Towards World Scene Graph Generation from Monocular Videos: A Structured World Representation for Embodied Agents

Rohith Peddi[1] Saurabh[2] Shravan Shanmugam[1]
Likhitha Pallapothula[1] Yu Xiang[1] Parag Singla[2] Vibhav Gogate[1]

[1]The University of Texas at Dallas [2]Indian Institute of Technology Delhi

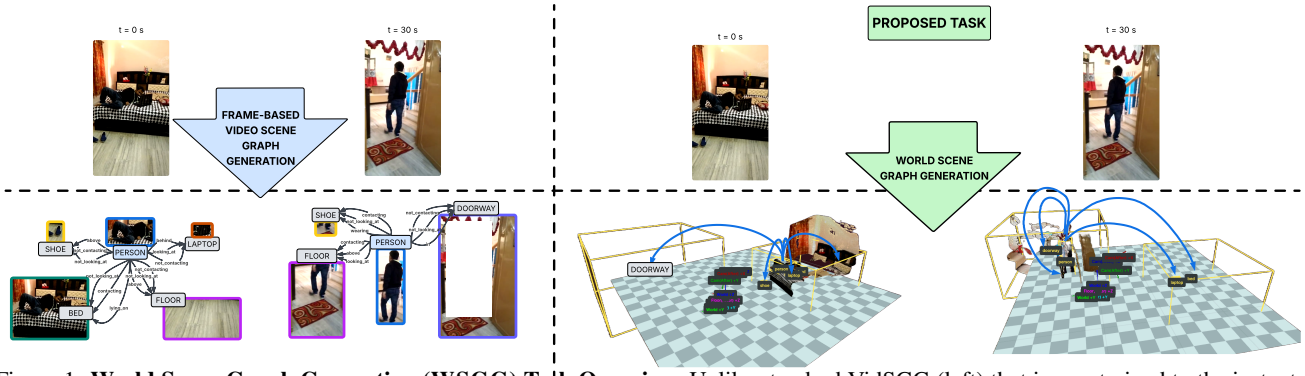

Figure 1. **World Scene Graph Generation (WSGG) Task Overview.** Unlike standard VidSGG (left) that is constrained to the instantaneous camera view and discards objects once they exit the frame or get occluded, our proposed task (right) grounds scene understanding in a global 3D world coordinate frame. WSGG requires a model to output a comprehensive world scene graph at each timestamp containing all interacting objects in the environment. As shown at $t = 30$s, out-of-view objects (e.g., bed, laptop) remain localized in 3D, enabling global, view-independent interpretable scene reasoning. Blue curves drawn from person to objects (right) represent relationships.

Abstract

Spatio-temporal scene graphs provide a principled representation for modeling evolving object interactions, yet existing methods remain fundamentally frame-centric: they operate primarily in 2D, reason only about currently visible objects, and discard entities upon occlusion. To address these limitations, we introduce ***World Scene Graph Generation*** **(WSGG)***, a novel task that constructs a temporally persistent world scene graph at each time step, encompassing all interacting objects both observed and unobserved in the scene. To support this task, we introduce* **ActionGenome4D***, a dataset upgrading ActionGenome videos to 4D scenes, and establish initial baselines by adapting existing Video Scene Graph Generation (VidSGG) methods. We then propose* **WorldWise***, a systematic, object-centric framework that captures joint visual and 3D object representations and treats occlusion as a natural masking signal, reconstructing missing features via camera-pose-conditioned cross-attention over an object's visible history. Furthermore, we formulate Unlocalized* WSGG *(UWSGG) as a challenging spatial intelligence task and present* **UWSGG-GraphRAG***, a simple yet strong baseline framework that leverages Multi-Modal Large Language Models (MLLMs) and Graph-Based Retrieval. Thus* WSGG *advances VidSGG toward world-centric, temporally persistent, 4D reasoning, providing a structured world representation that can serve as the state substrate for active sensing and closed-loop planning.*

1. Introduction

Scene graphs provide a structured, interpretable representation of visual scenes by encoding objects as nodes and their relationships as edges [35]. Despite significant progress in generating scene graphs from images and video [11, 29], the predominant paradigm remains fundamentally *frame-centric*. Current models process frames independently to predict flat, 2D scene graphs that lack both 3D spatial grounding and temporal consistency. Consequently, when an object becomes occluded or exits the camera's field of view, it simply vanishes from the graph. This frame-centric representation (left, Figure 1) fundamentally contrasts with how real-world agents perceive and act. Embodied agents accumulate observations over time, maintaining a persistent semantic memory of their environment and reasoning about the evolving relationships of objects even when they are no longer visible. Building such a persistent world representation is a prerequisite for active sensing deciding *what* and *when* to observe, and for closed-loop planning that continuously integrates new percepts with remembered state.

Developmental psychology has long recognized *object permanence* [77] as a foundational prerequisite for physical reasoning: the understanding that objects continue to exist when they leave our direct perception. Achieving such world-centric scene understanding (right, Figure 1) from monocular video requires three capabilities that existing datasets do not jointly provide: (i) 3D spatial grounding of all objects in a shared world coordinate frame, (ii) tempo-

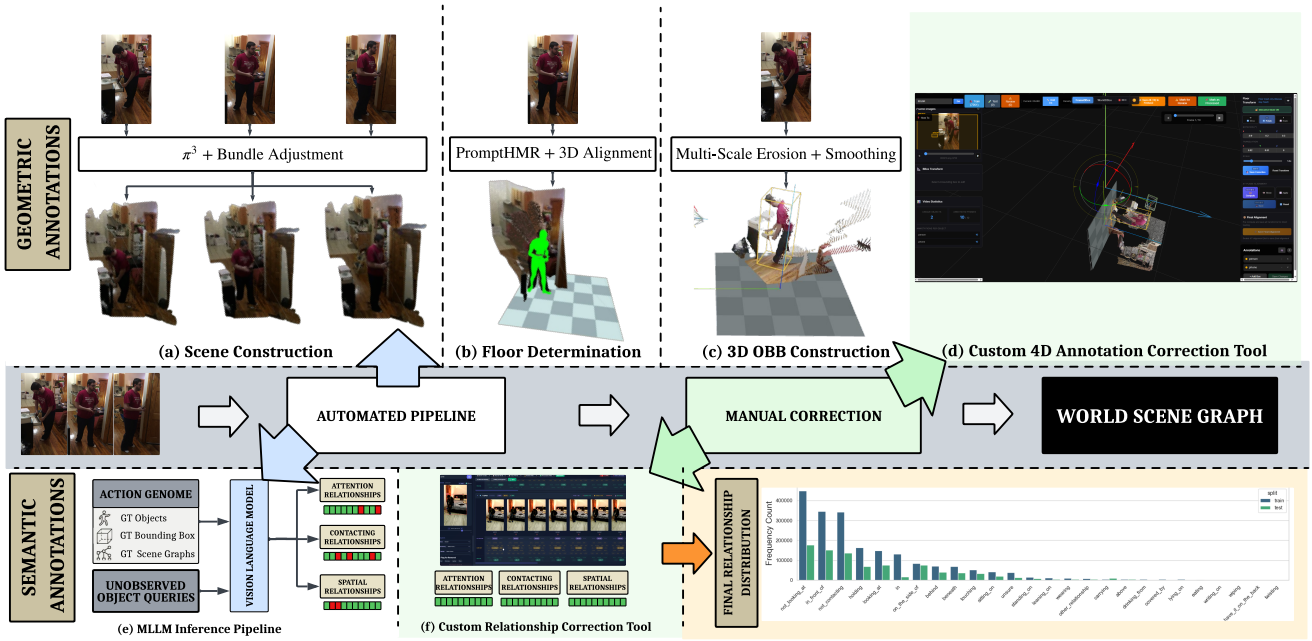

Figure 2. **Hybrid 3D Reconstruction and Annotation.** A human-in-the-loop pipeline with parallel geometric (top) and semantic (bottom) streams. **Geometric:** **(a)** π^3 reconstructs dense 3D point clouds; **(b)** PromptHMR anchors a canonical ground plane; **(c)** multi-scale erosion fits 3D OBBs. **Semantic:** **(e)** An MLLM predicts attention, spatial, and contacting predicates from 2D AG priors. **Manual:** **(d)** A custom 4D tool corrects OBB trajectories; **(f)** a relationship tool verifies predicates. (bottom right: long-tailed predicate distribution).

Table 1. Comparison of different SGG Tasks

Task	Input	Spatial Dim.	Temporal	Localization	Object Scope	Coord. Frame	Unobs. Objects	Rel. Persistence
Image SGG	Image	2D	✗	2D BBox	Detected	Image	✗	Per-frame
Video SGG	Video	2D	✓	2D BBox	Detected / frame	Image	✗	Cross-frame
3D SGG	3D Scan	3D	✗	3D BBox	All in scan	Scene	✗	Per-scan
4D SGG	Video + 3D	3D	✓	3D BBox	Detected / frame	Scene	✗	Cross-frame
Panoptic SGG	Image	2D	✗	2D Mask	Things + Stuff	Image	✗	Per-frame
Panoptic VSGG	Video	2D	✓	2D Mask	Things + Stuff	Image	✗	Cross-frame
WSGG (ours)	**Video**	**3D**	✓	**3D BBox**	**World state**	**World**	✓	**Through occlusion**

rally consistent object identity and tracking across frames, and (iii) dense semantic annotations between all interacting objects, including unobserved objects that are present in the scene but not visible in a given frame.

We aim to bridge this gap with two main contributions: a new task and a new dataset. We formalize the task of **WSGG**, which generalizes conventional VidSGG from frame-centric graphs to temporally persistent, world-anchored scene graphs that account for all interacting objects in the world state. WSGG requires a model to (a) localize all objects via 3D oriented bounding boxes in a shared world frame and (b) predict all pairwise relationships between interacting objects, including those that remain unobserved. We first adapt existing VidSGG methods (STTran [11], DsgDetr [8]) to the WSGG setting by augmenting them with a persistent memory buffer and 3D geometric scaffolding. We then propose **WorldWise**, a novel framework built on the *Masked Auto-Encoder* (MAE) paradigm: it treats occlusions and camera motion as natural masking events and reframes unseen-object reasoning as a structured completion problem. To this end, we introduce **ActionGenome4D**, upgrading ActionGenome [29] into a *4D spatio-temporal scene representation*. We further evaluate open-source MLLMs on unlocalized WSGG, and present **UWSGG-GraphRAG**, a graph-based retrieval-augmented generation framework for unlocalized WSGG[1].

2. Related work

Scene graph generation has branched into several task variants[2]. Table 1 contrasts these formulations along eight axes[3]. Scene graphs were first introduced as structured representations that encode objects and their pairwise relationships for static images [8, 35, 107]. Extending scene graphs to video [11, 18, 29, 59, 65, 71]; 3D [17, 32, 33, 53, 62, 99]; 4D scene graphs extend the 3D domain to the temporal dimension, reasoning about object-centric dynamics [44, 91].

3. Notation & Problem Description

Given an input video $V_1^T = \{I^t\}_{t=1}^T$ of T monocular frames, the *world state* $\mathcal{W}^t$ at timestamp t is the complete set of objects in the scene, partitioned as $\mathcal{W}^t = \mathcal{O}^t \cup \mathcal{U}^t$, $\mathcal{O}^t \cap \mathcal{U}^t = \emptyset$, where $\mathcal{O}^t = \{w_k^t\}_{k=1}^{N(t)}$ are *observed* objects visible in I^t and $\mathcal{U}^t = \{w_k^t\}_{k=N(t)+1}^{M(t)}$ are *unobserved* objects occluded or out of view. All objects persist across timestamps. A binary indicator $\text{vis}(k,t) \in \{0,1\}$ denotes whether $w_k^t \in \mathcal{O}^t$, and $\mathbf{T}^t \in \text{SE}(3)$ gives the camera pose. **Objects.** Each $w_k^t \in \mathcal{W}^t$ has category $c_k^t \in \mathcal{C}$ and a *3D oriented bounding box* $\mathbf{b}_k^t \in \mathbb{R}^{8\times 3}$ (eight corners in world frame). Observed objects additionally carry

[1]Code and data will be made publicly available.
[2]Refer Appendix for extended survey.
[3]Object Scope: *Things*: countable objects; *Stuff*: amorphous regions (e.g., wall, floor). 4D SGG methods require RGB-D or multi-view inputs.

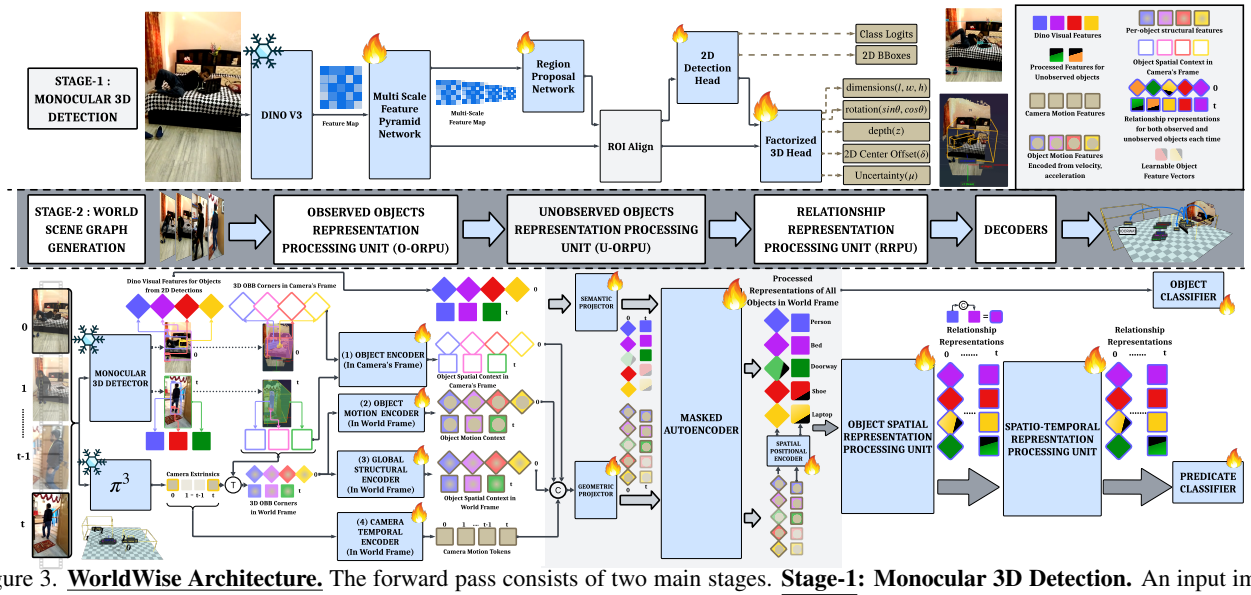

Figure 3. **WorldWise Architecture.** The forward pass consists of two main stages. **Stage-1: Monocular 3D Detection.** An input image is passed through a frozen DINO V3 backbone to extract visual features, which are refined by a Multi-Scale Feature Pyramid Network. A Region Proposal Network (RPN) and RoI Align extract object proposals, which are routed to parallel heads: a 2D Detection Head for class logits and 2D bounding boxes, and a Factorized 3D Head predicting 3D dimensions (l, w, h), rotation $(\sin\theta, \cos\theta)$, depth (z), 2D center offset (δ), and uncertainty (μ). **Stage-2: World Scene Graph Generation.** First, the **Observed Objects Representation Processing Unit (O-ORPU)** encodes DINO visual features and 3D bounding box corners. An Object Encoder processes spatial context in the camera frame, while camera extrinsics (π^3) transform 3D corners into the world frame. These transformed coordinates are processed by an Object Motion Encoder, Global Structural Encoder, and Camera Temporal Encoder to extract kinematic, global layout, and ego-motion contexts. Next, the **Unobserved Objects Representation Processing Unit (U-ORPU)** aligns these features via Semantic and Geometric Projectors. To infer object permanence under occlusion, projected features are fused with learnable object feature vectors and processed by a Masked Autoencoder with spatial positional encoding, yielding unified representations for all entities. Finally, the **Relationship Representation Processing Unit (RRPU)** refines these dense node features using an Object Spatial Representation Processing Unit for per-frame structural interactions, followed by a Spatio-Temporal Representation Processing Unit to aggregate temporal dynamics. **Decoders** (object and predicate classifiers), output the final nodes and edges of the world scene graph.

a *2D bounding box* $\mathbf{d}_k^t \in \mathbb{R}^4$. **Relationships.** Each object pair (w_i^t, w_j^t) may exhibit multiple predicates $\{p_{ijk}^t\}_k$ with $p_{ijk}^t \in \mathcal{P}$, defining relationship instances $r_{ijk}^t = (w_i^t, p_{ijk}^t, w_j^t)$. **Scene Graphs.** The *frame-level scene graph* $\mathcal{G}^t = \{r_{ijk}^t \mid w_i^t, w_j^t \in \mathcal{O}^t\}$ covers observed objects only, while the *world scene graph* $\mathcal{G}_{\mathcal{W}}^t = \{r_{ijk}^t \mid w_i^t, w_j^t \in \mathcal{W}^t\}$ spans all interacting objects. Each observed object has a visual feature $\mathbf{f}_k^t \in \mathbb{R}^{d_{\text{roi}}}$. **Definition.** (1) **VidSGG** builds frame-level graphs $\{\mathcal{G}^t\}_{t=1}^T$: detecting observed objects and predicting their pair-wise relationships. (2) **WSGG** constructs $\mathcal{G}_{\mathcal{W}}^t$ at each t, requiring: (a) estimating 3D OBBs $\mathbf{b}_k^t$ for all $w_k^t \in \mathcal{W}^t$, and (b) predicting all pair-wise relationships for all objects in the scene, whether visible or not.

4. ActionGenome4D Dataset

The annotation pipeline (Figure 2) employs parallel geometric and semantic streams with human-in-the-loop refinement. **(a) 3D Scene Construction.** We reconstruct per-timestamp 3D scenes from egocentric AG [29] videos using the π^3 [87] feed-forward model, which jointly estimates per-pixel 3D points, confidence scores, and camera-to-world SE(3) poses $\{\mathbf{T}^t\}$ in a single pass. An optical-flow-guided adaptive sampler selects keyframes capturing significant motion. Residual pose drift is corrected via iterative bundle adjustment that jointly refines poses and 3D points by minimising reprojection error. **(b,c) Geometric Annotation.** We produce world-frame OBBs $\mathbf{b}_k^t \in \mathbb{R}^{8\times3}$ for every object through: GDINO [46] detection fused with GT annotations, SAM2 [67] segmentation, PromptHMR [86]-based floor determination (RANSAC alignment of SMPL meshes recovers metric scale and ground-plane orientation), and PCA-based OBB fitting with multi-scale erosion (kernels $\{0, 3, 5, 7, 10\}$ px) followed by Kalman + RTS smoothing for temporal consistency. **(e) Semantic Annotation.** AG4D extends AG with dense relationship labels for *all* objects in $\mathcal{W}^t$ including unobserved ones. The predicate set $\mathcal{P}$ spans three axes: **attention** (3 labels, single), **spatial** (6, multi), and **contacting** (17, multi). An MLLM bootstraps annotations from 2D AG priors and object queries (Section L). **(d,f) Manual Correction.** Annotators use a custom 3D viewer to adjust OBB positions, rotations, and trajectories **(d)**, and a relationship correction tool to verify and fix predicted predicates **(f)**. **Quality.** The automated pipeline achieves mean IoU_{3D} of 0.385; human refinement raises

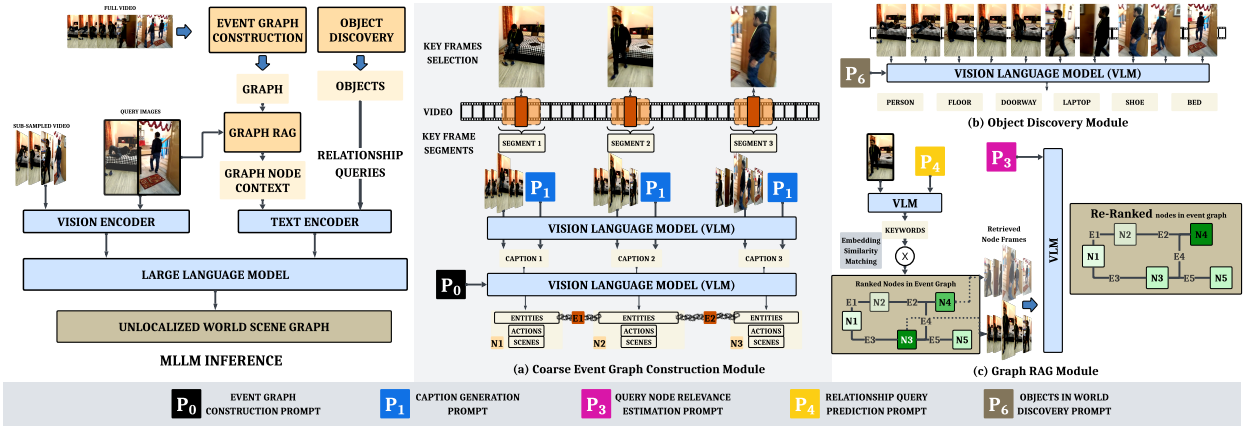

Figure 4. **Unlocalized WSGG via Graph RAG. (a)** A VLM extracts entities/actions/scenes per clip and builds a directed knowledge graph $\mathcal{G}_V$ via BGE [6] cosine similarity. **(b)** Object discovery yields the entity vocabulary $\mathcal{O}_V$. **(c)** Per-object BGE retrieval + yes/no VLM verification filters graph nodes ($K\times$ deduplication). **Inference:** Each (frame, object) prompt concatenates graph context, subtitles, and relationship query with the target clip. More details regarding prompt design and an ablation on MLLMs is available in the Appendix.

inter-annotator 3D IoU to >0.9. The Label Flip Rate (LFR) tracks predicate corrections. Inter-annotator study on 200 videos yields consensus of $\geq 90\%$ for all predicate types.

5. World Scene Graph Generation

We present **WorldWise**, a novel architecture for WSGG built upon the *Masked Auto-Encoder* (MAE) paradigm that *treats occlusions as natural masking events* and reframes unseen-object reasoning as a *structured completion problem*. As shown in Figure 3, WorldWise comprises:

Observed Objects RPU (O-ORPU). Encodes all available signals for visible objects into per-object per-frame tokens. A *GlobalStructuralEncoder* converts world-frame 3D OBB corners (8×3) into structural tokens via MLP projection with max-pooled global context. An *ObjectSpatialEncoder* derives view-dependent features from camera extrinsics $\mathbf{T} = [\mathbf{R}\,|\,\boldsymbol{\tau}]$, computing per-object direction, view alignment, and azimuth features enabling the model to distinguish field-of-view egress from physical occlusion. A *CameraTemporalEncoder* captures long-range ego-motion via relative-pose encoding with self-attention, and an *ObjectMotionEncoder* computes 3D velocity, acceleration, and camera-relative motion from world-frame OBB centers.

Unobserved Objects RPU (U-ORPU). Bridges observed and unobserved objects via two sub-modules. The *ScaffoldTokenizer* performs top-down initialization: every object receives a token at every frame regardless of visibility. Visible objects receive projected visual features; occluded objects receive a learnable $\mathbf{e}_{[\text{MASK}]}$. During training, a fraction p_{mask} of visible objects are artificially masked. All modalities (geometry, visual/mask, camera, motion) are fused via linear projection. The *AssociativeRetriever* restores masked states via per-object bidirectional cross-attention across the full video. Attention is asymmetric: queries span all frames while keys/values are restricted to visible frames. Camera-pose features are injected into Q/K projections (not V) to bias retrieval toward similar viewpoints while keeping retrieved content purely visual. A learned *VisibilityEmbedding* signals whether token was observed or reconstructed.

Table 2. **PredCls results on ActionGenome4D.**

	Recall (R@K)						Mean Recall (mR@K)					
	With Constraint			*No Constraint*			*With Constraint*			*No Constraint*		
Method	R@10	R@20	R@50	R@10	R@20	R@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
W-STTran	63.59	66.53	66.54	79.49	94.06	99.63	33.58	37.47	37.47	49.51	72.75	94.11
W-STTran^{++}	64.28	67.24	67.25	80.04	94.34	99.58	33.73	37.82	37.83	51.52	73.31	94.75
W-DsgDetr	63.27	66.26	66.27	79.73	94.17	99.60	32.70	36.51	36.51	51.38	73.27	94.59
W-DsgDetr^{++}	63.58	66.54	66.55	79.59	93.98	99.56	33.21	37.49	37.49	51.87	73.26	94.39
WorldWise$_{\text{DinoV2b}}$	64.04	66.94	66.95	79.82	94.23	99.65	33.30	37.31	37.32	50.61	72.10	94.08
WorldWise$_{\text{DinoV2L}}$	64.53	67.46	67.47	80.22	94.23	99.68	**33.85**	37.87	37.87	50.46	71.99	94.98
WorldWise$_{\text{DinoV3L}}$	**65.41**	**68.41**	**68.42**	**80.99**	**94.48**	**99.72**	33.41	**38.29**	**38.30**	**52.81**	**74.86**	**95.19**

Table 3. **SGDet results on ActionGenome4D.**

	Recall (R@K)						Mean Recall (mR@K)					
	With Constraint			*No Constraint*			*With Constraint*			*No Constraint*		
Method	R@10	R@20	R@50	R@10	R@20	R@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
W-STTran	40.53	55.34	65.48	25.42	37.20	56.08	6.63	14.36	23.03	15.58	29.07	53.74
W-STTran^{++}	43.02	58.33	69.21	26.60	37.90	56.24	8.46	17.25	27.06	18.73	32.83	55.57
W-DsgDetr	39.58	54.39	64.53	25.90	37.40	55.92	6.83	14.40	22.75	15.03	27.49	52.38
W-DsgDetr^{++}	**43.39**	**58.69**	69.50	26.87	38.20	**56.75**	8.85	18.21	28.60	18.51	32.65	**56.05**
WorldWise$_{\text{DinoV2b}}$	41.34	56.56	67.76	27.60	39.16	56.11	9.55	20.41	30.00	19.96	31.98	53.71
WorldWise$_{\text{DinoV2L}}$	42.05	57.18	69.44	**28.54**	**40.34**	56.69	**10.50**	21.46	33.48	**22.39**	**36.51**	54.18
WorldWise$_{\text{DinoV3L}}$	40.14	57.21	**70.93**	26.90	37.99	52.85	9.54	**21.61**	**35.53**	21.35	34.41	55.68

Relationship RPU (RRPU). Refines unified node features via two stages: an *InterObjectTransformer* applies pairwise 3D spatial positional encodings (inter-object distances, direction vectors, log-volume ratios) over the full world state; then a spatio-temporal module concatenates node features, union ROI features, and CLIP text embeddings, with intra-frame self-attention and cross-frame temporal attention.

An object classifier (2-layer MLP) predicts class logits; three predicate heads output attention (softmax), spatial and contacting (sigmoid) probabilities. A cross-view reconstruction head provides self-supervised signal on artificially masked objects. The loss combines scene-graph prediction ($\mathcal{L}_{\text{SG}}$, split into visible and VLM-pseudo-labeled unseen pairs), reconstruction MSE ($\mathcal{L}_{\text{recon}}$), and a simulation loss ($\mathcal{L}_{\text{sim}}$) that re-predicts relationships for masked objects against clean GT. Tables 2 and 3 report Predicate Classification (PredCls) and Scene Graph Detection (SGDet) results.

References

[1] Elena Agliari and Giordano De Marzo. Tolerance versus synaptic noise in dense associative memories, 2020. 12

[2] Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models, 2025. 12

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields, 2017. 17

[4] Hamza Tahir Chaudhry, Jacob A. Zavatone-Veth, Dmitry Krotov, and Cengiz Pehlevan. Long sequence hopfield memory, 2023. 12

[5] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities, 2024. 12

[6] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multilinguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2025. 4

[7] Pingyi Chen, Yujing Lou, Shen Cao, Jinhui Guo, Lubin Fan, Yue Wu, Lin Yang, Lizhuang Ma, and Jieping Ye. Sd-vlm: Spatial measuring and understanding with depth-encoded vision-language models, 2025. 12

[8] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation, 2019. 2, 11

[9] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision language models, 2024. 12

[10] David G. Clark. Transient dynamics of associative memory models, 2025. 12

[11] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation, 2021. 1, 2, 10, 11

[12] Yuren Cong, Jinhui Yi, Bodo Rosenhahn, and Michael Ying Yang. Ssgvs: Semantic scene graph-to-video synthesis, 2022. 12

[13] Helisa Dhamo, Azade Farshad, Iro Laina, Nassir Navab, Gregory D. Hager, Federico Tombari, and Christian Rupprecht. Semantic image manipulation using scene graphs, 2020. 12

[14] Kaize Ding, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu. Graph prototypical networks for few-shot learning on attributed networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, page 295–304, New York, NY, USA, 2020. Association for Computing Machinery. 12

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 25

[16] Azade Farshad, Sabrina Musatian, Helisa Dhamo, and Nassir Navab. Migs: Meta image generation from scene graphs, 2021. 12

[17] Mingtao Feng, Haoran Hou, Liang Zhang, Ziiie Wu, Yulan Guo, and Ajmal Mian. 3d spatial multimodal knowledge accumulation for scene graph prediction in point cloud. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9182–9191, 2023. 2, 12

[18] Shengyu Feng, Subarna Tripathi, Hesham Mostafa, Marcel Nassar, and Somdeb Majumdar. Exploiting long-term dependencies for generating dynamic scene graphs, 2022. 2, 11

[19] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981. 13

[20] Andreas Fürst, Elisabeth Rumetshofer, Johannes Lehner, Viet Tran, Fei Tang, Hubert Ramsauer, David Kreil, Michael Kopp, Günter Klambauer, Angela Bitto-Nemling, and Sepp Hochreiter. Cloob: Modern hopfield networks with infoloob outperform clip, 2022. 12

[21] Shijie Geng, Peng Gao, Moitreya Chatterjee, Chiori Hori, Jonathan Le Roux, Yongfeng Zhang, Hongsheng Li, and Anoop Cherian. Dynamic graph representation learning for video dialog via multi-modal shuffled transformers, 2021. 12

[22] Zexue He, Leonid Karlinsky, Donghyun Kim, Julian McAuley, Dmitry Krotov, and Rogerio Feris. Camelot: Towards large language models with training-free consolidated associative memory, 2024. 12

[23] Benjamin Hoover, Duen Horng Chau, Hendrik Strobelt, Parikshit Ram, and Dmitry Krotov. Dense associative memory through the lens of random features, 2024. 12

[24] Benjamin Hoover, Hendrik Strobelt, Dmitry Krotov, Judy Hoffman, Zsolt Kira, and Duen Horng Chau. Memory in plain sight: Surveying the uncanny resemblances of associative memories and diffusion models, 2024. 12

[25] Hao-Yu Hou, Chun-Yi Lee, Motoharu Sonogashira, and Yasutomo Kawanishi. Fross: Faster-than-real-time online 3d semantic scene graph generation from rgb-d images, 2025. 12

[26] Jerry Yao-Chieh Hu, Dennis Wu, and Han Liu. Provably optimal memory capacity for modern hopfield models: Transformer-compatible dense associative memories as spherical codes, 2024. 12

[27] Jiani Huang, Amish Sethi, Matthew Kuo, Mayank Keoliya, Neelay Velingker, JungHo Jung, Ser-Nam Lim, Ziyang Li, and Mayur Naik. Esca: Contextualizing embodied agents via scene-graph generation, 2025. 12

[28] Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, and Seunghyun Park. Egtr: Extracting graph from transformer for scene graph generation, 2024. 11

[29] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as composition of spatio-temporal scene graphs, 2019. 1, 2, 3, 10, 11, 13, 16, 18, 23

[30] Kibum Kim, Kanghoon Yoon, Jaehyeong Jeon, Yeonjun In, Jinyoung Moon, Donghyun Kim, and Chanyoung Park.

Llm4sgg: Large language models for weakly supervised scene graph generation, 2024. 11

[31] Kibum Kim, Kanghoon Yoon, Yeonjun In, Jaehyeong Jeon, Jinyoung Moon, Donghyun Kim, and Chanyoung Park. Weakly supervised video scene graph generation via natural language supervision, 2025. 11

[32] Ue-Hwan Kim, Jin-Man Park, Taek-jin Song, and Jong-Hwan Kim. 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE Transactions on Cybernetics*, 50(12): 4921–4933, 2020. 2, 10, 11

[33] Sebastian Koch, Pedro Hermosilla, Narunas Vaskevicius, Mirco Colosi, and Timo Ropinski. Sgrec3d: Self-supervised 3d scene graph learning via object-level scene reconstruction, 2023. 2, 10, 12

[34] Leo Kozachkov, Jean-Jacques Slotine, and Dmitry Krotov. Neuron–astrocyte associative memory. *Proceedings of the National Academy of Sciences*, 122(21):e2417788122, 2025. 12

[35] 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 Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016. 1, 2, 10, 11

[36] Andrey Kurenkov, Michael Lingelbach, Tanmay Agarwal, Emily Jin, Chengshu Li, Ruohan Zhang, Li Fei-Fei, Jiajun Wu, Silvio Savarese, and Roberto Martín-Martín. Modeling dynamic environments with scene graph memory, 2023. 12

[37] Huy Le, Nhat Chung, Tung Kieu, Jingkang Yang, and Ngan Le. Uno: Unifying one-stage video scene graph generation via object-centric visual representation learning, 2026. 11

[38] Phillip Y. Lee, Jihyeon Je, Chanho Park, Mikaela Angelina Uy, Leonidas Guibas, and Minhyuk Sung. Perspective-aware reasoning in vision-language models via mental imagery simulation, 2025. 12

[39] Lin Li, Jun Xiao, Guikun Chen, Jian Shao, Yueting Zhuang, and Long Chen. Zero-shot visual relation detection via composite visual cues from large language models, 2023. 11

[40] Pengteng Li, Pinhao Song, Wuyang Li, Weiyu Guo, Huizai Yao, Yijie Xu, Dugang Liu, and Hui Xiong. See&trek: Training-free spatial prompting for multimodal large language model, 2025. 12

[41] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer, 2022. 11

[42] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection, 2022. 25

[43] Yun Li, Yiming Zhang, Tao Lin, Xiangrui Liu, Wenxiao Cai, Zheng Liu, and Bo Zhao. Sti-bench: Are mllms ready for precise spatial-temporal world understanding?, 2025. 12

[44] Haozhe Lin, Zequn Chen, Jinzhi Zhang, Bing Bai, Yu Wang, Ruqi Huang, and Lu Fang. Realgraph: A multiview dataset for 4d real-world context graph generation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3735–3745, 2023. 2, 12

[45] Xin Lin, Chong Shi, Yibing Zhan, Zuopeng Yang, Yaqi Wu, and Dacheng Tao. TD^2-Net: Toward denoising and debiasing for dynamic scene graph generation, 2024. 11

[46] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024. 3, 16

[47] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 17

[48] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 26

[49] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 13

[50] Carlo Lucibello and Marc Mézard. Exponential capacity of dense associative memories. *Physical Review Letters*, 132 (7), 2024. 12

[51] Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso de Melo, Jianwen Xie, and Alan Yuille. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning, 2025. 12

[52] Wufei Ma, Luoxin Ye, Celso M de Melo, Jieneng Chen, and Alan Yuille. Spatialllm: A compound 3d-informed design towards spatially-intelligent large multimodal models, 2025. 12

[53] Yanni Ma, Hao Liu, Yun Pei, and Yulan Guo. Heterogeneous graph learning for scene graph prediction in 3d point clouds. In *Computer Vision – ECCV 2024*, pages 274–291, Cham, 2025. Springer Nature Switzerland. 2

[54] Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. Spatiallm: Training large language models for structured indoor modeling, 2025. 12

[55] Xingyu Miao, Haoran Duan, Quanhao Qian, Jiuniu Wang, Yang Long, Ling Shao, Deli Zhao, Ran Xu, and Gongjie Zhang. Towards scalable spatial intelligence via 2d-to-3d data lifting, 2025. 12

[56] Toki Migimatsu and Jeannette Bohg. Grounding predicates through actions, 2022. 12

[57] Yukuan Min, Aming Wu, and Cheng Deng. Environment-invariant curriculum relation learning for fine-grained scene graph generation, 2023. 11

[58] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 13

[59] Sayak Nag, Kyle Min, Subarna Tripathi, and Amit K. Roy Chowdhury. Unbiased scene graph generation in videos, 2023. 2, 10, 11

[60] Trong-Thuan Nguyen, Pha Nguyen, and Khoa Luu. Hig: Hierarchical interlacement graph approach to scene graph generation in video understanding, 2024. 12

[61] Trong-Thuan Nguyen, Pha Nguyen, Jackson Cothren, Alper Yilmaz, and Khoa Luu. Hyperglm: Hypergraph for video scene graph generation and anticipation, 2025. 11

[62] Jon Nyffeler, Federico Tombari, and Daniel Barath. Hierarchical 3d scene graphs construction outdoors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 26817–26826, 2025. 2, 12

[63] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 14, 25

[64] Rohith Peddi, Saksham Singh, Saurabh, Parag Singla, and Vibhav Gogate. Towards scene graph anticipation, 2024. 11

[65] Rohith Peddi, Saurabh, Ayush Abhay Shrivastava, Parag Singla, and Vibhav Gogate. Towards unbiased and robust spatio-temporal scene graph generation and anticipation, 2025. 2, 10, 11

[66] Tao Pu, Tianshui Chen, Hefeng Wu, Yongyi Lu, and Liang Lin. Spatial-temporal knowledge-embedded transformer for video scene graph generation, 2023. 11

[67] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. 3, 16, 17

[68] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 25

[69] Ivan Rodin, Antonino Furnari, Kyle Min, Subarna Tripathi, and Giovanni Maria Farinella. Action scene graphs for long-form understanding of egocentric videos, 2023. 11

[70] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 13

[71] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. Video visual relation detection. In *Proceedings of the 25th ACM International Conference on Multimedia*, page 1300–1308, New York, NY, USA, 2017. Association for Computing Machinery. 2, 10, 11

[72] Yifan Shen, Yuanzhe Liu, Jingyuan Zhu, Xu Cao, Xiaofeng Zhang, Yixiao He, Wenming Ye, James Matthew Rehg, and Ismini Lourentzou. Fine-grained preference optimization improves spatial reasoning in vlms, 2026. 12

[73] Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh, Georgios Kaissis, Volker Tresp, and Bjoern Menze. Relationformer: A unified framework for image-to-graph generation, 2022. 12

[74] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding, 2016. 23

[75] Evgeny Smirnov, Nikita Garaev, Vasiliy Galyuk, and Evgeny Lukyanets. Prototype memory for large-scale face representation learning. *IEEE Access*, 10:12031–12046, 2022. 12

[76] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 4080–4090, Red Hook, NY, USA, 2017. Curran Associates Inc. 12

[77] Elizabeth S. Spelke. Principles of object perception. *Cognitive Science*, 14(1):29–56, 1990. 1, 10

[78] Fan-Yun Sun, Weiyu Liu, Siyi Gu, Dylan Lim, Goutam Bhat, Federico Tombari, Manling Li, Nick Haber, and Jiajun Wu. Layoutvlm: Differentiable optimization of 3d layout via vision-language models, 2025. 12

[79] Ivan E. Sutherland and Gary W. Hodgman. Reentrant polygon clipping. *Communications of the ACM*, 17(1):32–42, 1974. 27

[80] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 16

[81] Guan Wang, Zhimin Li, Qingchao Chen, and Yang Liu. Oed: Towards one-stage end-to-end dynamic scene graph generation, 2024. 11

[82] Qi-Wei Wang, Da-Wei Zhou, Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. Few-shot class-incremental learning via training-free prototype calibration. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 12

[83] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy, 2024. 20

[84] Wenqi Wang, Reuben Tan, Pengyue Zhu, Jianwei Yang, Zhengyuan Yang, Lijuan Wang, Andrey Kolobov, Jianfeng Gao, and Boqing Gong. Site: towards spatial intelligence thorough evaluation, 2025. 12

[85] Yu Wang, Dmitry Krotov, Yuanzhe Hu, Yifan Gao, Wangchunshu Zhou, Julian McAuley, Dan Gutfreund, Rogerio Feris, and Zexue He. M+: Extending memoryllm with scalable long-term memory, 2025. 12

[86] Yufu Wang, Yu Sun, Priyanka Patel, Kostas Daniilidis, Michael J. Black, and Muhammed Kocabas. Prompthmr: Promptable human mesh recovery, 2025. 3

[87] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. 3, 10, 13, 14, 23

[88] Chuan Wen, Dinesh Jayaraman, and Yang Gao. Can transformers capture spatial relations between objects?, 2024. 12

[89] Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence, 2025. 12

[90] Rongsen Wu, Jie Xu, Hao Zheng, Zhiyuan Xu, Zixuan Li, Shixue Cheng, and Shumao Zhang. Spatio-temporal features with global–local transformer model for video scene graph generation. *Digital Communications and Networks*, 2025. 11

[91] Shengqiong Wu, Hao Fei, Jingkang Yang, Xiangtai Li, Juncheng Li, Hanwang Zhang, and Tat seng Chua. Learning 4d panoptic scene graph generation from rich 2d visual scene, 2025. 2, 10, 12

[92] Zhu Xu, Ting Lei, Zhimin Li, Guan Wang, Qingchao Chen, Yuxin Peng, and Yang liu. Trkt: Weakly supervised dynamic scene graph generation with temporal-enhanced relation-aware knowledge transferring, 2025. 11

[93] Jingkang Yang, Wenxuan Peng, Xiangtai Li, Zujin Guo, Liangyu Chen, Bo Li, Zheng Ma, Kaiyang Zhou, Wayne Zhang, Chen Change Loy, and Ziwei Liu. Panoptic video scene graph generation, 2023. 10, 11

[94] Jingkang Yang, Jun Cen, Wenxuan Peng, Shuai Liu, Fangzhou Hong, Xiangtai Li, Kaiyang Zhou, Qifeng Chen, and Ziwei Liu. 4d panoptic scene graph generation, 2024. 10, 12

[95] Yuncong Yang, Jiageng Liu, Zheyuan Zhang, Siyuan Zhou, Reuben Tan, Jianwei Yang, Yilun Du, and Chuang Gan. Mindjourney: Test-time scaling with world models for spatial reasoning, 2025. 12

[96] Jin Yao, Hao Gu, Xuweiyi Chen, Jiayun Wang, and Zeyu Shangguan. Ov-mono3d: Open-vocabulary monocular 3d object detection, 2023. 25, 26

[97] Qi Xun Yeo, Yanyan Li, and Gim Hee Lee. Statistical confidence rescoring for robust 3d scene graph generation from multi-view images, 2025. 12

[98] Kanghoon Yoon, Kibum Kim, Jaehyung Jeon, Yeonjun In, Donghyun Kim, and Chanyoung Park. Ra-sgg: Retrieval-augmented scene graph generation framework via multi-prototype learning, 2024. 12

[99] Chenyangguang Zhang, Alexandros Delitzas, Fangjinhua Wang, Ruida Zhang, Xiangyang Ji, Marc Pollefeys, and Francis Engelmann. Open-vocabulary functional 3d scene graphs for real-world indoor spaces, 2025. 2, 12

[100] Hang Zhang, Zhuoling Li, and Jun Liu. Scenellm: Implicit language reasoning in llm for dynamic scene graph generation, 2025. 11

[101] Jia Zhao, Ziyang Cao, Huiling Wang, Xu Wang, and Yingzhou Chen. Profusion: Multimodal prototypical networks for few-shot learning with feature fusion. *Symmetry*, 17(5), 2025. 12

[102] Shu Zhao and Huijuan Xu. Less is more: Toward zero-shot local scene graph generation via foundation models, 2023. 11

[103] Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors, 2025. 12

[104] Yiwu Zhong, Jing Shi, Jianwei Yang, Chenliang Xu, and Yin Li. Learning to generate scene graph from natural language supervision, 2021. 11

[105] Zijian Zhou, Miaojing Shi, and Holger Caesar. Vlprompt: Vision-language prompting for panoptic scene graph generation, 2024. 11

[106] Fangrui Zhu, Hanhui Wang, Yiming Xie, Jing Gu, Tianye Ding, Jianwei Yang, and Huaizu Jiang. Struct2d: A perception-guided framework for spatial reasoning in mllms, 2025. 12

[107] Guangming Zhu, Liang Zhang, Youliang Jiang, Yixuan Dang, Haoran Hou, Peiyi Shen, Mingtao Feng, Xia Zhao, Qiguang Miao, Syed Afaq Ali Shah, and Mohammed Bennamoun. Scene graph generation: A comprehensive survey, 2022. 2, 11

[108] Yiqi Zhu, Ziyue Wang, Can Zhang, Peng Li, and Yang Liu. Cospace: Benchmarking continuous space perception ability for vision-language models, 2025. 12

A. Technical Appendices and Supplementary Material

Contents

B. Motivation

Scene graph generation has matured considerably since the introduction of Visual Genome [35], yet the dominant paradigm remains *frame-centric*: a model observes a single image or a short video clip and outputs a flat graph whose nodes are the objects detected in the current view and whose edges encode pairwise semantic relationships. This design inherits three fundamental limitations:

1. **View dependence.** Every output graph is anchored to the camera's image plane. Object positions are expressed as 2D bounding boxes whose coordinates shift as the camera moves, offering no shared spatial reference frame.
2. **Observation gating.** Objects exist in the graph only if they are detected in the current frame. When an object leaves the field of view or becomes occluded it is silently dropped from the graph. Thus, there is no mechanism to retain its identity, location, or past relationships.
3. **Temporal fragmentation.** VidSGG methods that incorporate temporal modeling (e.g., STTran [11], Tempura [59], ImparTail [65]) process a sliding window of frames and produce per-frame graphs without a persistent, globally consistent world model.

These limitations are tolerable for image retrieval or captioning, where a snapshot summary suffices. They become critical, however, for downstream tasks that require *persistent world-state reasoning*: robotic manipulation that must track tools after they leave the camera, embodied navigation that builds a spatial memory of traversed rooms, and activity understanding that reasons about long-horizon human–object interactions spanning minutes.

Object Permanence as a Design Principle. Developmental psychology has long recognized *object permanence* [77] as a foundational cognitive milestone: the understanding that objects continue to exist when they leave direct perception. Infants as young as four months exhibit surprise when an object that was hidden behind a screen fails to reappear, implying an internal model of the world that persists beyond the immediate sensory input. This principle motivates our core design decision. A world-centric scene graph must maintain a complete inventory of all interacting objects known to exist in the environment; *both observed* ($\mathcal{O}^t$) *and unobserved* ($\mathcal{U}^t$). It must predict relationships involving every interacting object at every timestamp, regardless of its current visibility. Concretely, the world scene graph $\mathcal{G}^t_{\mathcal{W}}$ must cover observed–observed, observed–unobserved, and unobserved–unobserved object pairs.

B.1. Gaps in Existing Benchmarks

Table 4 compares existing SGG task formulations along eight diagnostic axes. Several observations motivate the need for WSGG: **Video SGG** (e.g., Action Genome [29], VidVRD [71]) operates in 2D, restricts the object scope to currently detected objects, and provides no world coordinate frame. Relationships are defined only within individual frames or short temporal windows. **3D SGG** (e.g., 3D scene graphs from point clouds [32, 33]) localizes objects in a shared 3D coordinate frame but lacks any temporal dimension: the graph is a static snapshot of a reconstructed scene. **4D SGG** [91, 94] combines 3D localization with temporal reasoning but typically requires multi-view or RGB-D input, restricts the object scope to per-frame detections, and does not maintain relationships for unobserved objects. **Panoptic (V)SGG** [93] enriches the object vocabulary to include both things and stuff but remains image-plane anchored and does not reason about occluded or out-of-view entities. No existing benchmark jointly satisfies the three requirements that define WSGG: (i) 3D spatial grounding in a world coordinate frame, (ii) temporally persistent object identity and tracking, and (iii) dense semantic annotations for *all* interacting objects.

Table 4. **SGG Tasks. Bold** entries highlight the unique capabilities of WSGG.

Task	Input	Spatial Dimension	Temporal	Localization	Object Scope	Coordinate Frame	Unobserved Objects	Relationship Persistence
Image SGG	Image	2D	✗	2D BBox	Detected	Image	✗	Per-frame
Video SGG	Video	2D	✓	2D BBox	Detected / frame	Image	✗	Cross-frame
3D SGG	3D Scan	3D	✗	3D BBox	All in scan	Scene	✗	Per-scan
4D SGG	Video + 3D	3D	✓	3D BBox	Detected / frame	Scene	✗	Cross-frame
Panoptic SGG	Image	2D	✗	2D Mask	Things + Stuff	Image	✗	Per-frame
Panoptic VSGG	Video	2D	✓	2D Mask	Things + Stuff	Image	✗	Cross-frame
WSGG (ours)	**Video**	**3D**	✓	**3D BBox**	**World state**	**World**	✓	**Through occlusion**

B.2. Why Monocular Video?

Although multi-view or RGB-D setups simplify 3D reconstruction, they impose hardware requirements that are incompatible with many practical deployment settings. Egocentric and surveillance videos, internet video, and autonomous driving footage are overwhelmingly monocular. By grounding WSGG on monocular input, we maximize applicability while simultaneously posing the harder perceptual challenge. Advances in feed-forward monocular 3D reconstruction (π^3 [87]) make it feasible to recover metrically consistent world-frame geometry from a single video, closing the gap between monocular and multi-view pipelines for the purpose of object localization and scene graph construction.

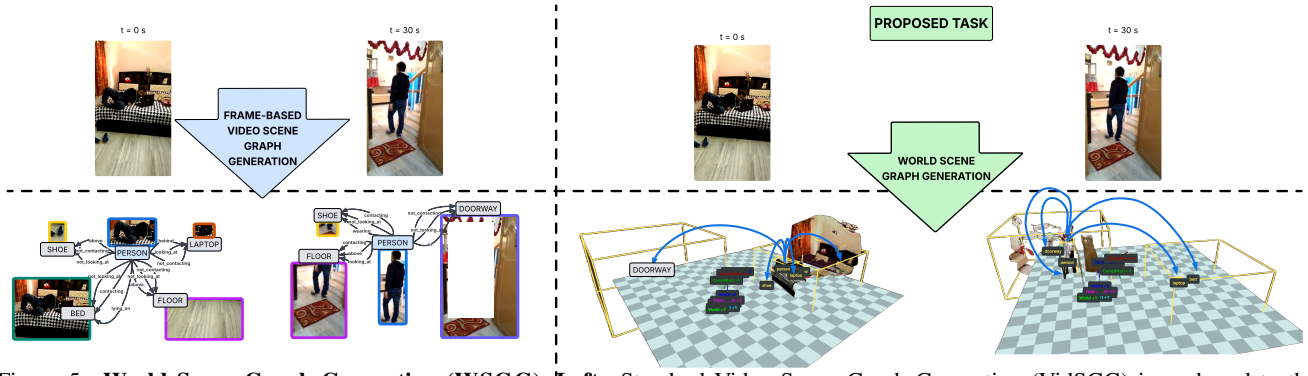

Figure 5. **World Scene Graph Generation (WSGG). Left:** Standard Video Scene Graph Generation (VidSGG) is anchored to the instantaneous camera view. Objects are localized as 2D bounding boxes in the image plane, and relationships are predicted only for currently detected entities. When an object exits the field of view or becomes occluded, it is silently dropped from the graph along with all of its relationships, resulting in an incomplete and temporally fragmented scene understanding. **Right:** Our proposed WSGG task grounds scene understanding in a persistent, global 3D world coordinate frame. Every object in the environment is represented by a 3D oriented bounding box (OBB) that persists across frames, regardless of whether the object is currently visible in the camera view. At each timestamp, the model outputs a *complete* world scene graph containing *all* known objects; observed ($\mathcal{O}^t$) and unobserved ($\mathcal{U}^t$); together with their pairwise semantic relationships (attention, spatial, and contacting predicates). As illustrated at t=30s, objects that have left the camera's field of view (e.g., bed, laptop) remain precisely localized in 3D world coordinates, enabling the model to continue predicting meaningful relationships for them. Blue curves drawn from person to objects denote predicted relationships. This view-independent, temporally persistent representation supports downstream tasks such as embodied navigation, robotic manipulation, and long-horizon activity understanding that require reasoning about the full state of the world, not just its currently visible slice.

C. Extended Related Work

C.1. Structured Scene Understanding

Scene graphs were first introduced as structured representations that encode objects and their pairwise relationships for static images [8, 35, 107]. SGTR [41] formulated end-to-end scene graph generation via Transformers that jointly predict entity and predicate proposals, while scene graph generation from natural language supervision [104] bridges vision–language alignment with structured scene representations. Environment-invariant curriculum relation learning [57] progressively trains on harder relational patterns for fine-grained SGG.

Extending scene graphs to the temporal domain requires modeling how relationships evolve across frames. The Video Visual Relation Detection (VidVRD) benchmark [71] first formalised this task, while the Action Genome (AG) dataset [29] introduced compositional spatio-temporal scene graphs grounded in human-object interactions. Following AG, several Transformer-based methods have been proposed: STTran [11] captures intra-frame spatial context and inter-frame temporal evolution via dual spatial-temporal attention; memory-augmented architectures [18] exploit long-term temporal dependencies; Tempura [59] mitigates predicate-level imbalance; and ImparTail [65] addresses unbiased and robust spatio-temporal scene graph generation. Recent methods further pushed architectural boundaries: OED [81] is a one-stage end-to-end VidSGG pipeline; UNO [37] unifies one-stage VidSGG via object-centric visual representation learning; HyperGLM [61] introduces a hypergraph approach; and action scene graphs [69] enable long-form understanding of egocentric videos. Panoptic VidSGG [93] extends scene graphs to jointly reason about stuff and thing regions. SceneSayer [64] formalised Scene Graph Anticipation (SGA).

Weakly-Supervised and Open-Vocabulary approaches have also gained traction: natural language supervision [31] for weakly-supervised VidSGG; TRKT for temporal-enhanced weakly-supervised [92] knowledge transferring; LLM4SGG [30] for LLM-guided weakly-supervised SGG; zero-shot visual relation detection via composite visual cues from LLMs [39]; zero-shot local SGG via foundation models [102]; and VLPrompt [105] for vision-language prompting in panoptic SGG. Additional Transformer-based VidSGG methods include: SKET [66], which embeds spatial-temporal knowledge into the Transformer; EGTR [28], which extracts graphs directly from Transformers; SpGLT [90], which combines global-local Transformer features with pose-aware visibility matrices; SceneLLM [100], which leverages implicit language reasoning in LLMs; and TD^2-Net [45], which jointly denoises and debiases dynamic scene graphs.

3D and 4D Scene Graph Generation Departing from the 2D setting, 3D scene graphs encode objects and their relationships in a shared world coordinate frame. 3D scene graphs were pioneered [32] as sparse, semantic representations for intelligent agents. 3D spatial multimodal knowl-

edge accumulation [17] advanced scene graph prediction in point clouds. SGRec3D [33] performs self-supervised 3D scene graph learning via object-level reconstruction; open-vocabulary functional 3D scene graphs [99] target real-world indoor spaces; and Nyffeler et al. [62] construct hierarchical 3D scene graphs outdoors. Statistical confidence rescoring [97] improves robust 3D SGG from multi-view images, and FROSS [25] achieves faster-than-real-time online 3D semantic SGG from RGB-D images. 4D scene graphs extend the 3D domain to the temporal dimension. 4D panoptic scene graph generation [94] extends scene graphs to the joint spatial-temporal-panoptic domain; learning 4D panoptic scene graphs from rich 2D visual scenes was explored in [91]; and RealGraph [44] contributes a multi-view dataset for 4D real-world context graphs.

Scene Graph Applications. Scene graphs have been applied beyond recognition: semantic image manipulation [13] and MIGS [16] demonstrate their utility for controllable synthesis; dynamic graph representations [21] have been used for video dialog; SSGVS [12] synthesises videos from semantic scene graphs; Relationformer [73] provides a unified framework for image-to-graph generation; and HIG [60] uses hierarchical interlacement graphs for video understanding. In robotics, Scene Graph Memory [36] models dynamic environments, relational predicate grounding [56] learns spatial and contact relationships from interactive manipulation, and ESCA [27] contextualises embodied agents via SGG.

C.2. Vision-Language Models for Spatial Understanding

VLMs have recently been adapted for spatial understanding tasks that require reasoning about 3D geometry and object relationships from visual input. SpatialVLM [5] endows VLMs with spatial reasoning through internet-scale spatial data; SpatialRGPT [9] grounds spatial reasoning through depth-aware region representations; SpatialBot [2] enables precise spatial understanding with depth integration; SD-VLM [7] encodes depth directly for spatial measuring; See&Trek [40] introduces training-free spatial prompting of multimodal LLMs; fine-grained preference optimization [72] and perspective-aware reasoning via mental imagery simulation [38] improve VLM spatial capabilities; Struct2D [106] is a perception-guided framework for spatial reasoning in large multimodal models; spatial relation capture by transformers was investigated in [88]; and SpatialMLLM [89] boosts MLLM capabilities in visual-based spatial intelligence. Benchmarks such as SITE [84], STI-Bench [43], and CoSpace [108] evaluate whether MLLMs are ready for precise spatial-temporal world understanding.

A growing line of work endows VLMs with explicit 3D understanding. SpatialLLM [52] is a compound 3D-informed design towards spatially-intelligent large multimodal models; SpatialReasoner [51] enables explicit and generalizable 3D spatial reasoning; SpatialLM [54] trains large language models for structured indoor modeling; scalable spatial intelligence via 2D-to-3D data lifting was proposed in [55]; LayoutVLM [78] introduces differentiable optimisation of 3D layout via vision-language models; MindJourney [95] leverages test-time scaling with world models for spatial reasoning; and learning from videos with 3D vision-geometry priors [103] enhances multimodal LLMs.

C.3. Neural Memory Models

Modern Hopfield networks and dense associative memories offer exponential storage capacity [1, 10, 26, 50] and differentiable retrieval dynamics, making them attractive for architectures requiring persistent state. The Hopfield framework has been extended to long sequences as attention-compatible memory layers [4], while biological extensions [34] and connections to diffusion models [23, 24] have broadened the theoretical foundations. These mechanisms have been integrated into large-scale models: CAMELoT [22] serves as a training-free associative memory for frozen LLMs, M+ [85] scales knowledge retention via long-term memory, and CLOOB [20] for contrastive vision-language pre-training.

Prototypical networks. [76] introduced the idea of learning a metric space in which classification is performed by computing distances to prototype representations of each class, establishing a foundational framework for few-shot learning. This paradigm has been extended in several directions: Graph Prototypical Networks [14] adapt prototypical representations to attributed networks, enabling few-shot node classification by constructing a pool of semi-supervised tasks that mimic the test environment; ProFusion [101] augments prototypical networks with multimodal feature fusion, constructing image, text, and fused prototypes from vision-language pre-trained models for robust few-shot classification; prototype memory [75] scales prototypical representations for large-scale face recognition; and training-free prototype calibration [82] addresses few-shot class-incremental learning by adjusting prototypes without retraining. In the context of scene graph generation, RA-SGG [98] proposes retrieval-augmented SGG via multi-prototype learning, where multiple prototypes per predicate class capture the diversity of visual relationship appearances and mitigate the long-tail distribution problem inherent in relationship annotations.

D. 3D Scene Construction Details

We construct per-timestamp 3D scene representations from egocentric Action Genome [29] videos using a feed-forward neural reconstruction model followed by post-hoc geometric alignment. The pipeline comprises four stages: (i) frame sampling and preprocessing, (ii) feed-forward 3D inference via the π^3 [87] model, (iii) static–dynamic scene decomposition, and (iv) per-frame alignment via trimmed Iterative Closest Point (ICP) with weighted Kabsch fitting.

4D Scene Representation. The output of the pipeline is a single 4D scene per video:

$$\text{4D Scene} = \Big(\underbrace{\mathcal{S}}_{\text{static background}} , \ \big\{ \underbrace{(\mathcal{F}^t,\ \mathbf{T}^t)}_{\text{per-frame dynamic}} \big\}_{t=1}^{T} \Big) \quad (1)$$

where $\mathcal{S}$ is the static background point cloud (built once and shared across all frames), $\mathcal{F}^t$ is the per-frame dynamic point cloud for frame I^t, and $\mathbf{T}^t \in \mathbb{R}^{4\times4}$ is the ICP-refined camera-to-world pose. This 4D scene provides the geometric substrate from which the world state $\mathcal{W}^t$ is populated: object detections within each frame's point cloud $\mathcal{F}^t$ yield the 3D oriented bounding boxes $\mathbf{b}_k^t$ for all interacting objects, while the refined poses $\mathbf{T}^t$ enable camera-relative reasoning for visibility determination. This representation supports two use cases:

- **World-frame 3D bounding boxes**: Per-frame foreground points provide the geometric basis for fitting OBB annotations to each object.
- **Scene graph construction**: The grounded full-frame representation provides complete spatial context for reasoning about relationships across time.

D.1. Adaptive Frame Sampling via Feature Descriptor Matching

Raw Action Genome [29] videos are captured at 24–30 fps, producing substantial temporal redundancy. Rather than uniform subsampling (e.g., every k-th frame), we employ a *Feature Descriptor Sampler* that adapts its density to camera/object motion: it retains more frames during rapid motion and fewer during static shots, using SIFT feature matching [49] and RANSAC homography estimation [19] to quantify visual overlap.

Homography-Based Overlap Estimation. The overlap between a retained reference frame I_A and a candidate frame I_B is estimated in four stages:

1. **SIFT extraction**: Both frames are converted to grayscale and SIFT [49] keypoints $\mathcal{K}$ and 128-d descriptors $\mathbf{D}$ are extracted. If either frame yields < 4 keypoints, the overlap defaults to 0.

Algorithm 1 Greedy Frame Selection by Visual Overlap

Require: Frames $\{I_0, \ldots, I_{T-1}\}$, threshold τ
Ensure: Selected frame indices $\mathcal{S}$
$\mathcal{S} \leftarrow \{0\}$; $I_{\text{ref}} \leftarrow I_0$
for $t = 1, \ldots, T-1$ **do**
  $\alpha \leftarrow \text{HomographyOverlap}(I_{\text{ref}}, I_t)$
  if $\alpha < \tau$ **then** ▷ Sufficient new content
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{t\}$; $I_{\text{ref}} \leftarrow I_t$
  end if
end for
return $\mathcal{S}$

2. **Feature matching**: Descriptors in I_A are matched to I_B via brute-force k-NN (k=2) with Lowe's ratio test [49] ($\rho = 0.75$), retaining only matches where the best candidate is substantially better than the second-best:

$$\mathcal{G} = \big\{(i,j) : \|\mathbf{d}_i^A - \mathbf{d}_{j_1}^B\|_2 \,/\, \|\mathbf{d}_i^A - \mathbf{d}_{j_2}^B\|_2 < \rho\big\}. \quad (2)$$

3. **RANSAC homography**: A projective transformation $\mathbf{H} \in \mathbb{R}^{3\times3}$ mapping I_B into I_A's coordinate frame is estimated from the good matches via RANSAC [19] (reprojection threshold 4.0 px, 2000 iterations, 0.995 confidence). The homography is rejected if < 4 inliers remain.
4. **Polygon intersection**: The four corners of I_B are warped into I_A's frame via $\mathbf{H}$, and the visual overlap is the normalised intersection area:

$$\alpha = \frac{\text{Area}(\mathcal{P}_A \cap \mathcal{P}'_B)}{\text{Area}(\mathcal{P}_A)} \in [0,1], \quad (3)$$

where $\mathcal{P}_A$ is the frame A rectangle and $\mathcal{P}'_B$ is the warped frame B quadrilateral, computed using the Shapely geometry library.

While Action Genome scenes are not strictly planar, the homography provides a sufficiently accurate alignment for overlap estimation in indoor environments dominated by large planar surfaces (walls, floors, tables). The RANSAC inlier ratio also serves as an implicit quality measure; low inlier counts indicate strong parallax or dynamic content, both signals of significant visual change.

Greedy Adaptive Selection. Given a video with T frames and an overlap threshold $\tau = 0.95$, Algorithm 1 greedily builds the selected set $\mathcal{S}$. The reference I_{ref} advances only upon frame retention, creating a natural temporal hysteresis: the reference stays fixed during static shots (skipping many near-duplicates) and advances rapidly during camera or object motion. At $\tau = 0.95$, any frame contributing $\geq 5\%$ novel visual content relative to the current reference is kept. The overlap-based criterion mirrors keyframe insertion in visual SLAM [58] and SfM [70],

where a new keyframe is triggered once tracked-feature counts fall below a reference threshold; a condition equivalent to low homography overlap. Our formulation makes this criterion geometrically explicit. Thus, the sampler compresses Action Genome videos from ∼900–9000 raw frames to ∼30–150 frames.

Caveats: (a) To preserve all supervised training targets, every frame carrying a ground-truth bounding-box annotation is unconditionally added after overlap filtering: $\mathcal{S} \leftarrow \mathrm{sort}(\mathcal{S} \cup \mathcal{A}_{\text{annotated}})$. (b) If the resulting set contains fewer than $K_{\min} = 17$ frames, the sampler reverts to uniform subsampling at stride $\Delta = \max(1, \lfloor T/K_{\min} \rfloor)$ and re-injects annotated frames, ensuring enough distinct viewpoints for reliable 3D reconstruction.

D.2. Feed-Forward 3D Inference via π^3

We employ the π^3 [87] model, a feed-forward neural network for visual geometry reconstruction. Given N unposed images, π^3 jointly predicts per-pixel 3D points, per-pixel confidence scores, and camera-to-world SE(3) poses in a single forward pass, without requiring a designated reference view. The model uses a frozen DINOv2-Large [63] encoder and a permutation-equivariant transformer decoder with alternating intra-view and cross-view attention, ensuring the reconstruction is invariant to the ordering of input views. We use π^3 as a black-box reconstruction module; we refer the reader to [87] for full architectural details.

Input Frame Preparation. From the selected frame set $\mathcal{S}$, we prepare two parallel sets of inputs: (a) **Static frames**: Inpainted versions in which all dynamic foreground objects (persons, interacted objects) have been removed via rectangular mask inpainting, depicting only the static background. (b) **Dynamic frames**: The original unmodified video frames at the same indices, containing all actors and objects in their natural positions. Both sets share identical indices and resolution to enable pixel-level correspondence between the reconstructions.

Inference Procedure. The static and dynamic frame sets are processed in **two independent forward passes** in mixed precision (bfloat16/float16). Each pass takes N frames batched as a tensor of shape $(1, N, 3, H, W)$ and produces all outputs simultaneously. This independence is critical: the two reconstructions inhabit different, unaligned world coordinate systems, necessitating the geometric alignment described in Section D.4. For a set of N input images, the model jointly estimates the quantities summarized in Table 5.

Post-Processing. Two filtering steps are applied to the raw predictions before downstream use: (i) **Depth-edge suppression**: pixels at depth discontinuities (relative depth difference $> 3\%$ in a 3×3 neighborhood) have their confidence set to zero; (ii) **Confidence thresholding**: points below $\tau_{\text{static}} = 0.1$ (static background) or $\tau_{\text{frame}} = 0.01$ (per-frame dynamic) are discarded. World-frame 3D points are obtained by applying the predicted camera-to-world pose $\mathbf{T}_i$ to the local points via homogeneous transformation: $\mathbf{p}_{\text{world}} = (\mathbf{T}_i \cdot \tilde{\mathbf{p}}_{\text{local}})_{1:3}$.

Table 5. Quantities estimated by the π^3 model for N input views at resolution $H \times W$.

Quantity	Shape	Type	Description
Local points	$(N, H, W, 3)$	float32	Per-pixel 3D coordinates in camera frame
World points	$(N, H, W, 3)$	float32	Per-pixel 3D coordinates in world frame
Confidence	$(N, H, W, 1)$	float32	Per-pixel reliability score $\in [0, 1]$
Camera poses	$(N, 4, 4)$	float32	Camera-to-world SE(3) matrices

D.3. Static-Dynamic Scene Decomposition

The two-pass inference strategy produces complementary scene representations:

Static Scene. The static reconstruction uses inpainted frames as input, yielding a clean, object-free 3D point cloud of the background environment. This point cloud is constructed by:

1. Applying a confidence threshold $\tau_{\text{static}} = 0.1$ to filter low-quality points.
2. Suppressing near-black pixels (per-channel intensity ≤ 8) that coincide with low confidence ($c < 1.0$), which removes inpainting artifacts that often manifest as dark, uncertain regions.
3. Removing all points with non-finite coordinates (NaN or Inf).

The resulting point cloud $\mathcal{S} = \{(\mathbf{s}_j, \mathbf{c}_j^{\text{rgb}})\}_{j=1}^{M}$ provides a stable geometric reference frame for subsequent alignment. The static scene is computed once per video and serves as the persistent, time-invariant component of the 4D scene.

Dynamic Scene. The dynamic reconstruction uses original (unmodified) frames, producing per-frame point clouds that include all actors and objects. Each frame t yields a set of world-frame 3D points $\mathcal{D}_t = \{(\mathbf{d}_k^t, \mathbf{c}_k^{t,\text{rgb}})\}$ and camera pose $\mathbf{T}_t^{\text{dyn}}$.

D.4. Per-Frame Geometric Alignment via Trimmed ICP

Since the static and dynamic scenes are reconstructed independently (through separate feed-forward passes of the π^3 model), their world coordinate frames are not aligned. We register each dynamic frame's point cloud to the static background via **Trimmed Iterative Closest Point (ICP)** with **confidence-weighted Kabsch fitting**. This can be viewed as a form of post-hoc bundle adjustment that refines both

Algorithm 2 Trimmed ICP with Weighted Kabsch Alignment

Require: Source points $\mathbf{A} = \{\mathbf{a}_k\}_{k=1}^{K} \subset \mathbb{R}^3$ (dynamic frame)
Require: Target points $\mathbf{B} = \{\mathbf{b}_j\}_{j=1}^{M} \subset \mathbb{R}^3$ (static background)
Require: Per-point confidence weights $\{w_k\}_{k=1}^{K}$
Require: Parameters: $I_{\max} = 100$, $\epsilon = 10^{-5}$, $\rho = 0.8$
Ensure: Rigid transform $(\mathbf{R}_{\text{total}}, \boldsymbol{\tau}_{\text{total}})$

Build KD-tree $\mathcal{T}$ on target $\mathbf{B}$
$\mathbf{R}_{\text{total}} \leftarrow \mathbf{I}_3$, $\boldsymbol{\tau}_{\text{total}} \leftarrow \mathbf{0}$
$\mathbf{A}' \leftarrow \mathbf{A}$ ▷ working copy
for iter $= 1$ **to** $I_{\max}$ **do**
  $(\{d_k\}, \{\text{nn}_k\}) \leftarrow \mathcal{T}.\text{query}(\mathbf{A}', k = 1)$ ▷ nearest neighbors
  $\mathcal{V} \leftarrow \{k : d_k < \infty\}$ ▷ valid correspondences
  $\delta \leftarrow \text{Percentile}(\{d_k\}_{k\in\mathcal{V}},\ \rho \cdot 100)$ ▷ trimming cutoff
  $\mathcal{V} \leftarrow \{k \in \mathcal{V} : d_k \le \delta\}$ ▷ keep closest ρ fraction
  if $|\mathcal{V}| < 10$ **then**
    break
  end if
  $(\mathbf{R}_{\text{inc}}, \boldsymbol{\tau}_{\text{inc}}) \leftarrow \text{WeightedKabsch}(\mathbf{A}'_{\mathcal{V}},\ \mathbf{B}_{\text{nn}_{\mathcal{V}}},\ \{w_k\}_{k\in\mathcal{V}})$
  $\mathbf{R}_{\text{total}} \leftarrow \mathbf{R}_{\text{inc}} \cdot \mathbf{R}_{\text{total}}$
  $\boldsymbol{\tau}_{\text{total}} \leftarrow \mathbf{R}_{\text{inc}} \cdot \boldsymbol{\tau}_{\text{total}} + \boldsymbol{\tau}_{\text{inc}}$
  $\mathbf{A}' \leftarrow (\mathbf{R}_{\text{inc}} \cdot \mathbf{A}'^{\top})^{\top} + \boldsymbol{\tau}_{\text{inc}}^{\top}$ ▷ update source
  $e \leftarrow \frac{1}{|\mathcal{V}|} \sum_{k\in\mathcal{V}} \|\mathbf{a}'_k - \mathbf{b}_{\text{nn}_k}\|^2$ ▷ MSE
  if $|e_{\text{prev}} - e| < \epsilon$ **then**
    break ▷ converged
  end if
end forreturn $(\mathbf{R}_{\text{total}}, \boldsymbol{\tau}_{\text{total}})$

the 3D point positions and camera poses to be consistent with the static reference frame.

Algorithm Overview. For each frame t, the alignment proceeds according to Algorithm 2. The source cloud is the dynamic frame's point set $\mathcal{D}_t$, and the target is the static background $\mathcal{S}$.

Weighted Kabsch Algorithm. At each ICP iteration, the optimal rigid transformation aligning source correspondences $\mathbf{A}_{\mathcal{V}}$ to target correspondences $\mathbf{B}_{\mathcal{V}}$ is found by solving the weighted least-squares problem:

$$\mathbf{R}^*, \boldsymbol{\tau}^* = \underset{\mathbf{R}\in\text{SO}(3),\ \boldsymbol{\tau}\in\mathbb{R}^3}{\arg\min} \sum_{k\in\mathcal{V}} w_k \|\mathbf{R}\mathbf{a}_k + \boldsymbol{\tau} - \mathbf{b}_{\text{nn}_k}\|^2 \quad (4)$$

where w_k is the per-point confidence from the π^3 model's sigmoid-activated confidence output. The closed-form solution proceeds as follows:

1. **Weighted centroids**:

$$\boldsymbol{\mu}_A = \frac{\sum_k w_k \mathbf{a}_k}{\sum_k w_k}, \quad \boldsymbol{\mu}_B = \frac{\sum_k w_k \mathbf{b}_k}{\sum_k w_k} \quad (5)$$

2. **Cross-covariance matrix**:

$$\mathbf{H} = \sum_{k\in\mathcal{V}} w_k (\mathbf{a}_k - \boldsymbol{\mu}_A)(\mathbf{b}_k - \boldsymbol{\mu}_B)^{\top} \in \mathbb{R}^{3\times 3} \quad (6)$$

3. **SVD decomposition**: $\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\top}$
4. **Optimal rotation** with reflection correction:

$$\mathbf{R}^* = \mathbf{V} \cdot \text{diag}\big(1,\ 1,\ \det(\mathbf{V}\mathbf{U}^{\top})\big) \cdot \mathbf{U}^{\top} \quad (7)$$

The diagonal correction ensures $\det(\mathbf{R}^*) > 0$, i.e., $\mathbf{R}^* \in \text{SO}(3)$.

5. **Optimal translation**:

$$\boldsymbol{\tau}^* = \boldsymbol{\mu}_B - \mathbf{R}^* \boldsymbol{\mu}_A \quad (8)$$

Trimming Strategy. Standard ICP is sensitive to outlier correspondences particularly problematic here because dynamic objects (persons, manipulated items) are present in the source but absent from the static target. These points will inevitably match to incorrect static surface regions. We employ **trimmed ICP** with fraction $\rho = 0.8$: at each iteration, only the closest 80% of correspondence pairs (ranked by Euclidean distance) are retained for the Kabsch fit. This allows up to 20% of source points to be outliers without corrupting the alignment.

Camera Pose Refinement. The ICP-derived rigid transform $\mathbf{T}_{\text{icp}} \in \text{SE}(3)$ is composed with the π^3-predicted camera pose to produce a refined pose aligned to the static reference frame. For camera-to-world (c2w) convention poses:

$$\mathbf{T}_t^{\text{refined}} = \mathbf{T}_{\text{icp}} \cdot \mathbf{T}_t^{\text{dyn}} \quad (9)$$

This follows from the fact that points in the dynamic world frame are transformed as $\mathbf{p}' = \mathbf{T}_{\text{icp}} \cdot \mathbf{p}$, so the camera-to-world mapping must be left-multiplied by the same transform. For the world-to-camera (w2c) convention we follow:

$$\mathbf{E}_t^{\text{refined}} = \mathbf{E}_t^{\text{dyn}} \cdot \mathbf{T}_{\text{icp}}^{-1} \quad (10)$$

D.5. Mask-Aware Scene Merging

For the final 4D scene construction, we leverage pre-computed interaction segmentation masks to selectively merge dynamic content with the static background. Per-frame binary interaction masks $\mathbf{M}_t \in \{0,1\}^{H\times W}$ are loaded from pre-computed segmentation outputs (from the SAM2 segmentation pipeline).

Table 6. Scene construction pipeline parameters.

Parameter	Value	Description
Backbone	DINOv2-L (ViT-L/14)	Frozen encoder with registers
Decoder depth	36 layers	Alternating intra-/cross-view attn
Positional encoding	RoPE2D (f=100)	2D rotary position embeddings
Precision	bfloat16 / float16	Mixed-precision inference
Frame sampling	10	Temporal stride for selection
τ_{static}	0.1	Confidence thr. (static scene)
τ_{frame}	0.01	Confidence thr. (dynamic)
Depth-edge rtol	0.03	Rel. tolerance for edge suppression
ICP iterations	100 (max)	Early stop at ϵ=10^{-5}
Trim fraction ρ	0.8	Keep closest 80% of corresp.
Dynamic voxel	0.01 m	Per-frame point reduction
Merge voxel	0.02 m	Static–dynamic overlap removal
Vertical FOV	0.96 rad ($\approx 55°$)	Pinhole camera model

Point Partitioning. For each frame t, the dynamic frame's valid points are split into two sets: (1) $\mathcal{D}_t^{\text{all}}$: All valid points (used for ICP alignment). (2) $\mathcal{D}_t^{\text{fg}} = \{k \in \mathcal{D}_t^{\text{all}} : \mathbf{M}_t(\text{pixel}(k)) > 0\}$: Foreground-only points (used for merging).

Selective Merging Procedure. The merging follows a two-stage approach:

1. **ICP alignment (full frame)**: The ICP registration (Section D.4) uses *all* valid dynamic points $\mathcal{D}_t^{\text{all}}$ in frame t—including both foreground and background—to estimate the rigid transform $\mathbf{T}_{\text{icp}}^{(t)}$. Using the full frame ensures a robust fit dominated by the abundant background correspondences.
2. **Foreground-only merging**: Only the masked foreground points $\mathcal{D}_t^{\text{fg}}$ are transformed by $\mathbf{T}_{\text{icp}}^{(t)}$ and concatenated with the static background. This prevents duplication of background geometry (which already exists in the static cloud) while placing detected objects into the static reference frame.

We then obtain the final merged point cloud for frame t (see Eq 11); then perform voxel deduplication.

$$\mathcal{P}_t = \mathcal{S} \cup \left\{ \mathbf{T}_{\text{icp}}^{(t)} \cdot \mathbf{d}_k^t \;\middle|\; k \in \mathcal{D}_t^{\text{fg}} \right\} \tag{11}$$

D.6. Implementation Details

Table 6 summarizes the key parameters of the scene construction pipeline. All computations were performed on a single NVIDIA GPU.

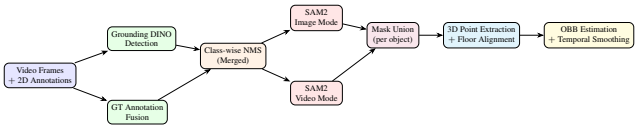

Figure 6. **3D bounding box pipeline overview.** Objects are detected via Grounding DINO fused with GT annotations, segmented via SAM2 in dual mode, projected into 3D via masked point extraction from the π^3 reconstruction, aligned to the floor plane, and enclosed with oriented bounding boxes (OBBs).

E. Geometric Annotation Details

E.1. 3D Bounding Box Pipeline Overview

We present a multi-stage pipeline that produces world-frame **oriented 3D bounding boxes (OBBs)** for all interacting objects in Action Genome [29] video clips. Starting from RGB frames and 2D annotations, the pipeline proceeds through five major stages: (i)**Detection:** object detection via Grounding DINO [46] with ground-truth annotation fusion, (ii)**Segmentation:** instance segmentation via SAM2 [67] in dual image-and-video mode, (iii)**Object Classification:** static/dynamic object classification via LLM-based reasoning, (iv)**Object Point Cloud Determination:** floor-aligned 3D point extraction with multiscale mask erosion, and (v)**OBB Estimation:** oriented bounding box estimation via PCA and floor-parallel minimum-area fitting, with temporal smoothing. The full pipeline is illustrated in Figure 6.

E.2. Object Detection

The detection stage produces per-frame, per-object 2D bounding boxes by fusing model-based detections with ground-truth annotations.

Grounding DINO Detection. We employ Grounding DINO [46] as a zero-shot, open-vocabulary detector. Each frame is prompted with the video's active object labels (e.g., `"person. chair. cup."`) using a box threshold of 0.25 and text threshold of 0.3. Output labels are normalized to a canonical vocabulary.

Active Object Labels & Static/Dynamic Classification. Active object categories are determined from two sources: (a) labels extracted directly from Action Genome ground-truth annotations, and (b) labels produced by an LLM-based reasoning pipeline using LLaMA [80]. The LLM classifies objects as *static* (e.g., floor, sofa, table) or *dynamic* (e.g., cup, phone, book) by extracting interactions from video captions and mapping them to the candidate label set via synonym-aware matching. This drives two separate detection-segmentation tracks: a static track for background elements and a dynamic track for actively used objects.

Ground Truth Annotation Fusion. To ensure annotated objects are not missed (especially small or occluded items), GT bounding boxes are assigned a pseudo-score of 1.001 and concatenated with Grounding DINO predictions. Two rounds of per-class NMS (IoU threshold 0.5) are applied; because GT scores exceed 1.0, they are preferred over detector duplicates while still allowing new discoveries.

E.3. Instance Segmentation via SAM2

Given per-frame detected bounding boxes, we produce per-object binary segmentation masks using SAM2 [67] (hiera-base-plus) in two complementary modes.

Image Mode. SAM2 processes each frame independently: per-label bounding boxes are passed as box prompts in single-mask mode, and the resulting per-label masks are combined via boolean OR to produce one union mask per frame.

Video Mode. SAM2's video predictor propagates masks temporally from seed frames. Seeds are selected from annotated (GT) frames when available (up to 10 per label, spaced $\geq$10 frames apart, prioritized by detection score); otherwise, the first detection of each label is used. A single propagation pass produces masks for all seeded objects across all frames, binarized at threshold 0.5.

Mask Union. The final per-object mask for each frame is the pixel-wise union of image-mode and video-mode masks:

$$\mathbf{M}_{\text{final}}^{(l,t)} = \mathbf{M}_{\text{image}}^{(l,t)} \vee \mathbf{M}_{\text{video}}^{(l,t)} \tag{12}$$

where l denotes the object label and t the frame index, preserving both per-frame precision and temporal consistency.

E.4. Floor Alignment via SMPL-Based Similarity Estimation

Before computing 3D bounding boxes, we establish a floor-aligned coordinate system by registering the π^3-reconstructed point cloud to a canonical frame where the floor plane is horizontal.

SMPL Mesh Correspondence. We leverage estimated SMPL [47] body meshes as anchors for scale-and-pose recovery. For each frame, 2D body keypoints (from OpenPose [3]) are paired with fitted SMPL 3D joint positions (in metric space), yielding K correspondence pairs $\{(\mathbf{s}_k, \mathbf{d}_k)\}$ between SMPL points and π^3 scene points. For each frame with $K \geq 3$ correspondences, a 7-DoF similarity transformation (scale s, rotation $\mathbf{R}$, translation $\boldsymbol{\tau}$) mapping scene space to metric space is estimated via RANSAC:

$$\mathbf{s}_k \approx s \cdot \mathbf{R} \cdot \mathbf{d}_k + \boldsymbol{\tau} \tag{13}$$

Global Floor Transform. Per-frame similarity estimates are robustly averaged (weighted by inlier count) to produce a single global similarity $(s_{\text{avg}}, \mathbf{R}_{\text{avg}}, \boldsymbol{\tau}_{\text{avg}})$, requiring at least $\max(3, 20\%)$ valid frames. In the floor-aligned frame, the Y-axis is the floor normal (up) and the XZ plane is the floor surface. The floor-alignment transform maps world points to floor-aligned coordinates:

$$\mathbf{p}_{\text{floor}} = \frac{1}{s_{\text{avg}}} \cdot \mathbf{R}_{\text{avg}} \cdot (\mathbf{p}_{\text{world}} - \boldsymbol{\tau}_{\text{avg}}) \tag{14}$$

and its inverse recovers world coordinates:

$$\mathbf{p}_{\text{world}} = s_{\text{avg}} \cdot \mathbf{R}_{\text{avg}}^{\top} \cdot \mathbf{p}_{\text{floor}} + \boldsymbol{\tau}_{\text{avg}} \tag{15}$$

E.5. Oriented 3D Bounding Box Computation

For each interacting object in each frame, we extract masked 3D points from the π^3 reconstruction and fit OBBs using two complementary methods: an unconstrained PCA-based OBB and a floor-parallel OBB.

Masked 3D Point Extraction. The 2D bounding box (GT or Grounding DINO) is resized to the π^3 prediction resolution and intersected with the corresponding SAM2 mask (Section E.3) to produce a tight object mask (falling back to the rectangular bbox if no mask exists). Points are then filtered by the per-pixel confidence map (thresholded at the frame's 5th-percentile, clamped $\geq 10^{-3}$) and by finite-coordinate validity.

Multiscale Erosion. To strip boundary artifacts from imprecise masks, we erode the object mask with elliptical kernels of size $\{0, 3, 5, 7, 10\}$ px, extract the surviving 3D points at each level (requiring $\geq$50 points), and select the erosion yielding the **minimum-volume** bounding box. Candidate volumes are computed from coordinate-wise extents in floor-aligned coordinates (Eq. (14)):

$$V = \prod_{d \in \{x,y,z\}} (\max_k p_{k,d}^{\text{floor}} - \min_k p_{k,d}^{\text{floor}}) \tag{16}$$

Using the filtered 3D points from the best erosion scale, we compute two OBB variants.

PCA OBB. A full 3-DoF rotation OBB is obtained via PCA on the object point cloud: the covariance matrix $\mathbf{C} = \frac{1}{N-1}\sum_k (\mathbf{p}_k - \boldsymbol{\mu})(\mathbf{p}_k - \boldsymbol{\mu})^{\top}$ is eigen-decomposed as $\mathbf{C} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\top}$, points are projected onto the principal axes $\mathbf{q}_k = \mathbf{V}^{\top}(\mathbf{p}_k - \boldsymbol{\mu})$, and extents are taken as $e_d = \max_k q_{k,d} - \min_k q_{k,d}$. The OBB center is adjusted for asymmetric distributions:

$$\mathbf{c}_{\text{obb}} = \boldsymbol{\mu} + \mathbf{V} \cdot \frac{\mathbf{q}_{\min} + \mathbf{q}_{\max}}{2} \tag{17}$$

Eight corners are generated from $\pm e_d/2$ along principal axes and rotated back to world via $\mathbf{V}$.

Floor-Parallel OBB. A constrained OBB restricted to 1-DoF yaw rotation around the floor normal: points are transformed to the floor-aligned frame (Eq. (14)), projected onto the XZ plane, and enclosed by a minimum-area 2D rectangle. The rectangle is extruded vertically over $[y_{\min}, y_{\max}]$ to produce 8 corners, which are transformed back to world coordinates via Eq. (15).

E.6. Temporal Smoothing

Bounding boxes computed independently per frame exhibit temporal jitter. Per-frame minimum-volume erosion selection (Section E.5) provides implicit scale normalization. For objects appearing across multiple frames, the 3D bounding box parameters (center, extents, yaw) are smoothed using a forward Kalman filter followed by Rauch-Tung-Striebel (RTS) backward smoothing, producing temporally consistent trajectories while preserving sharp transitions.

E.7. World-to-Final Coordinate Transform

For downstream consumption, all annotations are converted from the π^3 world frame to a **final coordinate system** that is floor-aligned (XY = floor, Z = up), metrically scaled (meters, via SMPL similarity), and origin-centered. The world-to-final transform is:

$$\mathbf{p}_{\text{final}} = \mathbf{A} \cdot (\mathbf{p}_{\text{world}} - \mathbf{o}_{\text{world}}) \tag{18}$$

where $\mathbf{A} = \frac{1}{s_{\text{avg}}}\mathbf{R}_{\text{avg}}$ and $\mathbf{o}_{\text{world}} = \boldsymbol{\tau}_{\text{avg}}$. Camera poses are similarly transformed:

$$\mathbf{R}_{\text{final}} = \mathbf{A} \cdot \mathbf{R}_{\text{world}}, \quad \boldsymbol{\tau}_{\text{final}} = \mathbf{A} \cdot (\boldsymbol{\tau}_{\text{world}} - \mathbf{o}_{\text{world}}) \tag{19}$$

All annotations including OBB corners, point clouds, camera poses, and floor meshes; are converted to this final frame and stored per-video. Table 7 summarizes the key hyperparameters used in the pipeline.

Table 7. 3D bounding box pipeline parameters.

Parameter	Value	Description
Detection		
Detector model	Grounding DINO (base)	Zero-shot open-vocabulary
Box threshold	0.25	Detection confidence
Text threshold	0.30	Text-box alignment
NMS IoU threshold	0.50	Per-class NMS
GT pseudo-score	1.001	Ensures GT survives NMS
Segmentation		
SAM2 model	hiera-base-plus	Balanced speed/quality
Mask threshold	0.50	Logit binarization
Max seeds per label	10	Video mode anchors
Anchor min gap	10 frames	Seed spacing
Precision	bfloat16	GPU autocast
Floor Alignment		
RANSAC iterations	500	Similarity estimation
RANSAC inlier thresh	0.03 m	Correspondence quality
Scale bounds	$[0.4, 3.0]$	Physical plausibility
Min valid frames	$\max(3, 20\%)$	Quality gate
3D Bounding Box		
Erosion kernels	$\{0, 3, 5, 7, 10\}$ px	Multiscale mask refinement
Min points per scale	50	Candidate quality
Confidence threshold	$\max(10^{-3}, P_5)$	Adaptive per-frame
Selection criterion	Minimum volume	Among erosion candidates

F. Semantic Annotation Details

World Scene Graphs provide a holistic, temporally-grounded representation of human-object relationships in video, extending beyond single-frame annotations to capture the full spatio-temporal dynamics of a scene. We employ current Vision-Language Models (VLMs) as a bridge to generate dense *pseudo semantic annotations* for *World Scene Graph Generation* (WSGG). Specifically, we leverage VLMs to predict human-object relationship labels across all annotated frames of a video, following AG's annotation schema [29]: (1) **Attention Relationships** (single-label): `looking_at`, `not_looking_at`, `unsure`; (2) **Contacting Relationships** (multi-label): `carrying`, `covered_by`, `drinking_from`, `eating`, `have_it_on_the_back`, `holding`, `leaning_on`, `lying_on`, `not_contacting`, `other_relationship`, `sitting_on`, `standing_on`, `touching`, `twisting`, `wearing`, `wiping`, `writing_on`; (3) **Spatial Relationships** (multi-label): `above`, `beneath`, `in_front_of`, `behind`, `on_the_side_of`, `in`.

F.1. Models

We employ three open-source state-of-the-art VLMs spanning different model families and parameter counts, all deployed behind the vLLM inference engine for efficient batched generation with tensor parallelism: (1) **Kimi-VL** (Kimi-VL-A3B-Instruct) - a lightweight mixture-of-experts VLM; (2) **InternVL 2.5** (InternVL2_5-8B-MPO) - a mid-scale VLM with multi-modal preference optimisation; (3) **Qwen 2.5-VL** (Qwen2.5-VL-7B-Instruct) - an instruction-

tuned VLM from the Qwen family. All models support variable-length multi-frame video input. A BGE embedding model (bge-large-en-v1.5) is additionally used for text-similarity computations in the RAG-based annotation method.

F.2. Generation Setup

Video Input. Video frames are loaded from disk and converted to a $T \times C \times H \times W$ float tensor. A pixel-budget constraint (`total_pixels`$= 128000$) is enforced to keep memory consumption manageable. We use annotation-driven per-frame clips. This provides fine-grained temporal context anchored to the frame being evaluated. The annotation-driven clips ensure that relationship predictions are grounded in the temporal neighbourhood of each annotated frame.

Relationship Query Prompt. For each object $o \in \mathcal{O}_v$ on each annotated frame, a structured prompt elicits predictions for all three relationship axes:

> *You are analyzing a video clip from a scene. The object "*$\{o\}$*" is one of the objects present in this video. It may or may not be visible in the current frame. A person IS visible in this frame. Based on the full video context, predict the relationships between the person and the object "*$\{o\}$*".*
> 1. ATTENTION: Pick EXACTLY ONE from: {attention labels}
> 2. CONTACTING: Pick ONE OR MORE from: {contacting labels}
> 3. SPATIAL: Pick ONE OR MORE from: {spatial labels}
>
> *Respond ONLY in JSON format...*

Discriminative Verification. After the generative prediction step, each predicted relationship label is independently verified through a discriminative Yes/No classification using the same VLM. For a predicted relationship (o, r), the verification prompt varies by axis:

> (1) **Attention Relationship**: "*In this video, is the person* $\{r\}$ *the* $\{o\}$*? Answer only Yes or No.*"; (2) **Contacting Relationship**: "*In this video, is the person* $\{r\}$ *the* $\{o\}$*? Answer only Yes or No.*"; (3) **Spatial Relationship**: "*In this video, is the* $\{o\}$ $\{r\}$ *the person? Answer only Yes or No.*"

All verification prompts across all frames and objects are collected and dispatched in a single batched VLM call (chunked into sub-batches of 64), yielding a confidence score p_{yes} for every predicted label.

F.3. Annotation Methods

We generate pseudo-annotations using:

RAG Generation. The primary method uses Retrieval-Augmented Generation (RAG) over a precomputed scene graph from Phase 1. Per-clip graphs are merged into a video-level directed graph, and a 4-step pipeline produces relationship predictions: (1) keyword extraction from query prompts via a batched VLM call; (2) cached node retrieval using BGE embeddings (cosine similarity > 0.5); (3) templatised node refinement via batched Yes/No verification; (4) frame-specific final answer using annotation-driven clips with graph-node context prepended to the prompt. Queries are processed and deduplicated across frames, and all batched calls are chunked into sub-batches of 64.

G. Manual Correction of 3D BBox Annotations

G.1. Correcting the Floor Transform

Monocular 3D reconstruction methods such as DUSt3R [83] produce an arbitrary world frame whose vertical axis does not coincide with gravity. We align the reconstructed scene into a *gravity-aligned reference frame* (XY-plane = floor, $+Z$ = up) via three stages: (i) *automatic floor alignment* from the reconstruction's floor mesh and global similarity transform, (ii) *manual interactive correction* with 6-DoF controls, and (iii) *automated XY-plane alignment* that analytically maps the corrected floor to $Z{=}0$.

Automatic Floor Alignment via Global Floor Similarity. The reconstruction pipeline extracts a floor mesh $\mathcal{F} = (\mathbf{V}_0, \mathbf{F}_0, \mathbf{C}_0)$ with vertex positions $\mathbf{V}_0 \in \mathbb{R}^{N_v \times 3}$, faces $\mathbf{F}_0$, and optional colors $\mathbf{C}_0$, together with a global floor similarity $(s_g, \mathbf{R}_g, \boldsymbol{\tau}_g)$. Floor vertices are transformed into the world frame as:

$$\mathbf{v}_{\text{world}} = s_g \cdot (\mathbf{v}_0 \cdot \mathbf{R}_g^\top) + \boldsymbol{\tau}_g, \quad \forall\, \mathbf{v}_0 \in \mathbf{V}_0. \tag{20}$$

From the columns of $\mathbf{R}_g$ we extract in-plane tangent directions $\boldsymbol{\ell}_1 = \mathbf{R}_g[:,0]$, $\boldsymbol{\ell}_2 = \mathbf{R}_g[:,2]$ and floor normal $\mathbf{n} = \mathbf{R}_g[:,1]$, forming the basis $\mathbf{F} = [\boldsymbol{\ell}_1 \mid \boldsymbol{\ell}_2 \mid \mathbf{n}]$. The world-to-floor rotation is $\mathbf{R}_{\text{align}} = \mathbf{F}^\top$ with translation $\boldsymbol{\tau}_{\text{align}} = -\mathbf{R}_{\text{align}} \cdot \boldsymbol{\tau}_g$:

$$\mathbf{x}_{\text{floor}} = \mathbf{R}_{\text{align}} \cdot \mathbf{x}_{\text{world}} + \boldsymbol{\tau}_{\text{align}}. \tag{21}$$

An optional ZY-plane mirror $\mathbf{M} = \text{diag}(-1,1,1)$ yields the final transform:

$$\mathbf{R}_{\text{final}} = \mathbf{M} \cdot \mathbf{R}_{\text{align}}, \quad \boldsymbol{\tau}_{\text{final}} = \mathbf{M} \cdot \boldsymbol{\tau}_{\text{align}}, \tag{22}$$

so that $\mathbf{x}_{\text{final}} = \mathbf{R}_{\text{final}} \cdot \mathbf{x}_{\text{world}} + \boldsymbol{\tau}_{\text{final}}$.

Manual Interactive Floor Correction. The automatic alignment is often imperfect. An interactive 3D tool renders the floor mesh, point cloud, and bounding boxes, allowing the annotator to adjust a correction transform $\mathbf{T}_\delta = (\mathbf{R}_\delta, \boldsymbol{\tau}_\delta, \mathbf{s}_\delta)$ via 6-DoF controls. Both original (wireframe) and corrected (solid) floor meshes are displayed simultaneously for visual feedback. The correction is persisted to Firebase per video; on reload it is applied automatically.

Automated XY-Plane Alignment. After manual correction the floor may still not lie in the canonical XY-plane. We compute $\mathbf{T}_{\text{XY}} = (\mathbf{R}_{\text{XY}}, \boldsymbol{\tau}_{\text{XY}})$ to map the corrected floor to $Z{=}0$ in six steps:

Step 1: Apply Floor Correction. The floor vertices $\mathbf{V}_0$ are first transformed by the correction $\mathbf{T}_\delta$ (Sec. G.1):

$$\mathbf{v}_{\text{corr}} = \mathbf{R}_\delta \cdot (\mathbf{s}_\delta \odot \mathbf{v}_0) + \boldsymbol{\tau}_\delta, \quad \forall\, \mathbf{v}_0 \in \mathbf{V}_0. \tag{23}$$

Step 2: Compute Floor Normal. The floor centroid $\bar{\mathbf{v}}$ and robust normal $\hat{\mathbf{n}}$ are estimated from triangle cross products (selecting the largest-magnitude result); the normal is flipped if needed so that $\hat{n}_z > 0$.

Step 3: Normal Alignment via Rodrigues' Rotation. The rotation $\mathbf{R}_{\text{norm}}$ aligning $\hat{\mathbf{n}}$ with $+Z$ is obtained via Rodrigues' formula with axis $\hat{\mathbf{k}} = \frac{\hat{\mathbf{n}} \times \hat{\mathbf{z}}}{\|\hat{\mathbf{n}} \times \hat{\mathbf{z}}\|}$ and angle $\phi = \arccos(\hat{\mathbf{n}} \cdot \hat{\mathbf{z}})$:

$$\mathbf{R}_{\text{norm}} = \cos\phi \cdot \mathbf{I} + \sin\phi \cdot [\hat{\mathbf{k}}]_\times + (1 - \cos\phi) \cdot \hat{\mathbf{k}}\hat{\mathbf{k}}^\top. \tag{24}$$

Step 4: In-Plane Rotation Correction. The residual in-plane misalignment is corrected by a Z-axis rotation $\mathbf{R}_z(\alpha)$ where $\alpha = -\text{atan2}(\tilde{x}_y, \tilde{x}_x)$ and $\tilde{\mathbf{x}} = \mathbf{R}_{\text{norm}} \cdot \mathbf{R}_\delta \cdot [1,0,0]^\top$. The combined rotation is $\mathbf{R}_{\text{XY}} = \mathbf{R}_z(\alpha) \cdot \mathbf{R}_{\text{norm}}$.

Step 5: Translation to Origin. The floor–origin intersection $\mathbf{p}_\cap = (\hat{\mathbf{n}} \cdot \bar{\mathbf{v}}) \cdot \hat{\mathbf{n}}$ is mapped to the origin: $\boldsymbol{\tau}_{\text{XY}} = -\mathbf{R}_{\text{XY}} \cdot \mathbf{p}_\cap$.

Step 6: Euler Angle Extraction. $\mathbf{R}_{\text{XY}}$ is decomposed into Euler angles $(\theta_x, \theta_y, \theta_z)$ (ZYX convention) for storage.

Bounding Box Refitting. After applying $\mathbf{T}_{\text{XY}}$, each OBB is refitted: inlier points (identified via the separating axis test) are transformed, and a 2D PCA-based OBB is estimated on the projected XY coordinates. The final 3D OBB combines the PCA box extents with the Z-range of the transformed inliers.

Final Alignment Composition. The three stages compose into a single rigid transformation:

$$\mathbf{x}_{\text{canonical}} = \mathbf{T}_{\text{XY}} \circ \mathbf{T}_\delta \circ \mathbf{T}_{\text{auto}}(\mathbf{x}_{\text{world}}), \tag{25}$$

where $\mathbf{T}_{\text{auto}}$ is from Eq. 22, $\mathbf{T}_\delta$ is the manual correction, and $\mathbf{T}_{\text{XY}}$ is the XY-plane alignment. The composed transform is cached per video.

G.2. Correcting World-Level 3D BBox Annotations

With the gravity-aligned frame established (Sec. G.1), 3D bounding boxes are lifted into world coordinates. The automatic annotations often exhibit errors (incorrect dimensions, missing objects, misaligned orientations, wrong labels), so we provide the *WorldBBox Viewer* for interactive 3D correction.

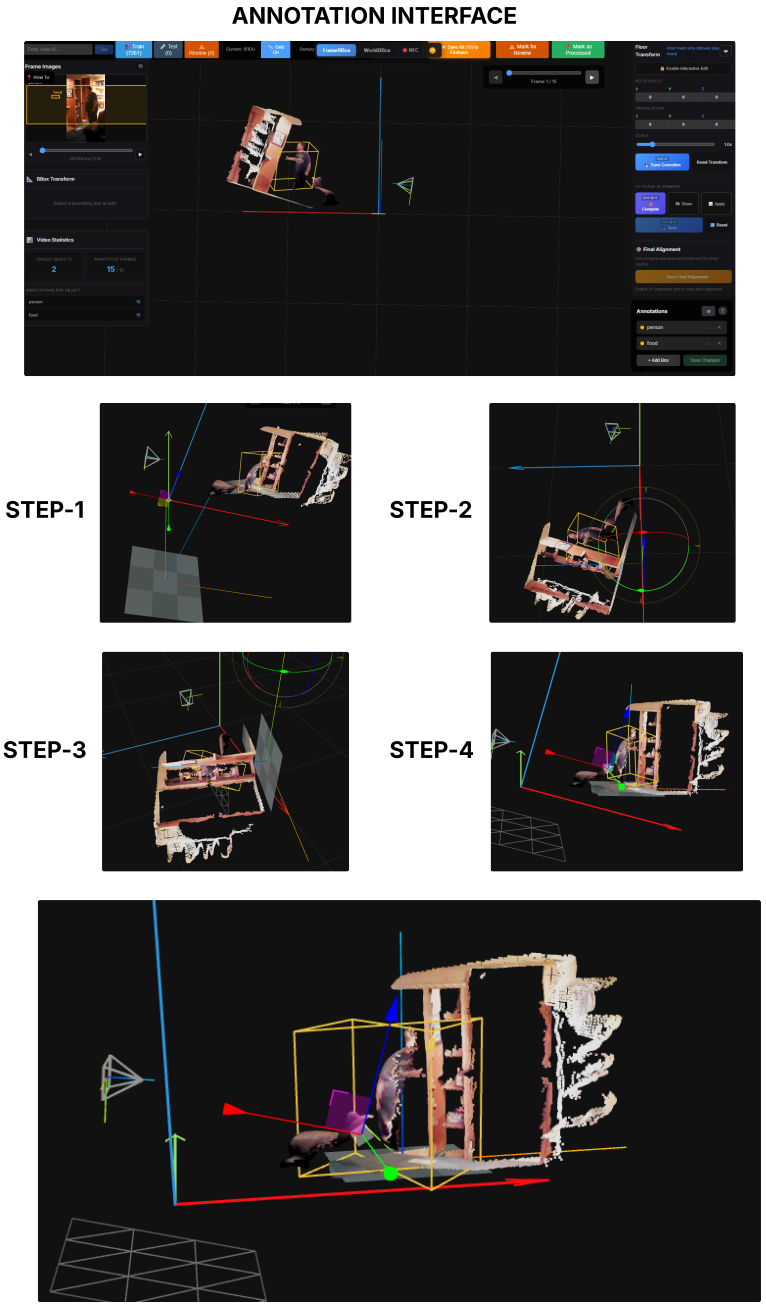

Figure 7. Overview of the annotation interface used for manual floor correction. The figure illustrates the sequential adjustment process across four steps, culminating in the final corrected transform for accurate 3D scene alignment.

Automatic World-Level BBox Generation. Initial world-level 4D annotations are generated automatically. Objects are classified as *static* (e.g., `table`, `sofa`) or *dynamic* (e.g., `person`); for static objects, missing frames are filled from the nearest known frame (object permanence). Each per-frame box ($\mathbf{C}_{\text{world}} \in \mathbb{R}^{8\times 3}$) is transformed to the gravity-aligned frame:

$$\mathbf{C}_{\text{final}} = (\mathbf{R}_{\text{final}} \cdot \mathbf{C}_{\text{world}}^{\top})^{\top} + \boldsymbol{\tau}_{\text{final}}^{\top}. \tag{26}$$

For each static label, a union bounding box is computed across all frames:

$$\mathbf{C}_{\text{union}} = \text{OBB}\left(\bigcup_{f \in \text{frames}} \mathbf{C}_{\text{final}}^{(\ell,f)} \right), \tag{27}$$

ensuring spatial consistency while preserving the original world-space boxes.

Interactive Correction Interface The WorldBBox Viewer provides a 3D canvas (via Three.js/React Three Fiber) rendering the floor mesh, reconstructed point cloud, bounding box wireframes with floating text labels, and relationship arcs connecting subject-predicate-object triples. The viewer supports orbital camera controls and frame-by-frame navigation. For each selected bounding box, the annotator can apply translate, rotate, and scale transforms via sliders with live preview. Transforms are applied sequentially (scale $\rightarrow$ rotate $\rightarrow$ translate about box center) and committed on confirmation, with an undo stack for reverting. The annotations panel additionally supports inline label editing, per-box visibility toggles, and adding/deleting boxes.

Temporal Propagation of Annotations A key challenge in world-level annotation is ensuring that object bounding boxes are consistent across all frames. The Propagation Panel provides tools to copy a bounding box annotation from the current frame to other frames: The annotator selects a bounding box and specifies: (1) **Direction**: Forward (to subsequent frames), backward (to preceding frames), or both. (2) **Frame Count**: A specific number of frames to propagate to, or "All" to propagate to the video's start/end. Upon clicking **Propagate**, the selected box's label, corners, center are duplicated to each target frame.

H. Manual Correction of WSG Annotations

H.1. Overview

The pseudo-annotations produced by the MLLM pipeline contain errors from hallucination, ambiguous visual context, or incorrect grounding. To produce gold-standard annotations, we deploy a *human-in-the-loop* manual correction stage using a purpose-built web annotation tool. Trained annotators systematically review and correct every pseudo-annotated relationship across three semantic axes: **Attention**, **Contacting**, and **Spatial**. The correction is designed as a *refinement* workflow: annotators start from MLLM predictions rather than labeling from scratch, significantly reducing annotation time.

H.2. Annotation Interface

The tool follows a two-panel layout. The *left panel* provides a high-fidelity frame viewer with:

- Full-resolution frame display with a slider for scrubbing through all video frames.
- An HTML5 canvas overlay rendering person bounding boxes (green) and object bounding boxes (15 distinct colors) with class-name labels.
- A lightbox modal for pixel-level inspection with zoom (0.5–5$\times$) and pan.

The *right panel* serves as the annotation workspace:

- **Video selection**: three dropdown selectors group videos by status (Pending / Review / Processed).
- **Object accordions**: each missing object is presented as a collapsible section showing the object name, frame count, and per-category correction statistics.
- **Annotation grid**: a table of frames $\times$ relationship categories. Each cell contains toggle buttons for every possible label. Attention is single-select; Contacting and Spatial are multi-select. Active labels are color-coded with abbreviated display names.

H.3. Correction Workflow

Per-Video Workflow. The annotator (1) selects a pending video, (2) inspects frames using the frame viewer and bounding-box overlay, (3) corrects each missing object's relationships by toggling labels in the annotation grid, (4) saves corrections, and (5) marks the video as Processed or flags it for Review.

Propagation Tools. To reduce repetitive annotation when relationships are stable across frames: (1) Copy Forward ($\rightarrow$): copies one frame's complete annotation to all subsequent frames for the same object. (2) Apply to All: broadcasts the first frame's annotations to all frames.

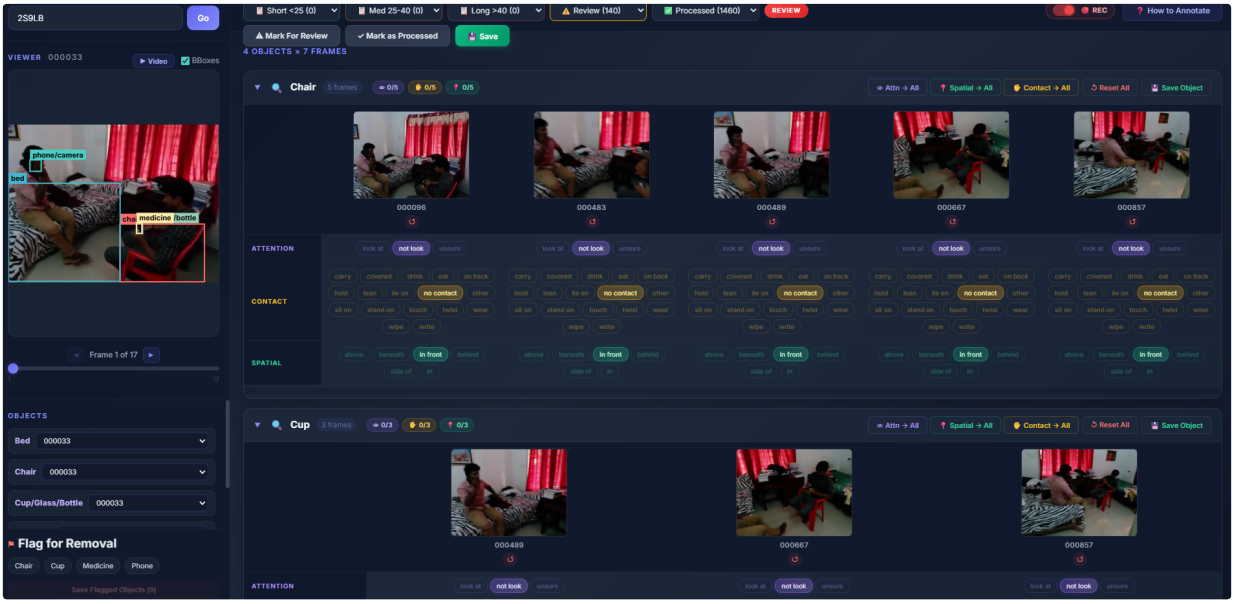

Figure 8. Interface for manual relationship correction in WorldSGG. The tool presents object-centric frame sequences and allows annotators to refine attention, contact, and spatial predicates across time for accurate scene graph construction.

H.4. Quality Tracking

The interface displays real-time correction statistics to monitor progress. For each missing object o across its N frame appearances, the system computes per-category correction counts by comparing compiled (post-edit) predictions against base (MLLM) predictions:

$$n^{o}_{\text{cat}} = \sum_{f=1}^{N} \mathbb{1}\big[F^{(f,o)}_{\text{comp}}.\texttt{cat} \neq F^{(f,o)}_{\text{base}}.\texttt{cat}\big], \quad \text{cat} \in \{\text{att}, \text{con}, \text{spa}\} \tag{28}$$

These counts are displayed as colored badges (n/N) in each object accordion header, providing at-a-glance visibility into annotator modifications.

H.5. Status Management

Videos follow a three-state lifecycle: **Pending** (uploaded, not yet reviewed), **Processed** (corrections complete), and **Review** (flagged for a second pass due to ambiguity). Status transitions are triggered explicitly by annotators and stored with timestamps in Firebase, enabling dataset-level progress tracking via the dropdown counters in the interface header.

I. Action Genome 4D Statistics

ActionGenome4D (AG4D) extends the original Action Genome [29] dataset into a 4D spatio-temporal representation by augmenting every video with (i) per-frame 3D scene reconstructions and camera poses, (ii) world-frame oriented 3D bounding boxes (OBBs) for all annotated objects, and (iii) dense semantic relationship annotations covering both *observed* and *unobserved* objects at every annotated timestamp. This section summarises the key statistics of the resulting dataset and contrasts them with the original Action Genome annotations.

Dataset Overview. Action Genome provides frame-level 2D bounding-box annotations and human–object relationship labels for videos sourced from the Charades dataset [74]. AG4D preserves the original train/test split and enriches every video along three axes:

1. **Geometric:** Per-frame 3D point clouds reconstructed via π^3 [87], floor-aligned via SMPL-based similarity estimation, and enclosed by oriented 3D bounding boxes (OBBs) in a shared world coordinate frame (Section E.1).
2. **Semantic:** Relationship annotations expanded from observed-only objects ($\mathcal{O}^t$) to the full world state ($\mathcal{W}^t = \mathcal{O}^t \cup \mathcal{U}^t$), covering attention, spatial, and contacting predicates for every (person, object) pair at every annotated frame (Section F.1).
3. **Camera:** Camera-to-world SE(3) poses $\{\mathbf{T}^t\}_{t=1}^{T}$ refined via iterative bundle adjustment, providing ego-

Table 8. **AG vs. ActionGenome4D.** AG4D extends AG along geometric and semantic dimensions. $\mathcal{O}^t$: observed objects; $\mathcal{U}^t$: unobserved objects; $\mathcal{W}^t$: full world state.

Property	Action Genome	ActionGenome4D
Scope		
Object scope per frame	$\mathcal{O}^t$ only	$\mathcal{W}^t = \mathcal{O}^t \cup \mathcal{U}^t$
Relationship coverage	Observed pairs	All world-state pairs
Coordinate frame	2D image	3D world
Object localization	2D BBox	2D BBox + 3D OBB
Camera poses	✗	✓ (SE(3) per frame)
3D scene reconstruction	✗	✓ (per-frame point clouds)
Annotations		
Predicate axes	Attention, Spatial, Contacting	Attention, Spatial, Contacting
Attention labels	3 (single-label)	3 (single-label)
Spatial labels	6 (multi-label)	6 (multi-label)
Contacting labels	17 (multi-label)	17 (multi-label)
Object categories	36 (+ person)	36 (+ person)
Unobserved object labels	✗	✓
Scale		
Videos	9,250	9,250
Train / Test videos	7,516 / 1,734	7,516 / 1,734
Annotated frames	232,103	232,103

motion trajectories for each video (Section D).

Annotation Strategy. The semantic annotation strategy differs between the training and test splits to balance scalability with annotation quality:

- **Training set - Pseudo-annotations:** Relationship labels for unobserved objects are generated automatically using a VLM-based pipeline (Section F.1). A Graph-RAG approach retrieves relevant spatio-temporal context from a precomputed coarse event graph, and a discriminative verification step assigns per-label confidence scores. These pseudo-labels serve as training targets, weighted by λ_{vlm} in the loss function to account for potential noise.
- **Test set - Manual corrections:** Every pseudo-annotated relationship in the test split is reviewed and corrected by trained human annotators using a purpose-built web-based correction interface (Section H.1). The correction workflow starts from the VLM predictions and refines them via toggle-based editing, copy-forward propagation, and temporal consistency enforcement. This produces gold-standard annotations for reliable evaluation.

The geometric annotations (3D OBBs and camera poses) are generated using the same automated pipeline for both splits, followed by manual verification and correction of 3D bounding boxes via 3D annotation tool (Section G.1).

Comparison with Action Genome. Table 8 contrasts the AG dataset with ActionGenome4D across key dimensions.

Predicate Vocabulary. AG4D inherits the Action Genome predicate vocabulary, organised into three disjoint axes:

1. **Attention** ($|\mathcal{P}_{\text{att}}| = 3$, single-label): `looking_at`, `not_looking_at`, `unsure`.
2. **Spatial** ($|\mathcal{P}_{\text{spa}}| = 6$, multi-label): `above`, `beneath`, `in_front_of`, `behind`, `on_the_side_of`, `in`.
3. **Contacting** ($|\mathcal{P}_{\text{con}}| = 17$, multi-label): `carrying`, `covered_by`, `drinking_from`, `eating`, `have_it_on_the_back`, `holding`, `leaning_on`, `lying_on`, `not_contacting`, `other_relationship`, `sitting_on`, `standing_on`, `touching`, `twisting`, `wearing`, `wiping`, `writing_on`.

Attention is a single-label classification (exactly one label per object pair per frame), while spatial and contacting are multi-label (an object pair may simultaneously exhibit multiple spatial or contacting relationships, e.g., a person can be both `holding` and `touching` an object).

Object Categories. The object vocabulary consists of 36 categories (excluding `person`) inherited from Action Genome. These span common indoor objects such as furniture (`chair`, `table`, `sofa`, `bed`), personal items (`phone/camera`, `bag`, `book`, `laptop`), kitchenware (`cup/glass/bottle`, `dish`, `sandwich`), and room elements (`door`, `doorway`, `window`, `mirror`, `television`). Including `person`, the full category set comprises 37 classes used for object classification in SGDet.

Geometric Annotation Statistics. The 3D geometric annotation pipeline (Section G.2) produces world-frame oriented 3D bounding boxes for every annotated object across all videos. Key statistics of the geometric annotations are:

- **Total 3D OBBs:** 828,213 OBBs across all videos and frames.
- **OBB fitting:** Each OBB is selected as the minimum-volume candidate from a multiscale erosion sweep ($\{0, 3, 5, 7, 10\}$ px kernels), with temporal smoothing via Kalman filtering and RTS backward pass.
- **Coordinate system:** All OBBs are expressed in the final coordinate system (floor-aligned, metrically scaled in meters, world-frame centered), alongside per-frame camera poses.

Semantic Annotation Statistics. AG4D extends the original AG relationship annotations from observed-only objects to the full world state. The key distinction is that every (person, object) pair receives relationship labels at every annotated timestamp, regardless of objects' visibility.

- **Total relationship instances:** 602,668 relationship triplets across all frames and splits (compared to 518,895 in the original AG, which covers observed objects only).
- **Observed pairs:** 518,895 relationship instances where both the person and object are visible in the frame ($w_k^t \in \mathcal{O}^t$).
- **Unobserved pairs:** 83,773 relationship instances involving at least one unobserved object ($w_k^t \in \mathcal{U}^t$) (new in ActionGenome4D).
- **Avg. objects per frame (world state):** 3.6 objects per

Table 9. **ActionGenome4D statistics.** The train set uses VLM pseudo-annotations for unobserved object relationships; the test set uses manually corrected annotations.

Statistic	Train	Test
Videos	7,516	1,734
Annotated frames	175,751	56,352
Object instances (observed)	549,315	201,683
Object instances (unobserved)	61,149	22,624
Object instances (total)	610,464	224,307
Relationship triplets (observed)	373,564	145,331
Relationship triplets (unobserved)	61,149	22,624
Relationship triplets (total)	434,713	167,955
3D OBBs	604,128	224,085
Annotation type (unobserved)	VLM pseudo-labels	Manual corrections

frame in $\mathcal{W}^t$, compared to 3.2 observed objects per frame in the original AG.

- **Avg. relationships per frame:** 2.6 (person, object) relationship instances per frame in $\mathcal{G}^t_{\mathcal{W}}$, compared to 2.2 in AG's observed-only $\mathcal{G}^t$.

Train/Test Split Statistics. Table 9 summarises the per-split statistics.

Predicate Distribution. The predicate label distributions in AG4D exhibit long-tail characteristics consistent with real-world relationship frequencies. For the **attention** axis, `not_looking_at` dominates (especially for unobserved objects, where the person is typically not attending to off-screen objects), followed by `looking_at` and `unsure`. For the **contacting** axis, `not_contacting` is the most frequent label (again driven by unobserved objects with no physical contact), while labels such as `holding`, `touching`, and `sitting_on` cover the active interactions. For the **spatial** axis, `in_front_of` and `on_the_side_of` are common due to typical indoor scene layouts, while `beneath` and `in` are relatively rare. The introduction of unobserved object annotations significantly amplifies the representation of `not_looking_at` and `not_contacting`, since most unobserved objects are neither attended to nor in physical contact with the person. This shifts the overall predicate distribution compared to the original AG and motivates the use of label smoothing and weighted losses during training.

Table 10. Supported backbone variants.

Key	Model	Hidden dim	Params	Patch size
`v2`	DINOv2 ViT-B	768	86M	14
`v2s`	DINOv2 ViT-S	384	22M	14
`v2l`	DINOv2 ViT-L	1024	304M	14
`v3l`	DINOv3 ViT-L	1024	304M	16

J. Monocular 3D Detection Pipeline

The monocular 3D object detector provides the geometric scaffolding for WorldSGG. Given a single RGB frame, it simultaneously produces 2D bounding boxes with class labels and 8-corner oriented 3D bounding boxes (OBBs) in camera coordinates. The 3D boxes are then lifted into the persistent world frame via extrinsic camera matrices, providing the structural prior consumed by all three WorldSGG methods (PWG, 4DST, MWAE). The system follows a Faster R-CNN [68] architecture with three key innovations: (i) a **frozen DINOv2/v3 ViT backbone** [63] paired with a learnable Simple Feature Pyramid Network (SimpleFPN) [42]; (ii) a **factorized 3D prediction head** decomposing 3D box regression into dimensions, rotation, depth, and center offset, reconstructed via pinhole back-projection; (iii) an **OVMono3D-style disentangled 3D loss** [96] with per-sample uncertainty weighting and Chamfer supervision.

J.1. Model Architecture

The full pipeline is: Image $\xrightarrow{\text{Transform}}$ Backbone $\xrightarrow{\text{FPN}}$ RPN $\xrightarrow{\text{ROI Heads}}$ 2D Detections + 3D Boxes.

Backbone: Frozen DINOv2/v3 ViT. The backbone is a Vision Transformer (ViT) [15] pretrained with DINOv2/v3 [63], kept entirely **frozen** during training. Four variants are supported (Table 10). Only the last $N_{\text{patch}} = H_p \times W_p$ spatial tokens are kept (stripping CLS and register tokens), reshaped to $\mathbf{F} \in \mathbb{R}^{B\times C\times H_p\times W_p}$.

Simple Feature Pyramid Network (SimpleFPN). Following ViTDet [42], the SimpleFPN converts the single-scale backbone output into a multi-scale pyramid at strides $\{P/4, P/2, P, 2P, 4P\}$ via transposed convolutions (upsampling) and max-pooling (downsampling), where P is the patch size. Each level is projected to 256 channels by a 1×1 convolution, BatchNorm, ReLU, and a 3×3 refinement convolution. All FPN parameters are learnable.

Region Proposal Network and ROI Heads. The RPN uses standard Faster R-CNN anchors of sizes $\{32, 64, 128, 256, 512\}$ at pyramid levels $\{p2, \ldots, p6\}$ with aspect ratios $(0.5, 1.0, 2.0)$, contributing objectness and box-regression losses. Proposals are processed by MultiScaleRoIAlign (7×7 output, sampling ratio 2) feeding two FC layers that produce a shared 1024-d feature, from which

parallel heads predict class logits and box deltas.

Factorized 3D Prediction Head. The 3D head is a lightweight branch sharing the 1024-d ROI features. It concatenates the shared features with normalized 2D box coordinates and camera intrinsics, passed through a shared context FC (ReLU, 512-d output). Five parallel heads produce:

$$\hat{\mathbf{d}} = \text{softplus}(\cdot) + \epsilon \in \mathbb{R}^3, \quad \text{(dimensions: } l, w, h\text{)} \tag{29}$$

$$[\hat{s}, \hat{c}] = \text{L2-normalize}(\cdot) \in \mathbb{R}^2, \quad \text{(rotation: } \sin\theta, \cos\theta\text{)} \tag{30}$$

$$\hat{z}_c = \text{softplus}(\cdot) + \epsilon \in \mathbb{R}^1, \quad \text{(depth)} \tag{31}$$

$$\hat{\delta} \in \mathbb{R}^2, \quad \text{(2D center offset)} \tag{32}$$

$$\hat{\mu} \in \mathbb{R}^1, \quad \text{(uncertainty)} \tag{33}$$

where $\epsilon = 10^{-4}$ ensures strictly positive outputs.

Pinhole Back-Projection. The 3D camera-frame center is recovered by intersecting the offset 2D center with the predicted depth via the pinhole model:

$$X_c = \frac{(u - c_x)}{f_x} \cdot \hat{z}_c, \quad Y_c = \frac{(v - c_y)}{f_y} \cdot \hat{z}_c, \quad Z_c = \hat{z}_c, \tag{34}$$

where $u = c_{x,\text{2d}} + \hat{\delta}_x$, $v = c_{y,\text{2d}} + \hat{\delta}_y$. The 8 corners are built at $(\pm l/2, \pm w/2, \pm h/2)$, rotated by the predicted yaw, and translated to $[X_c, Y_c, Z_c]^\top$.

Weight Initialization. Default initialization causes extreme variance in the initial 3D loss. We apply targeted initialization: Xavier-uniform for the context FC, $\mathcal{N}(0, 0.001)$ weights with sensible biases for depth (initial $\approx$1.3–1.8 m), dimensions (initial $\approx$0.7 m), and identity rotation ($\sin=0, \cos=1$).

J.2. OVMono3D Disentangled 3D Loss

The 3D loss follows OVMono3D [96], combining uncertainty-weighted supervision with geometry-level disentanglement.

Disentangled Attribute Losses. Given predicted corners $\hat{\mathbf{C}}$ and GT corners $\mathbf{C}^* \in \mathbb{R}^{8\times3}$, both are decomposed via PCA into center $\mathbf{p}$, oriented dimensions $\mathbf{d} = (l, w, h)$, and yaw θ. For each attribute group $a \in \{xy, z, \text{dims}, r\}$, a "mixed" box uses the predicted attribute with GT values for all others:

$$\mathcal{L}_{xy}: \quad \mathbf{C}_{xy} = \text{build}(\hat{p}_{xy} \| p^*_z, \mathbf{d}^*, \theta^*), \tag{35}$$

$$\mathcal{L}_{z}: \quad \mathbf{C}_{z} = \text{build}(p^*_{xy} \| \hat{p}_z, \mathbf{d}^*, \theta^*), \tag{36}$$

$$\mathcal{L}_{\text{dims}}: \quad \mathbf{C}_{d} = \text{build}(\mathbf{p}^*, \hat{\mathbf{d}}, \theta^*), \tag{37}$$

$$\mathcal{L}_{r}: \quad \mathbf{C}_{r} = \text{build}(\mathbf{p}^*, \mathbf{d}^*, \hat{\theta}). \tag{38}$$

Table 11. Key training configuration parameters.

Parameter	Default	Description
`model`	`v2`	Backbone variant (Table 10)
`lr`	10^{-4}	Peak learning rate
`weight_decay`	10^{-3}	AdamW weight decay
`batch_size`	128	Per-GPU batch size
`epochs`	70	Total training epochs
`max_grad_norm`	1.0	Gradient clipping threshold
`head_3d_mode`	`unified`	3D head integration (unified or separate)
`head_3d_version`	`v1`	3D head architecture version
`weight_3d`	1.0	Target 3D loss weight
`weight_3d_ramp_epochs`	5	Staged ramp duration (R)
`pixel_limit`	255,000	Max pixels for resize

An additional holistic loss $\mathcal{L}_{\text{all}} = \text{Chamfer}_{\text{SL1}}(\hat{\mathbf{C}}, \mathbf{C}^*)$ compares the full corners directly. All Chamfer computations use Smooth L1 distance for stability, and values are divided by the GT box diagonal for scale invariance.

Per-Sample Uncertainty Weighting. The uncertainty $\hat{\mu}_i$ is STE-clamped to $[-5, 10]$ and per-sample $\mathcal{L}_{\text{3D},i}$ is clamped at 100. The total per-sample 3D loss is:

$$\mathcal{L}_i = \sqrt{2} \cdot \exp(-\hat{\mu}_i) \cdot \mathcal{L}_{\text{3D},i} + \hat{\mu}_i, \quad \mathcal{L}_{\text{3D},i} = \mathcal{L}_{xy,i} + \mathcal{L}_{z,i} + \mathcal{L}_{\text{dims},i} + \mathcal{L}_{r,i} + \mathcal{L}_{\text{all}} \tag{39}$$

Total Training Loss. The total loss combines all Faster R-CNN losses and the 3D loss:

$$\mathcal{L}_{\text{total}} = w_{\text{cls}}\mathcal{L}_{\text{cls}} + w_{\text{box}}\mathcal{L}_{\text{box}} + w_{\text{obj}}\mathcal{L}_{\text{rpn-obj}} + w_{\text{rpn}}\mathcal{L}_{\text{rpn-box}} + w_{\text{3d}}(e) \cdot \mathcal{L}_{\text{3D}}, \tag{40}$$

where all weights default to 1.0. To stabilize early training, $w_{\text{3d}}(e)$ follows a three-phase ramp: zero for the first R epochs (2D-only), linearly increasing from R to $2R$, and full weight thereafter (R=5 by default).

J.3. Dataset and Training

Data Sources. The dataset loads 2D annotations (37 object classes) from standard Action Genome pickles, 3D annotations (8-corner OBBs in camera frame with per-video intrinsics) from per-video pickle files. Images are resized to an aspect-ratio-preserving resolution where both dimensions are multiples of patch size P, with total pixels $\leq$255K. Camera intrinsics are scaled proportionally. Images are grouped into *resolution buckets* sharing the same target size, enabling padding-free batching.

Training Configuration. Training uses AdamW [48] (lr=10^{-4}, wd=10^{-3}) with linear warmup (1% of steps) followed by cosine annealing. Mixed precision is enabled via `GradScaler`, with gradients clipped at norm 1.0. Non-finite losses trigger batch skipping. CUDA optimizations include cuDNN auto-tuning, TF32 arithmetic, and Flash Attention. Key hyperparameters are listed in Table 11.

Evaluation. Evaluation uses a single fused forward pass over the full test set, computing 2D and 3D metrics simultaneously.

Table 12. Comparison of DINO backbones under joint and separate training setups.

Backbone	Training	mAP	mAP_{50}	mAP_{75}
DINOv2-B	Separate	0.0988	0.2333	0.0692
DINOv2-L	Separate	0.1103	0.2627	0.0739
DINOv3-L	Separate	0.1762	**0.3667**	0.1461
DINOv2-B	Joint	0.0998	0.2377	0.0672
DINOv2-L	Joint	0.1086	0.2610	0.0712
DINOv3-L	Joint	**0.1799**	**0.3660**	**0.1552**

2D Metrics. We report COCO-style mAP over IoU thresholds $[0.50, 0.55, \ldots, 0.95]$, along with mAP_{50} and mAP_{75}.

3D Metrics. 3D evaluation is performed on matched prediction–GT pairs (matched via 2D IoU ≥ 0.5, greedy, same-class). Box-level metrics include bidirectional Chamfer distance, corner L2, and oriented 3D IoU (via Sutherland–Hodgman polygon clipping [79]). Attribute-level metrics include center L2, dimensions L1, and wrapped rotation error. IoU hit rates at 50% and 75% measure 3D regression quality for correctly detected objects.

Backbone and Training Mode Comparison. We evaluate the impact of backbone choice and training mode on 2D detection quality. Specifically, we compare three backbones; DINOv2-B (ViT-B/14, 86M), DINOv2-L (ViT-L/14, 304M), and DINOv3-L (ViT-L/16, 304M) under two training setups:

- **Separate**: The 2D detector and 3D head are trained independently; the 3D branch does not influence 2D detection gradients.
- **Joint**: Both 2D and 3D heads are trained end-to-end with the staged loss ramp (Eq. 40), allowing 3D supervision to refine shared representations.

Results are shown in Table 12. Several observations stand out.

DINOv3-L significantly outperforms DINOv2 variants. DINOv3-L achieves mAP of 0.18, a 60% relative improvement over DINOv2-L (0.11) and 80% over DINOv2-B (0.10). This gap is consistent across all IoU thresholds, with DINOv3-L reaching 0.37 mAP_{50} versus 0.26 for DINOv2-L. The improvement is attributable to DINOv3's stronger spatial representations and enhanced training recipe, which yield features better suited for localization.

Scaling from Base to Large within DINOv2 yields marginal gains. DINOv2-L improves over DINOv2-B by only $\sim$1 mAP point (0.11 vs. 0.10), suggesting that the DINOv2 representation quality—rather than model capacity—is the bottleneck. In contrast, the architecture and pretraining changes introduced in DINOv3 unlock substantially better performance.

Table 13. 3D localization performance of different DINO backbones under joint and separate training setups. Mean IoU_{3D} measures the average 3D overlap, while Hit@0.50 denotes the fraction of predictions exceeding the IoU_{3D} threshold.

Backbone	Training	Mean IoU_{3D}	Hit@0.50
DINOv2-B	Separate	**0.1446**	**0.0137**
DINOv2-L	Separate	0.1377	0.0117
DINOv3-L	Separate	0.1295	0.0111
DINOv2-B	Joint	0.1411	0.0120
DINOv2-L	Joint	0.1390	0.0104
DINOv3-L	Joint	0.1285	0.0128

Table 13 complements the 2D analysis with 3D localization quality.

Numerical Stability Notes. All 3D loss computation is forced to float32 to avoid Chamfer overflow in float16. The 3D loss operates on at most 64 positive proposals per batch, randomly subsampled, to bound memory under the $5\times$ disentangled expansion. Resolution bucketing reads only JPEG/PNG headers ($\leq$32 bytes) for the $\sim$7K per-video dimension queries.

K. World Scene Graph Generation

K.1. Baseline Architectures

All four baselines share a common infrastructure: a FasterRCNN-ResNet50 backbone for monocular 3D detection, an LKS (Last Known State) memory buffer for non-differentiable visual persistence, a GlobalStructuralEncoder that converts oriented bounding box (OBB) corners into per-object 3D structural tokens, an InterObjectTransformer for intra-frame spatial reasoning with continuous pairwise 3D positional encodings, and a three-head relationship predictor (attention, spatial, contacting). They differ in which optional temporal and spatial modules are activated, forming a controlled ablation ladder. Table 14 summarizes the module activation pattern.

Table 14. Module activation across baselines. ✓ = active, – = absent.

Module	W-STTran	W-STTran++	W-DsgDetr	W-DsgDetr++
LKS Buffer + GlobalStructuralEncoder	✓	✓	✓	✓
ObjectSpatialEncoder	–	✓	–	✓
ObjectMotionEncoder + MotionFusion	–	✓	–	✓
CameraTemporalEncoder	–	–	–	✓
TemporalObjectEncoder	–	–	✓	✓
TemporalEdgeAttention	–	✓	✓	✓

W-STTran. W-STTran is the minimal baseline, adapting STTran to the WorldSGG task. Object features extracted by the FasterRCNN backbone are held in the LKS buffer, which returns the nearest visible frame's feature $\mathbf{m}_n^{(t)}$ for occluded objects and a staleness signal $\Delta_n^{(t)}$. The GlobalStructuralEncoder converts each object's 8-corner OBB $\mathbf{C}_n \in \mathbb{R}^{8\times 3}$ into a centered, translation-invariant structural token $\mathbf{g}_n$ via a shared MLP. The LKS Tokenizer fuses geometry, buffered features, and log-staleness through a linear projection with LayerNorm: $\mathbf{x}_n^{(t)} = \mathrm{FusionProj}([\mathbf{g}_n^{(t)} \parallel \mathbf{m}_n^{(t)} \parallel \log(\Delta_n^{(t)} + 1)])$. The InterObjectTransformer then performs global spatial reasoning using additive pairwise positional encodings derived from 3D inter-object distances and volume ratios. A Node Predictor classifies objects, and the Relationship Predictor forms edge tokens from subject/object features, union RoI features, and CLIP semantic embeddings, refines them via self-attention, and outputs predictions through three independent MLP heads. W-STTran uses *no* camera, motion, or temporal modules, isolating the contribution of passive 3D memory alone.

W-STTran++. W-STTran++ augments the minimal baseline with camera-relative and motion-aware representations, but still omits explicit temporal object tracking. The ObjectSpatialEncoder computes per-object camera-relative features from the extrinsic matrix $\mathbf{T} = [\mathbf{R} \,|\, \boldsymbol{\tau}]$: log-distance, view alignment $\alpha_n = \hat{\mathbf{d}}_n \cdot (-\mathbf{R}_{:,2})$, and azimuth components, projected via MLP to $\mathbf{c}_n \in \mathbb{R}^{d_{\text{camera}}}$. The ObjectMotionEncoder captures 3D dynamics by computing world-frame velocity $\mathbf{v}_n^{(t)}$ and acceleration $\mathbf{a}_n^{(t)}$ from OBB center finite differences, alongside camera-relative velocity $\mathbf{R}^\top \mathbf{v}_n$, yielding an 11-dimensional feature projected to $\mu_n^{(t)}$. The Tokenizer concatenates geometry, buffer, spatial, and staleness features; the resulting token is then fused with motion via $\hat{\mathbf{x}}_n^{(t)} = \mathrm{LayerNorm}(\mathrm{Linear}([\mathbf{x}_n^{(t)} \parallel \mu_n^{(t)}]))$. After InterObjectTransformer reasoning, TemporalEdgeAttention performs cross-frame self-attention over each unique object pair's relationship tokens with learned temporal positional embeddings, enabling temporal consistency without explicit per-object tracking.

W-DsgDetr. W-DsgDetr adapts DSG-DETR to WorldSGG by introducing explicit per-object temporal tracking. It shares the LKS buffer, GlobalStructuralEncoder, and Tokenizer with W-STTran (without camera or motion features). The key addition is the TemporalObjectEncoder: for each object n, it collects the T-length sequence of tokenized representations and processes them with a Transformer encoder using sinusoidal temporal positional encodings: $\tilde{\mathbf{X}}_n = \mathrm{TransformerEncoder}(\{\mathbf{x}_n^{(t)} + \mathbf{PE}_{\text{sin}}(t)\}_{t=1}^{T})$. This allows the model to interpolate and contextualize object features across time, addressing occlusion gaps that the zero-order LKS buffer cannot. The temporally-refined tokens then undergo InterObjectTransformer spatial reasoning, followed by node prediction, relationship formation with intra-frame self-attention, and TemporalEdgeAttention for cross-frame edge consistency. This design isolates the contribution of explicit temporal tracking relative to W-STTran.

W-DsgDetr++. W-DsgDetr++ activates every available module, representing the most feature-rich baseline before WorldWise. Beyond W-DsgDetr's TemporalObjectEncoder, it adds the ObjectSpatialEncoder, ObjectMotionEncoder, and a CameraTemporalEncoder that models ego-motion dynamics. The CameraTemporalEncoder computes relative camera poses between consecutive frames ($\mathbf{R}_{\text{rel}}^{(t)} = \mathbf{R}_t \mathbf{R}_{t-1}^\top$, $\boldsymbol{\tau}_{\text{rel}}^{(t)} = \boldsymbol{\tau}_t - \mathbf{R}_{\text{rel}}^{(t)} \boldsymbol{\tau}_{t-1}$) and encodes the sequence via self-attention to produce per-frame ego-motion tokens $\mathbf{e}_t$. The MotionFusion stage integrates object motion and ego-motion: $\hat{\mathbf{x}}_n^{(t)} = \mathrm{LayerNorm}(\mathrm{Linear}([\mathbf{x}_n^{(t)} \parallel \mu_n^{(t)} \parallel \mathbf{e}_t]))$. The full pipeline then proceeds through TemporalObjectEncoder, InterObjectTransformer, and TemporalEdgeAttention. This configuration tests the upper bound of performance achievable within the passive-memory (LKS) paradigm.

Training Objective. All four baselines share a common loss function that partitions human-object pairs into *visible* (both entities observed) and *unseen* (at least one occluded) buckets. Visible pairs use standard cross-entropy (attention) and binary cross-entropy (spatial, contacting) with clean ground truth. Unseen pairs employ VLM pseudo-labels with label smoothing and a weighting factor λ_{vlm}:

$$\mathcal{L} = \sum_{r \in \{\text{att, spa, con}\}} \left(\mathcal{L}_r^{\text{vis}} + \mathcal{L}_r^{\text{unseen}}\right) + \mathcal{L}_{\text{node}}. \tag{41}$$

L. MLLMs for Unlocalized WSG Generation

This section assesses how well current Vision-Language Models (VLMs) perform on the task of *Unlocalized World Scene Graph Generation* (U-WSGG).

While the full WSGG task requires 3D-localised detection and tracking (addressed by WorldWise), the *unlocalized* variant (U-WSGG) isolates the core *semantic reasoning* challenge: given a known set of objects, predict their relationships to the person at each frame across attention, spatial, and contacting axes. Modern VLMs can be directly prompted for relationship prediction, but vanilla zero-shot querying suffers from *attention dilution* over long video contexts and the lack of *query-specific temporal grounding*. We propose **UWSGG-GRAPHRAG**, a Graph-structured Retrieval-Augmented Generation strategy (Figure 4) that constructs a video-level knowledge graph offline and uses it as a structured retrieval index, enabling query-specific context selection before the VLM call.

L.1. Task Definition

Given a video V with K annotated frames $\{f_1, \ldots, f_K\}$ and a video-level object set $\mathcal{O}_v$, the goal is to predict, for *every* annotated frame f_k and *every* object $o \in \mathcal{O}_v$, the three relationship types between the person and object o:

$$\hat{R}(f_k, o) = \left(\hat{a}_{k,o},\ \hat{C}_{k,o},\ \hat{S}_{k,o}\right) \tag{42}$$

where $\hat{a}$ is the predicted attention label (single-label), $\hat{C}$ is the predicted set of contacting labels (multi-label), and $\hat{S}$ is the predicted set of spatial labels (multi-label).

Object Set Resolution: PredCls vs. SGDet. The evaluation supports two standard modes:

PredCls (Predicate Classification). Ground-truth object labels from the annotations are used directly as the video-level object set $\mathcal{O}_v$. This isolates the relationship prediction ability of the VLM.

SGDet (Scene Graph Detection). Objects are *estimated* from the video content using the VLM itself. The estimation proceeds as follows:

1. **Caption-guided estimation.** The VLM is prompted with all available captions plus the canonical AG object vocabulary (36 classes) as a candidate list. It returns a JSON list of object name strings.
2. **Vocabulary filtering.** The estimated object set is intersected with the canonical AG vocabulary to ensure all labels conform to the closed label space: $\mathcal{O}_{\text{est}} = \hat{\mathcal{O}} \cap \mathcal{V}$.
3. **Discriminative verification.** For each estimated object $o \in \mathcal{O}_{\text{est}}$, a binary Yes/No verification query is sent to the VLM: "*Is there a* $\{o\}$ *in this scene? Answer only Yes or No.*" Using batched single-token inference with

log-probability extraction, a confidence score $p_{\text{yes}}(o)$ is obtained for every object.

L.2. Evaluation Protocol and Metrics

Matching Procedure. For each frame and each object, predictions are matched against ground truth by exact object name matching. For objects present in the GT but absent from the VLM predictions (or vice versa), the corresponding entry is recorded as a miss (false negative) or false positive, respectively.

Metrics. We compute the following metrics for each relationship axis: For each label l in the vocabulary of an axis (e.g., each of the 3 attention labels, 17 contacting labels, or 6 spatial labels):

$$\text{Precision}(l) = \frac{|\text{TP}(l)|}{|\text{TP}(l)| + |\text{FP}(l)|} \tag{43}$$

$$\text{Recall}(l) = \frac{|\text{TP}(l)|}{|\text{TP}(l)| + |\text{FN}(l)|} \tag{44}$$

$$\text{F1}(l) = 2 \cdot \frac{\text{Precision}(l) \cdot \text{Recall}(l)}{\text{Precision}(l) + \text{Recall}(l)} \tag{45}$$

Macro-averaged and micro-averaged F1. Macro-average computes the mean F1 across all labels; micro-average computes precision and recall over all instances globally.

Attention accuracy. Since attention is single-label, we additionally report top-1 accuracy.

Verification-aware metrics. When verification scores are available (p_{yes}), we evaluate at multiple confidence thresholds $\tau \in \{0.3, 0.5, 0.7, 0.9\}$ to generate precision–recall trade-off curves. A predicted label is accepted only if $p_{\text{yes}} \geq \tau$.

L.3. Method 1: Caption-Based Generation

The caption-based method provides a lighter-weight approach that bypasses graph construction, keyword extraction, and node retrieval entirely.

Caption-Enriched Prompting. VLM-generated natural-language descriptions of the video content are used to augment the relationship prompt. Captions $\mathcal{S} = \{(i_s, t_s)\}$ are loaded from the offline Phase-0 caption generation pipeline, where each (i_s, t_s) pairs an annotated frame index i_s with a text summary t_s of the corresponding video clip.

For each annotated frame f_k, captions are sorted by temporal proximity and formatted into a context string:

$$\textsc{SubCtx}(\mathcal{S}, f_k) = \text{format}\big(\text{sort}(\mathcal{S},\ \text{key} = |i_s - k|)\big), \tag{46}$$

Algorithm 3 U-WSGG-SUB: Caption-Enriched VLM Querying

Require: Video V, frames $\mathcal{F}$, objects $\mathcal{O}_v$, captions $\mathcal{S}$
Ensure: Predictions $\{\hat{R}(f_k, o_j)\}_{k,j}$

$\mathbf{T}_V \leftarrow \textsc{LoadVideo}(V)$
$\mathbf{A} \leftarrow \textsc{BuildAnnotatedCtx}(\mathcal{F}, V)$
$\mathcal{S} \leftarrow \textsc{LoadCaptions}(V)$ ▷ From Phase-0 pickle
for $f_k \in \mathcal{F}$ **do**
  $\mathbf{v}_k \leftarrow f_k \oplus \mathbf{A}$
  $s_k \leftarrow \textsc{SubCtx}(\mathcal{S}, f_k)$ ▷ Proximity-sorted
end for
$\mathcal{Q} \leftarrow \emptyset$
for $f_k \in \mathcal{F},\ o_j \in \mathcal{O}_v$ **do**
  $p \leftarrow s_k \oplus \textsc{RelQuery}(o_j)$ ▷ Caption + query
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(p,\ \mathbf{v}_k)\}$
end for
$\{\hat{R}\} \leftarrow \text{VLM}_{\text{batch}}(\mathcal{Q})$ ▷ 1 batched call
$\{p_{\text{yes}}\} \leftarrow \textsc{BulkVerify}(\{\hat{R}\})$

yielding entries such as "[Frame 45] A person is drinking from a cup." This context is prepended to the relationship prompt:

$$\hat{R}_{\text{cap}}(f_k, o_j) = \text{VLM}\big(\textsc{SubCtx}(\mathcal{S}, f_k) \oplus \textsc{RelQuery}(o_j),\ \textsc{VisCtx}(f_k)\big). \tag{47}$$

Annotation-Driven Per-Frame Clips. Each query uses the annotation-driven clip for its target frame. If no per-frame clip is available, the full-video tensor is used as fallback.

Caching. The caption context string is identical for all objects queried on the same frame. It is therefore computed once per frame and reused across the M object queries, reducing string formatting overhead from $K \times M$ to K operations.

Single Batched Inference. All caption-enriched prompts across all frames and all objects are collected into a single batch and processed via one batched VLM call (chunked into sub-batches of 64). Subsequently, one bulk verification call (if not skipped) scores all predictions.

Limitations. Captions provide unstructured, global scene context that does not adapt to the specific object being queried. A caption describing "the person picks up a cup and walks to the table" is equally prepended whether the query targets the cup, the table, or a chair that is out of frame. This motivates the structured, query-specific retrieval of **UWSGG-GRAPHRAG**.

L.4. Method 2: Graph RAG-Based Generation

The Graph RAG method leverages a precomputed scene graph from Phase 1 to produce semantically-grounded pseudo-annotations. Unlike the caption-based approach, Graph RAG constructs a structured retrieval index that enables *query-specific* context selection before the VLM call.

Precomputed Graph Loading. Per-clip graphs produced by Phase 1 are loaded and merged into a single video-level directed graph. Node IDs are re-indexed with offsets to avoid collisions across clips. An entity graph (inverted index mapping entity names to node sets) is reconstructed from all clip-level entities, actions, and scenes. Captions are collected as $(frame\_index, text)$ pairs. Clip intervals are also extracted to enable annotation-driven per-frame clip loading.

4-Step Query Resolution Pipeline. For efficiency, the pipeline deduplicates queries across frames since the same object prompts recur on every frame, only *unique* prompts are processed through the first three steps. The results are then broadcast back to all entries for the frame-specific final answer.

Step 1: Keyword Extraction. One batched VLM call extracts keywords from all unique query prompts. Each response contains entity names, scene descriptors, and action keywords used for graph retrieval.

Step 2: Cached Node Retrieval. Graph node retrieval uses a BGE embedding model with precomputed embeddings (computed once per video). Entity matching (cosine similarity > 0.5) and content re-ranking identify the top-$N_{\text{retrieval}} = 20$ graph nodes per query. No VLM call is required in this step.

Step 3: Templatised Node Refinement. Retrieved nodes are verified using fixed template sub-questions: (i) "*Is the '{object}' visible or interacted with in this video segment?*" (ii) "*Is there a person visible in this video segment?*" One batched VLM call processes all node-check prompts. Nodes are ranked by the number of "yes" answers.

Step 4: Frame-Specific Final Answer. Each (frame, object) pair receives its answer grounded on the *annotation-driven clip* for that specific frame (same temporal segmentation as graph construction). If no per-frame clip is available, the RAG-selected clip from node refinement is used; as a last resort, the full-video tensor serves as fallback. Graph-node context (entities, actions, scenes from the top-ranked node) is prepended to the prompt. One batched VLM call generates all relationship predictions.

Table 15. Comparison of the two U-WSGG generation methods.

Property	Caption-Based	Graph RAG-Based
Context signal	Frames + captions	Frames + captions + graph
Uses precomputed graph	×	✓
Query-specific retrieval	×	✓ (embedding)
Caption proximity sort	✓	✓
Node refinement	×	✓ (template Q&A)
Graph context injection	×	✓ (entity/action/scene)
Query deduplication	×	✓ ($K\times$ for Steps 1–3)
Embedding cache	×	✓ (BGE, once/video)
VLM generation calls	1 batched	4 batched
VLM verification calls	1 bulk	1 bulk
Per-frame visual input	Annotation clip	Annotation clip + RAG clip
Grounding signal	Captions + video	Graph context + captions + video

Table 16. VLM call counts per video. N_r: retrieved nodes per query; $|\hat{R}|$: avg. labels per prediction.

Method	Generation	Verification
Caption-Based	KM	$KM \cdot \lvert\hat{R}\rvert$
Graph RAG-Based	$M + M \cdot N_r + KM$	$KM \cdot \lvert\hat{R}\rvert$

Caption Context Integration. Both the generative and verification prompts are augmented with caption context. Captions are sorted by temporal proximity.

LLM Call Summary. In total, the RAG method uses: (1) one batched keyword extraction (unique prompts); (2) cached node retrieval (CPU-only, no VLM call); (3) one batched node checking (unique prompts); (4) one batched final answer (frame-specific clips); (5) one bulk relationship verification (optional, all frames and queries). All batched calls are chunked into sub-batches of 64.

L.5. Method Comparison

Table 15 provides a systematic comparison of the two methods along key architectural dimensions.

L.6. Efficiency and Complexity Analysis

Table 16 summarises the VLM call complexity for a video with K annotated frames and $M = |\mathcal{O}_v|$ objects.

Deduplication Savings. The overhead of Graph RAG over the caption baseline is $\mathcal{O}(M + M \cdot N_r) = \mathcal{O}(M \cdot N_r)$ for Steps 1–3. Crucially, this cost is *independent of* K due to deduplication. For typical Action Genome videos ($K \approx 30$, $M \approx 8$, $N_r = 20$), the naïve cost of performing Steps 1–3 per frame would be $K \cdot M \cdot N_r = 4{,}800$ calls; deduplication reduces this to $M + M \cdot N_r = 168$ calls—a $\sim 29\times$ reduction.

Embedding Cost. Step 2 (graph retrieval) requires **no VLM calls**—only cached matrix multiplications through the BGE embedding model. The forward pass through BGE

Table 17. Prompt usage per method. ✓ = used; × = not used.

	P0	P1	P2	P3	P4	P5
Caption-Based	×	×	×	×	✓	✓
Graph RAG-Based	✓	✓	✓	✓	✓	✓

for M keyword sets is negligible compared to VLM inference time.

Sub-Batching. All batched VLM calls are chunked into sub-batches of 64 prompts to prevent GPU OOM and vLLM scheduler thrashing. The BGE embedding model ($\sim$670 MB) runs on GPU alongside the VLM with minimal memory contention.

Prompt Usage Summary. Table 17 cross-references which prompt templates are used by each method.

Implementation Details. All VLMs are deployed via the vLLM inference engine with tensor parallelism. Key performance optimisations include: (i) 8$\times$ frame subsampling for videos with $>$120 frames, (ii) query deduplication (RAG only), processing unique prompts across steps 1–3 for an $\sim K\times$ reduction, (iii) precomputed graph embeddings shared across queries (RAG only), (iv) annotation-driven clips matching Phase 1 segmentation, and (v) chunked batching of all VLM calls into sub-batches of 64 to prevent OOM.

M. Conclusion & Future Work

Limitations. Although we see that GraphRAG with MLLMs is a promising direction, we note that with a better model, we can further improve the performance of the proposed task. Also in WSGG, we note that our performance is limited by the performance of the 3D object detection models. **Future Work.** Extending **WORLDWISE** to online, variable-length contexts for real-time world-state maintenance, enabling active sensing policies that query the representation to decide where to look next. Replacing the multi-stage geometric pipeline with end-to-end 3D-aware detectors. Extending object and predicate vocabularies via VLM language grounding. Integrating world scene graphs as the state representation for closed-loop planning agents that continuously replan based on updated world state. Evaluating world scene graphs in activity understanding, embodied navigation, and robotic manipulation.

World Scene Graphs as Structured World Representations. We note that the world scene graph is not itself a world *model* that predicts future states or simulates the consequences of actions; rather, it is a structured, interpretable, and queryable *representation* of the current and remembered world state. As such, it provides the foundational state substrate on which world models, active sensing policies, and closed-loop planners can operate—explicitly encoding what is known, what is uncertain (the unobserved set $\mathcal{U}^t$), and how entities relate, thereby grounding downstream decision-making in a persistent, 3D-anchored scene understanding.