions $H \times W \times 4$, our objective is to generate a dynamic panoptic SG $\mathcal{G} = \{\mathcal{O}, \mathcal{M}, \mathcal{R}\}$. The RGB-D sequence can also be treated as two parallel sequences: RGB images and single-channel depth images. Here, $\mathcal{O} = \{\boldsymbol{o}_n\}_{n=1}^{N}$ are the set of objects present in the scene, for example, in Fig. 1, the object, "*person-48*", "*person-4*", "*road-barrier-295*", "*wall-1004*", etc. $\mathcal{M} = \{\boldsymbol{m}_n\}_{n=1}^{N}$ denotes the corresponding binary mask tubes, where $\boldsymbol{m}_i \in \{0,1\}^{T\times H\times W}$ tracks the spatial extent of object o_i over time T, as illustrated by the color-coded regions in Fig. 1. The relation set $\mathcal{R} = \{\boldsymbol{r}_k\}_{k=1}^{K}$ defines interactions between objects, with each $\boldsymbol{r}_k$ linking a subject and an object through a predicate class over a specific period (t_s, t_e). For instance, the relation, "*person-48 talking to person-4*", is recognized, with its temporal duration represented by the length of the corresponding color block.

C. Framework Architecture

Here, we detailed the framework architecture of the three estimators employed in the spatial-temporal 2D-to-4D tran-

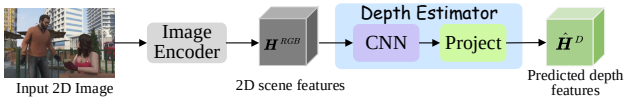

Figure 2. The framework of Depth Estimator.

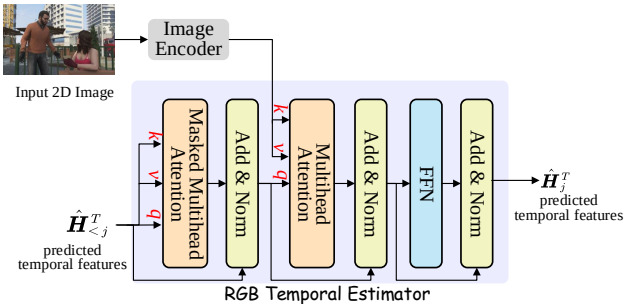

Figure 3. The framework of RGB Temporal Estimator.

scending mechanism.

Depth Estimator. As depicted in Fig. 2, given an input 2D image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$, an image encoder is to model the input image and yield 2D scene features $\boldsymbol{H}^{RGB} \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times d}$, where p represents the patch size, d is the feature dimensionality. Then, we implement the convolution (CNN) with a 3×3 kernel and then a projector using 1×1 convolutions to project the input representation to match the dimension of ground-truth depth features $\boldsymbol{H}^{D}$.

RGB Temporal Estimator. As shown in Fig. 3, given an input 2D image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$, an image encoder is to model the input image and yield 2D scene features $\boldsymbol{H}^{RGB} \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times d}$, where p represents the patch size, d is the feature dimensionality. Inspired by [19], we design the RGB Temporal Estimator F_{rte} as an autoregressive transformer, which is applied to predict the temporal features. The masked multi-head attention, combined with the fact that the output features are offset by one position, ensures that the predictions for position j can depend only on the known outputs at positions $< j$. This autoregressive mechanism enables effective modeling of temporal dependencies. The probabilistic formulation is as follows:

$$p(\hat{\boldsymbol{H}}^{T}, \boldsymbol{H}^{RGB}, F_{rte}) = \prod_{j} F_{rte}(\hat{\boldsymbol{H}}_{j}^{T} | \hat{\boldsymbol{H}}_{<j}^{T}, \boldsymbol{H}^{RGB}), \quad (1)$$

where $\hat{\boldsymbol{H}}_{j}^{T}$ denotes the predicted temporal features at step j, and $\hat{\boldsymbol{H}}_{<j}^{T}$ refers to the features predicted for previous steps.

Depth Temporal Estimator. As shown in Fig. 4, given an input depth image $\mathcal{I} \in \mathbb{R}^{H\times W\times 1}$, a depth encoder is to model the input image and yield depth features $\boldsymbol{H}^{D} \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times d}$, where p represents the patch size, d is the feature dimensionality. Similarly, the Depth Temporal Estimator F_{dte} employs the same architecture but different parameters

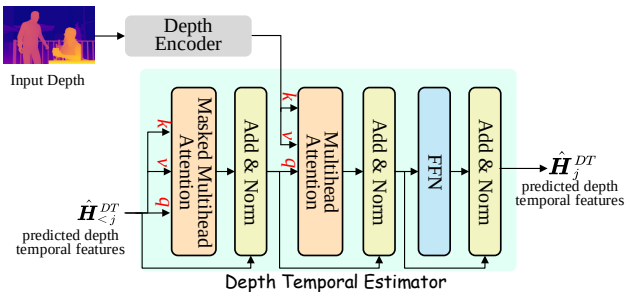

Figure 4. The framework of Depth Temporal Estimator.

to predict the depth temporal features:

$$p(\hat{\boldsymbol{H}}^{DT}, \boldsymbol{H}^{D}, F_{dte}) = \prod_{j} F_{dte}(\hat{\boldsymbol{H}}_{j}^{DT} | \hat{\boldsymbol{H}}_{<j}^{DT}, \boldsymbol{H}^{D}). \quad (2)$$

D. Extended Training Framework

This section details the training process, including the datasets, quantities, and parameters utilized at each step.

► **Step 1: 4D Scene Perception Initiation Learning.** The training begins with 4D Scene Perception Initiation Learning, designed to establish a foundational understanding of 4D scenes in the LLM for generating 4D-PSG. In this step, we utilize the PSG4D dataset to train the 4D-LLM. The input comprises 4D scenes, while the supervision signals are the ground-truth 4D-PSGs, which include textual scene graph triplets and corresponding object mask tubes. The optimization is performed according to Eq.(1).

► **Step 2: 2D-to-4D Scene Transcending Learning.** In this step, we focus on 2D-to-4D Scene Transcending Learning to enable the transition from 2D to 4D scene. This process is further divided into three substeps:

1) RGB-to-Depth Transcending Learning: We utilize 200K depth estimation samples from the DIML dataset [2] to train the depth estimator. The input comprises 2D RGB images, with corresponding ground-truth depth images providing supervision. The optimization follows Eq. (2).

2) RGB Temporal Learning: For temporal learning, we employ 288K video data from the AG dataset. The first frame of each video serves as the input 2D RGB image, and the subsequent frames are used as ground-truth supervision for optimizing the RGB Temporal Estimator, guided by Eq. (4).

3) Depth Temporal Learning: We leverage depth sequences from the PSG4D dataset. Specifically, the first depth image in each sequence is used as the input, and the remaining depth images in the sequence are utilized as ground-truth for optimizing the Depth Temporal Estimator, following Eq. (5).

These three substeps are independent and can be conducted concurrently, ensuring an efficient training process for the 2D-to-4D transcending mechanism.

► **Step 3: Pseudo 4D Scene Transfer Initiation Learning.** In this step, we leverage 3K samples from the PSG4D dataset for learning. Specifically, each 4D scene in the training data is firstly decomposed into three components: 3D scenes, video sequences (i.e., RGB sequences), and depth sequences. These components are then used to train the 2D-to-4D Scene Transcending Module, explicitly optimizing the Depth Estimator, RGB Temporal Estimator, and Depth Temporal Estimator. Secondly, the pseudo-4D scenes generated using the trained 2D-to-4D transcending module serve as input to the 4D-LLM, which predicts the final 4D-PSGs, i.e., SG triplets and mask tubes. Ground-truth 4D-PSGs are employed as supervision to optimize the 4D-LLM further. The overall loss function integrates these components, as detailed in Eq. (7), ensuring cohesive optimization across the whole framework.

► **Step 4: Large-scale Visual Scene Transfer Learning.** In this step, we leverage a large-scale dataset consisting of 150K 2D-SG samples, including VG [9] and PSG [20], to train the 4D-LLM. The process begins by feeding 2D scenes into the 2D-to-4D Scene Transcending Module, transforming the input into representations suitable for the 4D-LLM. The 4D-LLM then interprets these representations and generates the corresponding 2D-PSGs. The predicted 2D-PSGs are supervised using the ground-truth 2D-PSGs, ensuring accurate SG generation.

► **Step 5: 4D Scene Fine-tuning.** To ensure optimal model performance, we incorporate an additional training step focused on 4D scene fine-tuning. In this step, we repeat the training process outlined in Step 1 using the P4G4D dataset, allowing the model to further refine its understanding of 4D scenes and enhancing its ability to generate accurate 4D-PSGs.

E. System Inference

To improve the quality and address out-of-vocabulary (OOV) issues in 4D-PSG generation, we employ a chained inference mechanism during the inference phase. The inference process is divided into four sequential stages:

- **Inference stage 1: Object Description and Categorization.** In this stage, the input 4D scene is analyzed to identify all objects present. To handle OOV issues, the LLM first generates detailed descriptions of each object before assigning them specific categories, ensuring a robust recognition, even for unseen or ambiguous objects.
- **Inference stage 2: Semantic Relation Identification.** Based on the identified objects, the LLM determines which object pairs exhibit semantic relationships, establishing the foundation for constructing meaningful scene graphs.
- **Inference stage 3: Precise Relation Description.** To refine the semantic relationships, the LLM generates predicates that offer precise and contextually relevant descriptions of the interactions between object pairs. This step avoids overly general or coarse-grained predicates, ensuring a higher granularity and interpretability.
- **Inference stage 4: Temporal Span Determination.** For object pairs with confirmed semantic relationships, the model further infers the temporal span during which these relationships are valid within the given 4D scene.

To enhance the model's reasoning ability and comprehension of complex instructions, we employ in-context learning throughout the chained inference process. Below, we detail the prompts used to guide the LLM effectively.

Chained Scene Graph Inference

Input Data: 4D Scene, the duration
Instruction: You are a scene expert with professional skills in generating an SG triplets sequence. You follow these four detailed steps to ensure a logical, step-by-step approach to SG generation:

Inference stage 1: Object Description and Categorization. For each object in the scene, do not immediately output its name. Instead, start by describing each object in detail. Provide a description of each object based on its appearance, shape, structure, and any unique characteristics observed in the scene. After giving a detailed description, assign a category to the object that best fits the objects (e.g., "person", "table", "chair", etc.).
Expected Output: (description, object$_1$), $\cdots$

Inference stage 2: Semantic Relation Identification. Based on the identified objects, analyze which pairs of objects may have semantic relations. Consider spatial positioning, interactions, and any logical connections that might exist between them. Identify only pairs that have a meaningful relationship, and briefly explain why these pairs might be related.
Expected Output: (object$_i$, object$_j$), $\cdots$

Inference stage 3: Precise Relation Description For each object pair identified in Step 2, describe the exact nature of the relation between the two objects as precisely as possible. Use clear, concise language to specify the relation type (e.g., "sitting on," "holding," "near," etc.) and provide additional context if necessary to ensure the relation is unambiguous.
Expected Output: (object$_i$, relation$_k$ object$_j$), $\cdots$

Inference stage 4: Temporal Span Determination For each identified relation, determine its duration or time span. Indicate if the relation is continuous, occurs intermittently, or exists only at a specific moment within the scene. Use a numerical value for the duration, such as a time interval (e.g., (0.1, 0.7))
Expected Output: (object$_i$, relation$_k$ object$_j$, start_time, end_time), $\cdots$

Final Output Format: For each object pair and relation, generate SG triplets in the following format:
Expected Output: (object$_i$, relation$_k$ object$_j$, start_time, end_time), $\cdots$

In-context Examples

[Example-1]
Input Data: 4D Scene  , the duration

Inference stage 1: Object Description and Categorization.
Object 1:
Description: A person wearing a white T-shirt, gray pants, and black gloves. They are wearing a dark blue helmet and appear to be running on a sandy field.
Category: Person
Object 2:
Description: A person wearing a green shirt, gray pants, and a green baseball cap. They are holding a baseball glove and appear to be positioned slightly behind Object 1.
Category: Person
Object 3:
Description: A green outfield area with a mix of trees in the background.
Category: Field
Object 4:
Description: A sandy dirt area forming the base path of the field. *Category*: Ground

Inference stage 2: Semantic Relation Identification.
(Person, Person) - The runner (Object 1) is running toward the fielder (Object 2), indicating a potential play interaction.
(Person, Ground) - The runner (Object 1) is running along the base path (Object 3).
(Person, Ground) - The fielder (Object 2) is positioned on or near the base path (Object 3).
(Ground, Field) - The Ground (Object 3) is part of the larger field (Object 4).

Inference stage 3: Precise Relation Description
(Person 1, running toward, Person 2)
(Person 1, running along, Ground)
(Person 2, standing near, Ground)
(Ground, part of, Field)

Inference stage 4: Temporal Span Determination
(Person 1, running toward, Person 2, start time: 0.2, end time: 0.8), (Person 1, running along, Ground, start time: 0.1, end time: 0.9), (Person 2, standing near, Ground, start time: 0.0, end time: 1.0), (Ground, part of, Field, start time: 0.0, end time: 1.0).

Final Output Format:
(Person 1, running toward, Person 2, 0.2, 0.8) (Person 1, running along, Ground, 0.1, 0.9) (Person 2, standing near, Ground, 0.0, 1.0) (Ground, part of, Field, 0.0, 1.0)

F. Dataset Specification

PSG4D. This dataset [21] contains 2 subsets: **(a) PSG4D-GTA** selected from the SAIL-VOS 3D [6] dataset, containing contains 67 videos with an average length of 84 seconds, amounting to 27,700 RGB-D images, 28.3 billion point clouds, and comprises 35 object categories, and 43 relationship categories; **(b) PSG4D-HOI** from HOI4D [13] dataset, including 2,973 videos with an average duration of 20 seconds, equating to 891,000 RGB-D images across 282 indoor scenes. This dataset includes 46 object categories and 15 object-object relationship categories.

Visual Genome (VG). We leverage the original VG dataset [9] for training, which contains the 5,996 types of objects, 1,014 types of predicates, and approximately 108k images.

Panoptic Scene Graph (PSG). Filtered from COCO [11] and VG datasets [9], the PSG dataset [20] contains 133 object classes, including things, stuff, and 56 relation classes. This dataset has 46k training images and 2k testing images with panoptic segmentation and scene graph annotation. We follow the same data-processing pipelines from [20].

Action Genome (AG). AG [8] annotates 234,253 frame scene graphs for sampled frames from around 10K videos, based on the Charades dataset [15]. The annotations cover 35 object categories and 25 predicates. The overall predicates consist of three types of predicates: attention, spatial, and contracting.

DIML. DIML [2] comprises 2M color images and their corresponding depth maps from a great variety of natural indoor and outdoor scenes. The indoor dataset was constructed using the Microsoft Kinect v2, while the outdoor dataset was built using the stereo cameras (ZED stereo camera and built-in stereo camera). We randomly select the 200K samples for training.

G. Detailed Experimental Implementations

We employ Imagebind [4] as our 4D scene encoder. Similarly, Imagebind applies the image, video decoder, and depth encoder when performing the 2D-to-4D transfer learning. The design of the aggregator utilized for fusing RGB and depth features follows in [1]. The projector is implemented as a 2-layer MLP. The LLM is instantiated with LLaMA2 [18] and fine-tuned using LoRA [5]. We initialize the mask decoder with SAM2 [14] weights. The Depth Estimator consists of a 3×3 convolutional layer and a projector implemented as a 1×1 convolution layer to predict depth features. Both the RGB Temporal Estimator and the Depth Temporal Estimator use 6 transformer layers, with a 512-dimensional embedding dimension, and 8 attention heads. The optimizer is AdamW, with an inverse square root learning rate schedule and warm-up steps. The training is carried out end-to-end on 8 H100 80GB GPUs with distributed training based on DeepSpeed. We summarize the training recipes for 4D-LLM in Tab. 1.

H. Additional Experiments

Analyzing 4D-LLM. To comprehensively understand the strengths and motivations behind 4D-LLM, it is essential to analyze its design and advantages.

First, the decision to utilize LLMs stems from their inherent richness in knowledge and emergent capabilities. LLMs excel in handling diverse textual tasks due to their extensive pretraining, which equips them with a robust internal

Configuration	Step-1	Step-2			Step-3	Step-4
		Subprocess-a	Subprocess-b	Subprocess-c		
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Precision	bfloat16	bfloat16	bfloat16	bfloat16	bfloat16	bfloat16
Peak learning rate of LLM	5e-5	-	-	-	5e-5	5e-5
Peak learning rate of Visual Part	5e-4	2e-3	5e-3	2e-4	2e-4	5e-4
Weight Decay	0.05	0.1	0.1	0.1	0.05	0.05
Learning Rate Scheduler	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine
LR Warmup Steps	500	500	500	500	500	500
Training Data	PSG4D [21]	DIML [2]	AG [8]	PSG4D [21]	PSG4D [21]	VG [9], PSG [20]

Table 1. Training recipes for 4D-LLM.

Method	R@50	R@100	mR@50	mR@100
• Supervised learning				
Motifs [22]	**28.9**	**33.1**	6.4	7.7
Motifs + CFA [10]	-	-	**11.6**	**13.2**
VCTree [17]	28.3	31.9	6.5	7.4
VETO [16]	26.1	29.0	7.0	8.1
• Zero-shot setting				
4D-LLM (ours)	28.0	32.3	10.9	13.1

Table 2. Zero-shot 2D image SG generation performance on GQA [7] dataset.

knowledge base. By leveraging this knowledge, 4D-LLM is designed to achieve fine-grained perception across various scenes. This aligns with our goal of fully utilizing the internal capabilities of LLMs to address the challenges of 4D-PSG generation, ensuring precise understanding and representation of complex environments.

Another key motivation for using an LLM-based approach is its potential to seamlessly integrate with downstream tasks. The actual utility of 4D-PSG lies in its ability to serve broader applications, such as robotic navigation and role-playing simulations. LLM-based models are particularly advantageous in these scenarios due to their adaptability and capacity for rapid transfer to new tasks. This flexibility ensures that 4D-LLM can maximize its impact beyond PSG generation, making it an ideal candidate for practical applications in dynamic, multimodal environments.

Experimentally, we have demonstrated that the LLM-based 4D-PSG generation method achieves substantial performance improvements over baseline models. These gains validate the efficacy of incorporating LLMs into 4D scene understanding tasks. Furthermore, by integrating our innovative 2D-to-4D visual scene transfer learning approach, we observed additional performance enhancements. This highlights the synergy between LLM-based architectures and our novel methods, underscoring the effectiveness of 4D-LLM in advancing 4D scene graph generation and its broader applicability.

The Generalization Capability. We further evaluate the generalization capability of the proposed 4D-LLM by testing its performance on SG generation tasks for 2D images and

Method	PSG4D-GTA		PSG4D-HOI	
	R/mR@20	R/mR@50	R/mR@20	R/mR@50
PSG4DFormer$^{\text{one-stage}}$ + $V^{2\to4}$-VST	6.47 / 3.56	6.85 / 3.01	5.42 / 3.78	5.86 / 3.45
4D-LLM w/o PSG4D	8.45 / 5.03	9.09 / 5.32	6.45 / 4.01	8.45 / 6.79

Table 3. Zero-shot analysis: The PSG4D dataset is excluded from training, with only 4D data used for 2D-to-4D scene transcending.

Method	With Constraint		No Constraint	
	R@20	R@50	R@20	R@50
• Supervised learning				
VCTree [17]	32.6	34.7	35.3	46.8
GPS-Net [12]	33.1	35.1	35.7	47.3
STTran [3]	34.1	**37.0**	36.2	**48.8**
• Zero-shot setting				
4D-LLM (ours)	**34.8**	36.1	**40.9**	48.3

Table 4. Zero-shot 2D video SG generation performance on AG [8] dataset.

videos. Specifically, we conduct zero-shot inference using 4D-LLM on the GQA [7] and AG [8] datasets, with results presented in Tab. 2 and 4. Compared to fully supervised baselines, our model achieves comparable performance in 2D image and video SG generation tasks. This demonstrates the strong generalization ability of 4D-LLM, which not only excels in 4D-PSG generation but also performs effectively across other modalities. These findings further highlight the model's versatility in handling various SG generation tasks, making it applicable to various scenarios, such as robot navigation and other downstream tasks.

Detailed Example of Chained Inference Results. Here, we show two detailed examples of chained inference results.

Detailed results of the chained inference process for the example shown in Fig. 5

Input Data: 4D Scene  , the duration

Inference stage 1: Object Description and Categorization.
Object 1:
Description: A pair of hand tools with metal jaws and a

handle wrapped in a white material. It has a pivot point that allows it to grip or manipulate objects.
Category: Pliers

Object 2:
Description: A rectangular, flat surface with a smooth, glossy finish, decorated with an image of two reindeer and text. Likely a piece of furniture used to hold items.
Category: Table

Object 3:
Description: A hand with a visible wrist, reaching toward the pliers, likely indicating an interaction with them.
Category: Hand

Object 4:
Description: A soft cushion with a yellow fabric cover, lying on a blue quilted surface with floral patterns.
Category: Pillow

Object 5:
Description: A large, cushioned seating area with a fabric cover in a neutral color, placed near the table and likely supporting the pillow. *Category*: Sofa

Inference stage 2: Semantic Relation Identification.
(Hand, Pliers) - The hand is reaching toward or interacting with the pliers, suggesting a possible action or manipulation.
(Pliers, Table) - The pliers are resting on the table surface, indicating a positional relationship.
(Pillow, Table) - The pillow is lying on the table or adjacent furniture, indicating a nearby or resting position.
(Sofa, Table) - The sofa is positioned near the table, indicating a spatial relationship.

Inference stage 3: Precise Relation Description
(Hand, reaching toward, Pliers)
(Hand, picking up, Pliers)
(Hand, holding, Pliers)
(Pliers, resting on, Table)
(Pillow, lying near, Table)
(Sofa, positioned near, Table)

Inference stage 4: Temporal Span Determination
(Hand, reaching toward, Pliers, start time: 0.0, end time: 0.2)(Hand, picking up, Pliers, start time: 0.2, end time: 0.55)(Hand, holding, Pliers, start time: 0.5, end time: 1.0)
(Pliers, resting on, Table, start time: 0.0, end time: 0.4)
(Pillow, lying near, Table, start time: 0.0, end time: 1.0)
(Sofa, positioned near, Table, start time: 0.0, end time: 1.0)

Final Output Format:
(Hand, reaching toward, Pliers, 0.0, 0.2) (Hand, picking up, Pliers, 0.2, 0.55) (Hand, holding, Pliers, 0.5, 1.0) (Pliers, resting on, Table, 0.0, 0.4) (Pillow, lying near, Table, 0.0, 1.0) (Sofa, positioned near, Table, 0.0, 1.0)

Detailed results of the chained inference process for the example shown in Fig. 6

Input Data: 4D Scene , the duration

Inference stage 1: Object Description and Categorization.
Object 1:
Description: A person with short black hair, wearing a blue shirt, yellow pants, and gloves, standing on or near a railroad track, facing another person.
Category: Person

Object 2:
Description: A second person with short hair, wearing a dark jacket, bending forward near the railroad track, partially obscured in the foreground.
Category: Person

Object 3:
Description: A set of metal railroad tracks, extending through the scene, with parallel rails and wooden ties.
Category: Railroad Track

Object 4:
Description: A low concrete barrier running alongside the railroad track, separating the path from the track area.
Category: Road Barrier
Object 5:
Description: An industrial structure in the background, with metal and concrete components, possibly part of a factory or warehouse.
Category: Industrial Building

Object 6:
Description: There are irregularly shaped stones covering the ground near the railway tracks.
Category: gravel

Inference stage 2: Semantic Relation Identification.
(Person 1, Person 2) - Person 1 and Person 2 are positioned close to each other, suggesting a possible interaction or confrontation.
(Person 1, Railroad Track) - Person 1 is standing near or on the railroad track, indicating a spatial relationship.
(Person 2, Railroad Track) - Person 2 is positioned close to the railroad track, indicating proximity.
(Person 1, Road Barrier) - The person is positioned near the road barrier, suggesting a spatial relationship.
(Person 1, gravel) - The person is positioned on the gravel, suggesting a spatial relationship.
(Railroad Track, Industrial Building) - The railroad track leads towards or is located near the industrial building, indicating spatial context.

Inference stage 3: Precise Relation Description
(Person 1, in front of, Person 2)
(Person 1, talking to, Person 2)
(Person 1, walking on, Gravel)
(Person 1, behind, Road Barrier)
(Person 1, walking alongside, Railroad Track)
(Person 2, walking alongside, Railroad Track)
(Railroad Track, near, Industrial Building)

Inference stage 4: Temporal Span Determination
(Person 1, in front of, Person 2, start time: 0.8, end time: 1.0)
(Person 1, talking to, Person 2, start time: 0.8, end time: 1.0)
(Person 1, walking on, Gravel, start time: 0, end time: 0.2)
(Person 1, behind, Road Barrier, start time: 0, end time: 0.2)
(Person 1, walking alongside, Railroad Track, start time: 0.2, end time: 0.8)
(Person 2, walking alongside, Railroad Track, start time: 0.8, end time: 1.0)
(Railroad Track, near, Industrial Building, start time: 0.1, end time: 0.6)

Final Output Format:
(Person 1, in front of, Person 2, 0.8, 1.0),(Person 1, talking to, Person 2, 0.8, 1.0), (Person 1, walking on, Gravel, 0,

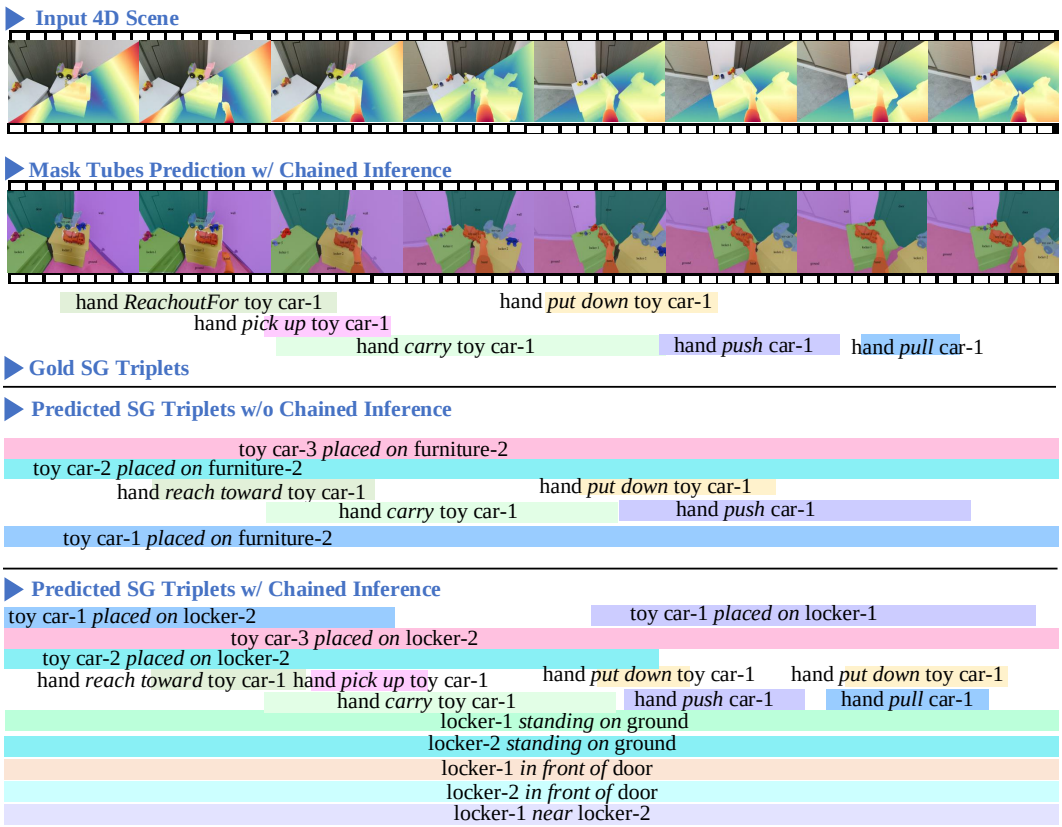

Figure 5. A case illustrating the prediction of 4D-LLM on 4D-PSG

0.2), (Person 1, behind, Road Barrier, 0, 0.2), (Person 1, walking alongside, Railroad Track, 0.2, 0.8), (Person 2, walking alongside, Railroad Track, 0.8, 1.0), (Railroad Track, near, Industrial Building, 0.1, 0.6)

More Visualizations. Fig. 5 provides an additional example of predictions generated by 4D-LLM, illustrating its ability to identify both semantic actions and fine-grained spatial relationships within the scene. For instance, the model successfully captures interactions such as "toy-car *placed on* furniture-2", "locker-1 *standing on* ground". NNotably, with the integration of the chained inference mechanism, 4D-LLM has significantly enhanced its ability to distinguish finer object details, providing more robust performance in differentiating "furniture" from "locker". Additionally, the chained inference mechanism enhances the model's accuracy in recognizing key semantic relationships over extended timeframes, further improving its performance in scenarios involving long-duration activities. These results emphasize the robust capability of 4D-LLM for a detailed and precise understanding of complex 4D environments, validating its effectiveness in generating high-quality 4D-PSGs.

References

[1] Xiaokang Chen, Kwan-Yee Lin, Jingbo Wang, Wayne Wu, Chen Qian, Hongsheng Li, and Gang Zeng. Bi-directional cross-modality feature propagation with separation-and-aggregation gate for RGB-D semantic segmentation. In *ECCV*, 2020. 4

[2] Jaehoon Cho, Dongbo Min, Youngjung Kim, and Kwanghoon Sohn. DIML/CVL RGB-D dataset: 2m RGB-D images of natural indoor and outdoor scenes. *CoRR*, abs/2110.11590, 2021. 2, 4, 5

[3] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *ICCV*, 2021. 5

[4] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind one embedding space to bind them all. In *CVPR*, 2023. 4

[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 4

[6] Yuan-Ting Hu, Jiahong Wang, Raymond A. Yeh, and Alexander G. Schwing. SAIL-VOS 3d: A synthetic dataset and baselines for object detection and 3d mesh reconstruction from video data. In *CVPR*, 2021. 4

[7] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019. 5

[8] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *CVPR*, 2020. 4, 5

[9] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 2017. 3, 4, 5

[10] Lin Li, Guikun Chen, Jun Xiao, Yi Yang, Chunping Wang, and Long Chen. Compositional feature augmentation for unbiased scene graph generation. In *ICCV*, 2023. 5

[11] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 4

[12] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *CVPR*, 2020. 5

[13] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4d egocentric dataset for category-level human-object interaction. In *CVPR*, 2022. 4

[14] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross B. Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *CoRR*, abs/2408.00714, 2024. 4

[15] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 4

[16] Gopika Sudhakaran, Devendra Singh Dhami, Kristian Kersting, and Stefan Roth. Vision relation transformer for unbiased scene graph generation. In *ICCV*, 2023. 5

[17] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019. 5

[18] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. 4

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[20] Jingkang Yang, Yi Zhe Ang, Zujin Guo, Kaiyang Zhou, Wayne Zhang, and Ziwei Liu. Panoptic scene graph generation. In *ECCV*, 2022. 3, 4, 5

[21] Jingkang Yang, Jun Cen, Wenxuan Peng, Shuai Liu, Fangzhou Hong, Xiangtai Li, Kaiyang Zhou, Qifeng Chen, and Ziwei Liu. 4d panoptic scene graph generation. *NeurIPS*, 2023. 4, 5

[22] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 5