CHRONOGRAPH: FUNCTIONAL 4D SCENE GRAPHS WITH VISION-LANGUAGE MODELS FOR INTERACTION UNDERSTANDING AND GROUNDED PLANNING

Anonymous authors
Paper under double-blind review

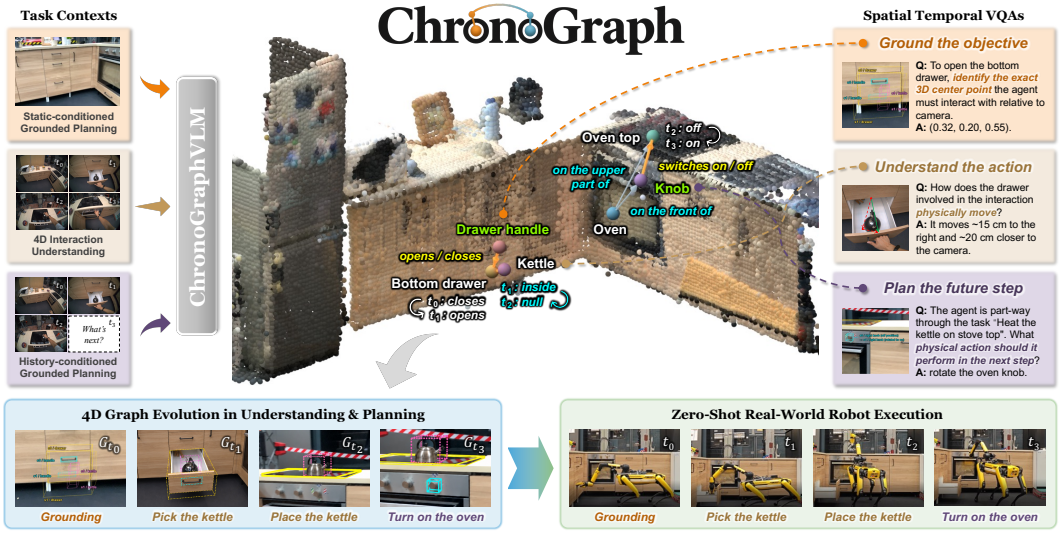

Figure 1: **ChronoGraph connects 4D interaction understanding with spatially grounded planning.** ChronoGraphVLM generates functional 4D scene graphs linking affordance-level actions to semantic and geometric state changes for interaction-grounded VQA and planning. Its predicted plans and affordance grounding guide zero-shot real-world mobile manipulation.

ABSTRACT

Embodied agents must determine where to act, anticipate the resulting scene changes, and interpret observed outcomes to guide subsequent actions. This requires connecting *4D interaction understanding*, which explains how past actions changed the scene, with *spatially grounded planning*, which determines how and where to act toward a goal and anticipates the resulting scene changes. We introduce **ChronoGraph**, a functional 4D scene graph that links actions on affordance parts to semantic and geometric state changes. By representing observed and anticipated transitions in the same form, it provides a shared basis for understanding and planning. We construct **ChronoGraphBench** through an automatic data engine that converts human-interaction videos and simulated robot trajectories into graph-annotated questions for training and evaluating Vision-Language Models (VLMs) on both tasks. Using these annotations, we train **ChronoGraphVLM** by adapting pretrained VLMs in two stages. Graph-as-Chain-of-Thought supervised fine-tuning teaches the models to reconstruct observed transitions and predict future ones as graph traces before answering. Subsequent joint 4D graph reinforcement learning directly rewards graph properties and answer correctness. Experiments across model scales show improvements over the corresponding pretrained baselines and zero-shot transfer to VLM4D. Real-world demonstrations further show that graph-based planning and affordance grounding support mobile manipulation through existing robot skills without additional fine-tuning. [1]

[1] Code and data will be released.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Embodied tasks demand more than recognizing objects or describing their arrangement. When retrieving an item from a drawer, a robot must locate the handle and anticipate that pulling it will expose the contents. After acting, it must inspect and understand whether the drawer opened sufficiently and where the item can now be grasped. Its next decision depends on interpreting the actual outcome and identifying the affordances available in the updated scene. 4D understanding and grounded planning thus address complementary sides of the same interaction: one interprets realized semantic and geometric effects, while the other predicts the next action on a grounded affordance and anticipates future changes based on the interpreted current observation.

Existing benchmarks for Vision-Language Models (VLMs) largely assess these capabilities separately. Evaluations of 3D reasoning emphasize spatial attributes and relations (Chen et al., 2024; Cheng et al., 2024; Delitzas et al., 2024), while 4D benchmarks examine how objects and scenes change over time (Zhou et al., 2026; Huang et al., 2026; Yin et al., 2026), focusing on observed object-level dynamics rather than interaction affordances. Planning benchmarks investigate action selection and task decomposition (Ju et al., 2025; Li et al., 2024; Rana et al., 2023), but rarely consider affordance-level grounding or successor-state prediction. This separation leaves the connection between understanding an interaction and planning the next one underexplored, particularly when an action changes the spatial configuration or accessibility of relevant affordances.

To make this connection explicit, we introduce **ChronoGraph**, a *functional 4D scene graph* that provides a common representation for understanding and planning. Object and affordance-part nodes encode semantic states and 3D attributes, while edges describe functional and spatial relations. Action-linked transitions specify how manipulating a particular affordance changes the scene. A graph sequence can therefore explain observed interactions or describe anticipated outcomes using the same entities, relations, and state transitions. This shared structure provides a basis for learning both how interactions unfold and how their consequences inform subsequent actions.

Using this representation, we construct **ChronoGraphBench** from egocentric human-interaction videos (Perrett et al., 2025; Engelbracht et al., 2025) and simulated robot trajectories (Nasiriany et al., 2026). Our automatic data engine recognizes interactions in the recordings, generates graph annotations, and derives interaction-grounded Visual Question Answering (VQA) pairs. The benchmark comprises two tracks: *4D interaction understanding*, which localizes observed interactions and recovers their actions, affordances, and semantic and geometric state changes, and *spatially grounded planning*, which predicts goal-directed actions and their consequences. History-conditioned planning requires interpreting preceding interactions to determine the current state and subsequent steps (Chen et al., 2023; Sermanet et al., 2024). Static-conditioned planning instead starts from a current image and additionally requires explicit grounding of execution locations.

We further turn these annotations into structured reasoning supervision for pretrained VLMs through a two-stage training pipeline, yielding **ChronoGraphVLM**. We first develop Graph-as-Chain-of-Thought supervised fine-tuning, which teaches models to reconstruct observed transitions and predict future ones as graph traces before answering. While this stage establishes the reasoning format, imitating a trace does not directly optimize its semantic and geometric correctness. We therefore follow it with joint 4D graph reinforcement learning, which provides feedback on graph semantic and geometric properties as well as answer correctness.

Experiments on ChronoGraphBench demonstrate improvements over the corresponding pretrained baselines across model scales in both interaction understanding and planning, and these gains extend to zero-shot evaluation on VLM4D (Zhou et al., 2025b). Real-world mobile-manipulation demonstrations further show that the explicitly generated functional 4D scene graph, together with the learned planning and affordance-grounding capabilities, supports sequential execution through existing robot skills without any additional fine-tuning. In summary, our contributions are:

- ChronoGraph, a functional 4D scene graph formulation that connects affordance-level interaction understanding with spatially grounded planning through observed and predicted action-conditioned state changes.
- ChronoGraphBench with a graph-based data engine, supporting training and evaluation of the interleaved understanding and planning tasks.

- ChronoGraphVLM from a training pipeline combining Graph-as-Chain-of-Thought supervised fine-tuning with joint 4D graph reinforcement learning, demonstrating improvements across model scales and benchmarks alongside transferring to robotic applications.

2 Related Work

3D and 4D Understanding with VLMs. Early benchmarks evaluate spatial understanding through visual question answering (Azuma et al., 2022; Ma et al., 2022; Chen et al., 2024; Cheng et al., 2024), while later efforts extend to multi-view and broader 3D reasoning (Yang et al., 2025; Ma et al., 2026; Yang et al., 2026; Jia et al., 2026). Image-based methods improve these capabilities through specialized training, preference optimization, and multi-image reasoning (Zhang et al., 2026a; Shen et al., 2026; Xu et al., 2026; Ouyang et al., 2025). Other approaches incorporate geometry through 3D scene inputs (Huang et al., 2024a; Deng et al., 2025), depth cues (Zhu et al., 2025; Liu et al., 2026), or reconstruction priors (Zheng et al., 2026; Wu et al., 2026; Guo et al., 2025). Recent work addresses the temporal evolution of objects and scenes (Zhou et al., 2025b; Li et al., 2025; Zhou et al., 2026; Huang et al., 2026; Yin et al., 2026), alongside spatial traces and low-level embodied grounding (Zhou et al., 2025a; Zhang et al., 2026b). These directions primarily emphasize spatial relations, object and camera motion, or observed dynamics. We instead connect observed interactions to future actions at the level of functional affordance parts. Models must localize interactions, infer their semantic and spatial effects, and use the resulting state to ground the next action and predict its consequences.

Embodied Planning with VLMs. VLM-based planning translates visual observations and language instructions into action sequences or spatial constraints (Li et al., 2024; Niu et al., 2024; Guo et al., 2024; Huang et al., 2023; 2024b; Zheng et al., 2025). Graph-based approaches represent task-relevant entities and relations for planning and verification (Rana et al., 2023; Dai et al., 2024; Ekpo et al., 2024), with MomaGraph (Ju et al., 2025) incorporating objects, interactive parts, and their spatial and functional relations in form of plain text to describe static scenes. Our formulation connects this functional structure with observed and predicted interaction dynamics in temporal domain, linking affordance-level actions to semantic and geometric changes within a shared 4D grounded graph representation.

3 ChronoGraph: Functional 4D Scene Graph for Understanding and Planning

We extend functional 3D scene graphs (Zhang et al., 2025; Hu et al., 2026; Fu et al., 2026) to model the action-triggered 4D evolution of an interactive scene. A functional 4D scene graph represents not only the objects and affordances present at a particular timestep, but also the semantic and spatial changes induced by interaction. At time step t, we define the scene graph as:

$$G_t = (\mathcal{V}_t, \mathcal{E}_t), \qquad \mathcal{V}_t = \mathcal{O}_t \cup \mathcal{P}_t, \tag{1}$$

where $\mathcal{O}_t$ denotes task-relevant object nodes and $\mathcal{P}_t$ denotes affordance-part nodes. Each node $v \in \mathcal{V}_t$ is associated with attributes of a name label, a semantic state, and 3D geometric information. Each directed edge $e = (v_1, v_2, r) \in \mathcal{E}_t$ is labeled by a relation $r \in \mathcal{R}$, which contains both functional and spatial relations (Ju et al., 2025). Thus, G_t jointly represents what entities are present, which parts can be acted upon, and how their functional and spatial states are currently related.

An interaction a_t is grounded on an affordance node p_t and induces an action-conditioned transition

$$G_t \xrightarrow{a_t} G_{t+1}, \tag{2}$$

where the relevant node states, geometric attributes, and relation edges have been updated.

For an entire task with goal z, the scene evolves through a sequence of atomic interactions:

$$\mathcal{G}_{0:T} = \left(G_0 \xrightarrow{a_0} G_1 \xrightarrow{a_1} \cdots \xrightarrow{a_{T-1}} G_T \right). \tag{3}$$

Each transition captures the local effect of one action, while the full chain represents the evolving functional and spatial state required to accomplish the goal. Each action changes the objects, accessible affordances, and spatial configuration on which subsequent actions depend. We name this explicit 4D representation **ChronoGraph**.

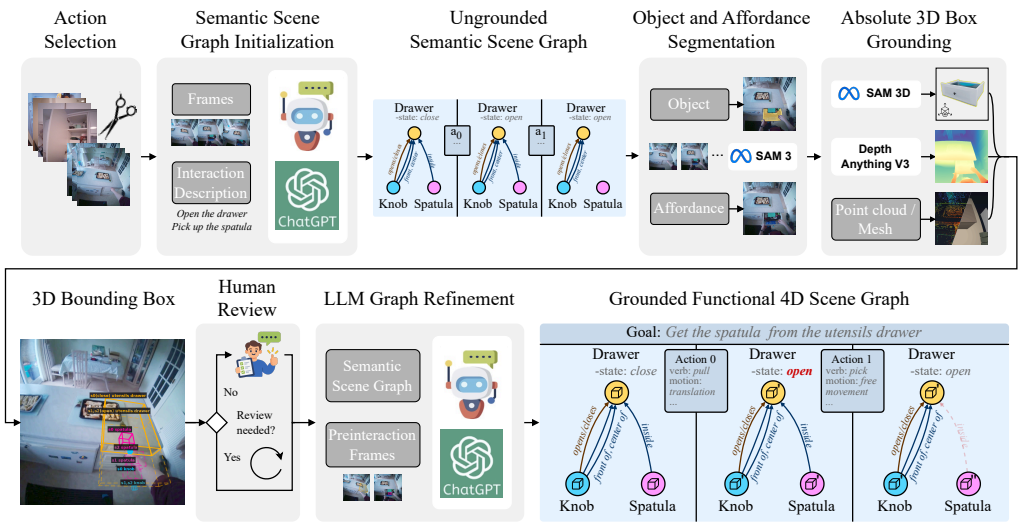

Figure 2: **ChronoGraphBench annotation pipeline.** We convert interaction recordings into functional 4D scene graphs linking affordance-level actions to semantic and geometric state changes. The pipeline combines semantic graph initialization, object and affordance grounding, human verification, and graph refinement.

4 ChronoGraphBench: Dataset and Benchmark

Connecting interaction understanding with planning requires annotations linking affordance-level actions to semantic and spatial changes. We utilize the representation of functional 4D scene graph $\mathcal{G}_{0:T}$ to construct the dataset and benchmark from egocentric human-interaction videos in HD-EPIC (Perrett et al., 2025) and HOI! (Engelbracht et al., 2025), alongside synthetic robot trajectories from RoboCasa (Nasiriany et al., 2026). These annotations support structured reasoning supervision and scalable generation of 4D-grounded VQA examples.

As a result, our dataset comprises 1,669 interaction samples, 5,143 graph states, and 15,094 QA pairs from human egocentric recordings and robot simulations. It covers 357 object categories with questions spanning all conditioning settings and reasoning categories.

4.1 ChronoGraph Annotation

All sources follow a shared annotation pipeline (Figure 2). The dataset metadata, annotations or simulator ground-truth are used to substitute corresponding pipeline components when available.

Semantic graph initialization. We first extract interaction clips with their temporal annotations, semantic labels, and descriptions. Given sampled videos and the metadata, GPT-5.5 is adopted to generate an initial semantic graph of entities, actions, semantic states, and relations. Few-shot examples specify the graph schema, establishing the semantic structure for subsequent grounding.

Object and affordance grounding. For each interaction clip, SAM3 (Carion et al., 2026) segments the manipulated object and its affordance region. We reconstruct these regions with SAM-3D (Chen et al., 2026) and estimate axis-aligned 3D bounding boxes for the corresponding graph nodes. To recover metric scale, we rescale each reconstruction using a low percentile of sparse metric point depths projected inside the mask, reducing contamination from background points. For masks without projected points, we use Depth Anything V3 (Lin et al., 2025) estimates calibrated to the metric point cloud via RANSAC. Boxes across all states share a world frame aligned with the scene's dominant horizontal direction.

Human verification and graph refinement. Automatic checks trigger human review for incomplete or invalid graph fields, ambiguous visual or spatial evidence, and low-confidence vision model outputs. We then refine the verified graphs with language descriptions. GPT-5.5 generates referring expressions that distinguish interaction-relevant objects and, conditioned on the verified graph, describes executed actions, intended goals, and scene configurations. Each completed record combines

the semantic graph, object and affordance masks, 3D annotations, and descriptions for structured supervision and question generation.

4.2 Question Construction and Evaluation Tasks

Question generation and validation. We generate graph-derived questions using 62 Python templates spanning atomic, short-horizon (2–4 sub-actions), and long-horizon (5+) interactions, given different characteristics of different time range. Before full-dataset application, templates are iteratively refined on a representative subset across multiple review rounds to address phrasing, ambiguity, answerability, and distractor quality. Questions must satisfy template-specific graph constraints. Each multiple-choice question pairs a graph-derived answer with three semantically or functionally plausible distractors that are inconsistent with the annotations. Distractor selection uses the dataset vocabulary, object statistics, and graph constraints, with GPT-5.6 Terra supplying additional candidates subject to answer-type, format, and mutual-exclusivity checks. To reduce goal-based shortcuts in atomic action questions, we compare GPT-5.6 Terra's answer correctness with and without video frames under the same textual prompt. Candidates are rejected if the frames provide no improvement across three repeated runs. We do not apply this to longer-horizon questions, which inherently require integrating temporal evidence from the video, according to our development experiments.

Evaluation settings. *4D interaction understanding* evaluates realized transitions from videos, covering spatial tracking, action recognition, and functional and semantic state changes. *Static-conditioned planning* provides a pre-interaction image and goal for predicting the next action, target affordance, and successor state, including 2D/3D grounding of the interaction location. *History-conditioned planning* provides a goal and video prefix containing the first k completed sub-actions, requiring inference of the resulting state and prediction of the next action and successor state. This history provides observations of task-relevant objects and state changes that a single current egocentric frame may omit, *e.g.*, after an agent retrieves food from a refrigerator and moves to a countertop, its current view may no longer reveal whether the refrigerator was left open.

Reasoning categories. Questions fall into five categories. *Action reasoning* covers observed or future actions, including sequences, next steps, and goal-relevant sub-actions. *Affordance reasoning* identifies interactive object parts and their functions, while *state change reasoning* concerns semantic transitions, affected entities, and transition times. *Spatial reasoning* assesses object and affordance motion, including direction, relative configuration, changes in position, and camera-relative distance. Finally, *2D/3D grounding* requires numerical predictions of target object or affordance centers in static-conditioned planning rather than multiple-choice answers.

5 ChronoGraphVLM: Graph-Guided Learning for Understanding and Planning

We train ChronoGraphVLM to generate a functional 4D scene graph trace before answering questions. Training comprises Graph-as-Chain-of-Thought supervised fine-tuning, which initializes structured graph generation and graph-conditioned answering, followed by joint 4D graph reinforcement learning, which optimizes graph consistency and answer quality.

5.1 Graph-as-Chain-of-Thought Supervised Fine-Tuning

Task-conditioned graph generation. We represent each supervised example as an input–target pair (x, y) and structure the target as an ordered *graph–evidence–answer* sequence. The graph first makes the task-relevant entities, affordances, actions, and state transitions explicit. The evidence field then selects the graph properties that support the prediction, and the final answer is generated conditioned on this structured context. This ordering encourages the model to ground its answer in an explicit scene representation.

Let q denote the VQA question, $V_{\leq t}$ the available video, I_t the current image, and z the task goal. The model inputs and corresponding graph outputs for interaction understanding, history-conditioned planning, and static-conditioned planning are

$$x_{\text{und}} = (V_{\leq t}, q) \longmapsto (\hat{G}_{0:t}, \hat{a}_{0:t-1}), x_{\text{hist}} = (V_{\leq t}, z, q) \longmapsto (\hat{G}_{0:T}, \hat{a}_{0:T-1}), x_{\text{stat}} = (I_t, z, q) \longmapsto (\hat{G}_{t:T}, \hat{a}_{t:T-1}) \tag{4}$$

where $T > t$ denotes the planning horizon. Interaction understanding reconstructs the observed graph transitions, whereas the two planning settings predict future actions and their resulting states, conditioned on the task goal. Each output trace follows the functional 4D scene graph representation $\mathcal{G}$ introduced in Section 3. All graph attributes are serialized as text, while specifically 3D geometric information is serialized through a 3D center $\mathbf{p}_{i,s} \in \mathbb{R}^3$ for each node i in state s. All centers share the coordinate system of the first input frame, or the current image for static-conditioned planning, which supports consistency of observed and predicted positions.

After generating the graph, the model outputs an evidence field $\hat{E} = [(\ell_j, c_j)]_{j=1}^{M}$, where ℓ_j identifies a graph property relevant to the question, such as a node name, semantic state or functional relation, *etc.*, and c_j is the content claimed for that property. Ground-truth evidence is generated jointly with the VQA examples described in Section 4. Placing this field between the graph trace and the answer explicitly links the final prediction to its supporting graph content.

Training objective. Writing the serialized target as the token sequence $y = (y_1, \ldots, y_{|y|})$, we minimize the autoregressive negative log-likelihood:

$$\mathcal{L}_{\mathrm{SFT}} = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{SFT}}} \left[\sum_{k=1}^{|y|} \log \pi_\theta \left(y_k \mid x, y_{<k} \right) \right], \tag{5}$$

where k indexes output tokens and π_θ denotes the VLM's conditional next-token distribution. The graph and evidence precede the answer, providing explicit context for answer generation.

5.2 Joint 4D Graph Reinforcement Learning

Token-level likelihood does not directly optimize structural consistency or task correctness. We therefore refine the model using Group-Relative Policy Optimization (Shao et al., 2024) with a composite reward that jointly evaluates graph properties and answer correctness:

$$R = \lambda_{\mathrm{fmt}} R_{\mathrm{fmt}} + \lambda_{\mathrm{graph}} R_{\mathrm{graph}} + \lambda_{\mathrm{mc}} R_{\mathrm{mc}} + \lambda_{\mathrm{gnd}} R_{\mathrm{gnd}}, \tag{6}$$

where the coefficients weight format validity, graph quality, and answer correctness.

Prediction alignment. Inserted or omitted graph states can misalign all subsequent index-wise comparisons. We therefore apply Needleman–Wunsch alignment (Needleman & Wunsch, 1970) using graph-state similarity:

$$\mathrm{sim}(n_p, n_g) = \eta\, \mathcal{M}\left(d_p^{\mathrm{state}}, d_g^{\mathrm{state}}\right) + (1-\eta)\, \mathrm{softF1}(\mathcal{N}_p, \mathcal{N}_g), \tag{7}$$

where p, g indicate the predicted and reference node, $\mathcal{M}$ measures semantic state agreement, $\mathcal{N}_p$ and $\mathcal{N}_g$ are the respective node name sets, and η balances the terms. Soft F1 is the harmonic mean of soft precision and recall, each obtained by averaging the best cross-set name agreement in its respective direction. Semantic state descriptions distinguish consecutive states that share the same entities, while the monotone alignment preserves temporal order.

Format and graph rewards. The format reward R_{fmt} checks the output wrapper, nonempty state sequence, contiguous indices, and graph state–action interleaving. The graph reward R_{graph} evaluates node names and edge types using soft F1, semantic state agreement between name-matched nodes, and graph state count. It also scores semantic and relational changes across aligned transitions. Action sequence which connects consecutive graph states is scored with equal credit to the correct affordance verb and target affordance node name. For each name-matched node, the reward on each node's 3D center $\mathbf{p}_{i,s}$ is defined as $\exp\left[-\frac{1}{2}\left(\frac{d_{uv}}{\sigma_{uv}}\right)^2 - \frac{1}{2}\left(\frac{d_z}{\sigma_z}\right)^2\right]$, where d_{uv} is the Euclidean error in normalized image coordinates, d_z is the absolute depth error, and σ_{uv} and σ_z control the respective tolerances. To capture temporal motion beyond absolute position, we additionally compute a reward of $\max\left(0, \frac{\Delta\mathbf{c}_{\mathrm{gt}}^\top \Delta\mathbf{c}_{\mathrm{pred}}}{\|\Delta\mathbf{c}_{\mathrm{gt}}\|_2 \|\Delta\mathbf{c}_{\mathrm{pred}}\|_2}\right)$, where $\Delta\mathbf{c} = \mathbf{c}_{s+1} - \mathbf{c}_s$ denotes the center motion across corresponding transitions. Transitions with negligible reference motion are excluded because their direction is unstable.

Answer rewards. Multiple-choice questions use exact-match accuracy to form R_{mc}. Grounding questions use $R_{\mathrm{gnd}} = \exp(-\alpha d)$, where d is the distance from the predicted point to the target 2D or 3D box, defined as zero inside it. This assigns full credit within the target and smoothly decreasing credit outside it.

Table 1: **Main comparison on ChronoGraphBench for 4D interaction understanding and spatially grounded planning.** Planning scores are reported as *history-conditioned/static-conditioned*, with 2D/3D grounding evaluated only in the static-conditioned setting. The overall average is the unweighted mean of per-question scores across all settings.

Model	4D Interaction Understanding					Spatially Grounded Planning						Overall Avg.
	Action Recognition	Affordance Recognition	Spatial Reasoning	State Change Recognition	Avg.	Action Prediction	Affordance Prediction	Spatial Prediction	State Change Prediction	2D/3D Grounding	Avg.	
Commercial Models												
GPT 6 Astra	93.8	72.6	75.5	88.1	85.5	85.4/85.4	81.1/81.7	78.1/69.2	85.3/96.6	—/67.9	81.3/74.2	79.4
GPT 5.6 Sol	86.8	72.6	69.6	83.5	80.1	78.0/85.4	78.4/78.5	75.0/57.8	88.2/93.1	—/61.2	78.4/68.1	73.9
GPT 5.6 Terra	84.9	74.2	64.1	77.8	76.5	75.6/82.6	67.6/83.9	65.6/54.0	61.8/93.1	—/61.5	67.3/67.4	70.9
Gemini 3.1 Pro	85.3	74.2	65.2	82.5	78.2	75.6/86.1	73.0/84.9	67.7/63.5	88.2/100.0	—/61.0	73.6/70.5	73.8
Gemini 3.8 Flash	91.9	74.2	72.8	85.1	83.4	78.0/86.8	75.7/84.9	67.7/67.3	91.2/98.3	—/66.2	75.0/73.7	77.6
Claude Opus 5	70.9	56.5	54.3	70.1	65.0	68.3/80.6	62.2/82.8	61.5/57.3	67.6/93.1	—/51.7	63.9/63.4	64.1
Kimi K3 2.8T	83.3	71.0	64.1	83.0	77.1	68.3/86.1	73.0/78.5	68.8/58.8	82.4/93.1	—/61.0	71.6/68.3	72.1
Open-Source Models												
Qwen-3.8 27B	72.1	69.4	45.7	71.1	64.6	53.7/78.5	45.9/81.7	56.2/47.9	58.8/91.4	—/45.9	54.3/58.2	60.2
Qwen-3.5 27B	72.1	72.6	54.9	72.2	67.6	58.5/77.1	54.1/84.9	63.5/45.0	70.6/93.1	—/40.0	62.0/55.1	60.7
4D / Planning Experts												
MomaGraph 7B	55.8	61.3	43.5	64.9	55.6	39.0/77.8	27.0/76.3	44.8/36.0	73.5/87.9	—/33.8	45.2/49.2	51.2
SpatialVLM 13B	43.8	40.3	32.1	34.0	37.7	41.5/56.9	32.4/72.0	46.9/28.9	29.4/84.5	—/30.6	40.4/42.1	40.2
DSR 7B	54.3	58.1	45.7	62.9	54.7	41.5/76.4	27.0/78.5	46.9/33.2	79.4/82.8	—/31.8	47.6/47.3	50.2
MLLM-4D 8B	58.1	66.1	41.8	63.9	56.2	51.2/74.3	54.1/86.0	43.8/46.9	41.2/86.2	—/33.5	46.6/51.9	53.0
Baselines v.s. Ours												
Qwen-3.5 9B	62.8	59.7	38.0	60.3	55.3	48.8/71.5	45.9/79.6	55.2/41.7	50.0/89.7	—/29.1	51.4/47.9	51.2
ChronoGraphVLM 9B	78.3	87.1	62.5	82.0	75.9	53.7/88.9	51.4/92.5	74.0/60.7	79.4/96.6	—/73.7	66.8/76.5	75.2
Qwen-3.5 4B	53.1	56.5	37.0	53.6	49.3	46.3/74.3	24.3/82.8	46.9/36.0	35.3/86.2	—/36.8	40.9/50.5	48.9
ChronoGraphVLM 4B	79.1	87.1	63.0	76.3	74.8	51.2/84.7	51.4/87.1	74.0/49.3	79.4/98.3	—/74.9	66.3/73.2	73.0

6 EXPERIMENTS

6.1 EXPERIMENTAL SETUP

Datasets and metrics. We partition the benchmark data into mutually exclusive subsets of the level of the household or simulation set-up for supervised fine-tuning, reinforcement learning and holdout evaluation, comprising 20%, 68%, and 12% of the data, respectively. This deliberately short supervised fine-tuning stage primarily teaches the model the structured graph evidence output format and provides a stable initialization for reinforcement learning. It also reduces overfitting to the supervised trajectories and better preserves the general visual and linguistic capabilities of the pretrained model (Chu et al., 2025). The evaluation set contains 202 interaction samples, including 129 human and 73 robotic interactions, from which we construct 1,811 VQAs, with careful human verification. The benchmark contains single-answer multiple-choice questions and open-numerical grounding questions. We evaluate multiple-choice questions using exact-match accuracy. For 2D and 3D grounding, we use the soft score $\exp(-\alpha d)$ (Zhang et al., 2026b), where d denotes the distance from the predicted point to the target box and is defined as zero when the prediction lies inside the box. We set $\alpha = 8$ for normalized 2D coordinates and $\alpha = 5$ for metric 3D coordinates.

Baselines. We compare our method against three groups of baselines. First, we evaluate state-of-the-art commercial models, including GPT 6 Astra, 5.6 Sol, 5.6 Terra, Gemini 3.1 Pro, 3.8 Flash, Claude Opus 5, and Kimi K3, with the medium reasoning effort configuration. Second, we evaluate open-source VLMs from the Qwen model families across multiple parameter scales. Finally, we compare with specialized methods for 4D interaction understanding and embodied planning, including MomaGraph (Ju et al., 2025), SpatialVLM (Chen et al., 2024), DSR (Zhou et al., 2026), and MLLM-4D (Yin et al., 2026). All models are evaluated using the same input protocol: for 4D interaction understanding and history-conditioned planning, videos are uniformly sampled at 4 FPS; for static-conditioned planning, each model receives the corresponding pre-interaction image.

6.2 MAIN RESULTS

Table 1 compares 4D interaction understanding and both planning settings. GPT-6 Astra leads the commercial baselines overall, while the Qwen series exhibits an overall scaling trend. However, strong semantic predictions do not imply equally accurate spatial reasoning: in static-conditioned planning, GPT-6 Astra scores 96.6 on state-change prediction but 69.2 on spatial prediction. Specialized 4D understanding or planning methods also remain limited under the unified evaluation,

Table 3: **Ablations of graph-guided reasoning and training with Qwen3.5-9B.** ChronoGraph-Bench planning scores are reported as *history-conditioned / static-conditioned*. The final column reports zero-shot accuracy on the egocentric split of VLM4D.

	4D Interaction Understanding					Spatially Grounded Planning							VLM4D
Method	Action Recognition	Affordance Recognition	Spatial Reasoning	State Change Recognition	**Avg.**	Action Prediction	Affordance Prediction	Spatial Prediction	State Change Prediction	2D/3D Grounding	**Avg.**	**Overall Avg.**	**Acc. (%)**
(a) Graph-Guided Reasoning													
Qwen3.5-9B (Direct)	62.8	59.7	38.0	60.3	55.3	48.8/71.5	45.9/79.6	55.2/41.7	50.0/89.7	—/29.1	51.4/47.9	51.2	60.5
Qwen3.5-9B + Naive Thinking	68.3	65.6	59.0	59.7	63.8	46.2/76.9	44.4/77.2	45.5/47.6	76.4/92.7	—/26.0	49.5/50.6	55.8	62.2
Qwen3.5-9B + Graph-as-CoT	67.8	67.7	50.0	70.1	63.8	51.2/80.6	45.9/77.4	58.3/49.8	50.0/87.9	—/42.4	53.3/56.7	59.0	63.9
(b) Supervised Fine-Tuning													
SFT: Answer Only	75.2	79.0	51.6	75.3	69.3	56.0/82.6	45.9/81.7	59.4/49.8	67.6/93.1	—/64.3	57.7/67.5	67.0	49.4
Ours: Graph-as-CoT SFT	76.0	82.3	55.4	71.1	69.8	51.2/86.1	48.6/89.2	64.6/55.5	79.4/91.4	—/68.9	61.5/72.0	69.9	64.2
(c) Reinforcement Learning													
RL: Answer Reward Only	78.3	85.5	62.5	78.4	74.9	56.0/88.9	43.2/89.2	70.8/54.0	76.5/94.8	—/72.2	63.9/73.8	73.0	65.4
Ours: Answer + Graph Reward	78.3	87.1	62.5	82.0	75.9	53.7/88.9	51.4/92.5	74.0/60.7	79.4/96.6	—/73.7	66.8/76.5	75.2	67.1

reaching at most 53.0 overall. ChronoGraphVLM improves understanding and both planning settings at both model scales, increasing the overall score by 24.0 at 9B and 24.1 at 4B over the corresponding baselines. The resulting scores of 75.2 and 73.0 surpass all evaluated open-source and specialized baselines. The category-level strengths align with the explicit semantic and geometric guidance from the graph. Nevertheless, a substantial gap remains in history-conditioned action and affordance prediction. These results distinguish improved interaction grounding from the still-challenging problem of selecting subsequent actions from an evolving interaction history.

6.3 Zero-Shot Cross-Benchmark Results

We evaluate zero-shot transfer on the egocentric split of VLM4D (Zhou et al., 2025b) without additional fine-tuning. As shown in Table 2, ChronoGraphVLM achieves 67.1% accuracy at 9B and 61.0% at 4B, improving over the corresponding pretrained initializations by 6.6% and 7.3%. Since VLM4D differs in both evaluation data and question construction, these gains suggest that graph-guided training develops transferable spatiotemporal reasoning rather than only improving performance on ChronoGraphBench-specific questions. As shown in Section 6.4, this transfer relies on graph supervision, since answer-only fine-tuning reduces VLM4D accuracy below the pretrained model.

Table 2: **Zero-shot transfer to VLM4D.** We report multiple-choice accuracy on the egocentric split.

Method	**Acc. (%) ↑**
Qwen3.5-9B	60.5
Qwen3.5-4B	53.7
Ours-9B	67.1
Ours-4B	61.0

6.4 Ablation Studies

Table 3 reports performance on ChronoGraphBench and zero-shot transfer to the egocentric split of VLM4D for the ablations of graph-guided reasoning and training with Qwen3.5-9B.

Graph-guided reasoning. Graph-as-CoT prompting instructs the pretrained model to identify task-relevant nodes and relations before answering, without requiring the complete functional 4D graph schema or metric 3D predictions. This training-free intervention improves the overall ChronoGraph-Bench score by 7.8 over direct answering and by 3.2 over the model's native free-form thinking mode. It also improves zero-shot VLM4D accuracy by 3.4% and 1.7% respectively. Notably, graph-guided models, including ours, do not produce longer outputs on average than models using free-form reasoning. The improvement over free-form thinking indicates that explicitly organizing entities and relations produces more intelligent behavior even without parameter updates.

Graph-as-chain-of-thought supervised fine-tuning. Relative to the baseline, answer-only supervision raises the ChronoGraphBench overall average from 51.2 to 67.0, but reduces zero-shot VLM4D accuracy from 60.5% to 49.4%. Graph-as-CoT instead improves both, reaching 69.9 and 64.2% respectively. This contrast suggests that supervising intermediate entities and state transitions improves both performance on in-domain data and cross-benchmark generalization.

Joint 4D graph reinforcement learning. Answer-only reward only increases the ChronoGraph-Bench overall score from 69.9 to 73.0 and VLM4D accuracy from 64.2% to 65.4%, while including graph rewards raises them further to 75.2 and 67.1%. These results suggest that feedback on graph properties improving understanding and planning with cross-benchmark generalization.

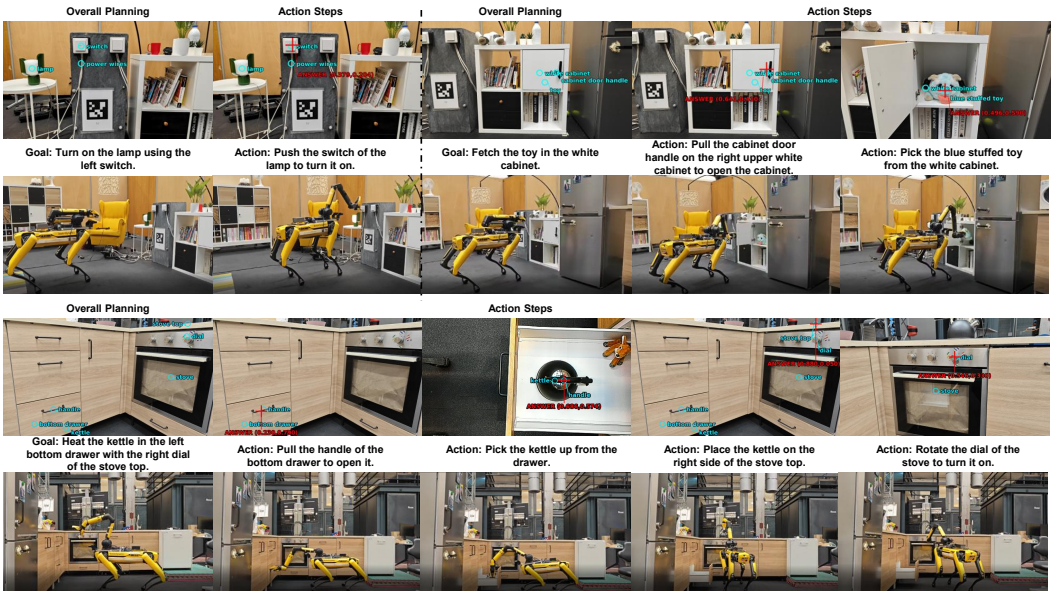

Figure 3: **Zero-shot real-world mobile manipulation with a Boston Dynamics Spot robot.** ChronoGraphVLM predicts sub-actions and updates affordance grounding before each interaction, guiding tasks of increasing horizon through existing robot skills.

6.5 Application: Mobile Manipulation

We deploy ChronoGraphVLM zero-shot on a Boston Dynamics Spot quadruped without environment- or task-specific fine-tuning. Given an initial RGB observation and a high-level goal, the model generates a subaction sequence and a task-relevant functional scene graph. Before each subaction, it uses the observation history to predict subsequent actions and grounds the target affordance in the current image. The predicted location is converted to a 3D target using measured depth and registered in the robot's local map for execution through the corresponding motion primitive. The robot then observes the updated scene and repeats this perception–grounding–action loop.

Figure 3 shows three tasks of increasing horizon: pushing a switch to activate a lamp; opening a cabinet and grasping the newly exposed toy; and opening a bottom drawer, retrieving a kettle, placing it on the appropriate stovetop region, and rotating the corresponding dial. These qualitative demonstrations illustrate how goal decomposition and observation-conditioned affordance grounding support sequential manipulation through existing robot skills in an unseen environment.

7 Conclusion and Limitations

We introduce ChronoGraph, a functional 4D scene graph that links affordance-level actions to semantic and geometric changes, connecting 4D interaction understanding with spatially grounded planning. Building on it, we construct ChronoGraphBench from human and robot interactions and train ChronoGraphVLM with Graph-as-Chain-of-Thought supervised fine-tuning and joint 4D graph reinforcement learning. The resulting models improve over pretrained baselines across scales, transfer zero-shot to VLM4D, and support real-world mobile manipulation through existing robot skills without additional fine-tuning. These results highlight the value of learning how interactions change the scene and inform subsequent actions. However, dataset scale is limited by the cost of affordance-level 3D annotation, and inference latency hinders real-time closed-loop replanning. Future work could broaden coverage across environments and robot platforms, and explore distillation and streaming inference for incremental graph and plan updates.

REFERENCES

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19107–19117. IEEE, 2022.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026.

Yi Chen, Yuying Ge, Yixiao Ge, Mingyu Ding, Bohao Li, Rui Wang, Ruifeng Xu, Ying Shan, and Xihui Liu. Egoplan-bench: Benchmarking multimodal large language models for human-level planning. *arXiv preprint arXiv:2312.06722*, 2023.

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. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Zhirui Dai, Arash Asgharivaskasi, Thai Duong, Shusen Lin, Maria-Elizabeth Tzes, George Pappas, and Nikolay Atanasov. Optimal scene graph planning with large language model guidance. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14062–14069. IEEE, 2024.

Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jiajun Deng, Tianyu He, Li Jiang, Tianyu Wang, Feras Dayoub, and Ian Reid. 3d-llava: Towards generalist 3d lmms with omni superpoint transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3772–3782. IEEE, 2025.

Daniel Ekpo, Mara Levy, Saksham Suri, Chuong Huynh, Archana Swaminathan, and Abhinav Shrivastava. Verigraph: Scene graphs for execution verifiable robot planning. *arXiv preprint arXiv:2411.10446*, 2024.

Tim Engelbracht, René Zurbrügg, Matteo Wohlrapp, Martin Büchner, Abhinav Valada, Marc Pollefeys, Hermann Blum, and Zuria Bauer. Hoi!-a multimodal dataset for force-grounded, cross-view articulated manipulation. *arXiv preprint arXiv:2512.04884*, 2025.

Zhengyu Fu, René Zurbrügg, Kaixian Qu, Marc Pollefeys, Marco Hutter, Hermann Blum, and Zuria Bauer. Funfact: building probabilistic functional 3d scene graphs via factor-graph reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23848–23858, 2026.

Yanjiang Guo, Yen-Jen Wang, Lihan Zha, and Jianyu Chen. Doremi: Grounding language model by detecting and recovering from plan-execution misalignment. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12124–12131. IEEE, 2024.

Zhongbin Guo, Jiahe Liu, Yushan Li, Wenyu Gao, Zhen Yang, Chenzhi Li, Xinyue Zhang, and Ping Jian. Beyond flatlands: Unlocking spatial intelligence by decoupling 3d reasoning from numerical regression. *arXiv preprint arXiv:2511.11239*, 2025.

Xinggang Hu, Chenyangguang Zhang, Alexandros Delitzas, Xiangkui Zhang, Marc Pollefeys, Francis Engelmann, and Xiangyang Ji. Hierarchical and holistic open-vocabulary functional 3d scene graphs for indoor spaces. *arXiv preprint arXiv:2605.15753*, 2026.

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems*, 37:113991–114017, 2024a.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024b.

Yuzhi Huang, Kairun Wen, Rongxin Gao, Dongxuan Liu, Yibin Lou, Jie Wu, Jing Xu, Jian Zhang, Zheng Yang, Yunlong Lin, et al. Thinking in dynamics: How multimodal large language models perceive, track, and reason dynamics in physical 4d world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33446–33456, 2026.

Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models. In *International Conference on Learning Representations*, volume 2026, pp. 35634–35670, 2026.

Yuanchen Ju, Yongyuan Liang, Yen-Jen Wang, Nandiraju Gireesh, Yuanliang Ju, Seungjae Lee, Qiao Gu, Elvis Hsieh, Furong Huang, and Koushil Sreenath. Momagraph: State-aware unified scene graphs with vision-language model for embodied task planning. *arXiv preprint arXiv:2512.16909*, 2025.

Kanxue Li, Baosheng Yu, Qi Zheng, Yibing Zhan, Yuhui Zhang, Tianle Zhang, Yijun Yang, Yue Chen, Lei Sun, Qiong Cao, et al. Muep: A multimodal benchmark for embodied planning with foundation models. In *IJCAI*, pp. 129–138, 2024.

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? In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5622–5632. IEEE, 2025.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Yang Liu, Ming Ma, Xiaomin Yu, Pengxiang Ding, Han Zhao, Mingyang Sun, Siteng Huang, and Donglin Wang. Ssr: Enhancing depth perception in vision-language models via rationale-guided spatial reasoning. *Advances in Neural Information Processing Systems*, 38:123926–123958, 2026.

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. *Advances in Neural Information Processing Systems*, 38:140751–140774, 2026.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. *arXiv preprint arXiv:2603.04356*, 2026.

Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

Dantong Niu, Yuvan Sharma, Giscard Biamby, Jerome Quenum, Yutong Bai, Baifeng Shi, Trevor Darrell, and Roei Herzig. Llarva: Vision-action instruction tuning enhances robot learning. *arXiv preprint arXiv:2406.11815*, 2024.

Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*, 2025.

Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Kumar Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, et al. Hd-epic: A highly-detailed egocentric video dataset. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23901–23913, 2025.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. *arXiv preprint arXiv:2307.06135*, 2023.

Pierre Sermanet, Tianli Ding, Jeffrey Zhao, Fei Xia, Debidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan, Gabriel Dulac-Arnold, Sharath Maddineni, Nikhil J Joshi, Pete Florence, Wei Han, Robert Baruch, Yao Lu, Suvir Mirchandani, Peng Xu, Pannag Sanketi, Karol Hausman, Izhak Shafran, Brian Ichter, and Yuan Cao. Robovqa: Multimodal long-horizon reasoning for robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Yifan Shen, Yuanzhe Liu, Jingyuan Zhu, Xu Cao, Xiaofeng Zhang, Yixiao He, Wenming Ye, James Rehg, and Ismini Lourentzou. Fine-grained preference optimization improves spatial reasoning in vlms. *Advances in Neural Information Processing Systems*, 38:17929–17960, 2026.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *Advances in neural information processing systems*, 38: 13569–13597, 2026.

Runsen Xu, Weiyao Wang, Hao Tang, Xingyu Chen, Xiaodong Wang, Fu-Jen Chu, Matt Feiszli, and Kevin J Liang. Multi-spatialmllm: Multi-frame spatial understanding with multi-modal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31078–31088, 2026.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643. IEEE, 2025.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, et al. Mmsi-bench: A benchmark for multi-image spatial intelligence. In *International Conference on Learning Representations*, volume 2026, pp. 157051–157088, 2026.

Xingyilang Yin, Chengzhengxu Li, Jiahao Chang, Chi-Man Pun, and Xiaodong Cun. Mllm-4d: Towards visual-based spatial-temporal intelligence. *arXiv preprint arXiv:2603.00515*, 2026.

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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19401–19413, 2025.

Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Jilin Mei, Chunhui Chen, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *Advances in Neural Information Processing Systems*, 38, 2026a.

Jiyao Zhang, Mingxu Zhang, Yitong Peng, Haoxuan Liu, Chenshuo Wang, Yuxing Long, Haoyang Huang, Dongjiang Li, Nan Duan, Hui Shen, et al. Embodied3dbench: Benchmarking low-level embodied spatial intelligence of vision language models. *arXiv preprint arXiv:2605.29074*, 2026b.

Duo Zheng, Yanyang Li, Liwei Wang, et al. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. *Advances in neural information processing systems*, 38:20560–20586, 2026.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, volume 2025, pp. 54277–54296, 2025.

Enshen Zhou, Cheng Chi, Yibo Li, Jingkun An, Jiayuan Zhang, Shanyu Rong, Yi Han, Yuheng Ji, Mengzhen Liu, Pengwei Wang, et al. Robotracer: Mastering spatial trace with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2512.13660*, 2025a.

Shengchao Zhou, Yuxin Chen, Yuying Ge, Wei Huang, Jiehong Lin, Ying Shan, and Xiaojuan Qi. Learning to reason in 4d: Dynamic spatial understanding for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9637–9646, 2026.

Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Xin Eric Wang, and Achuta Kadambi. Vlm4d: Towards spatiotemporal awareness in vision language models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8600–8612. IEEE, 2025b.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4295–4305. IEEE, 2025.

Table 4: Robot domain comparisons on RoboCasa data source.

Model	4D Interaction Understanding					Spatially Grounded Planning						Overall Avg.
	Action Recognition	Affordance Recognition	Spatial Reasoning	State Change Recognition	Avg.	Action Prediction	Affordance Prediction	Spatial Prediction	State Change Prediction	2D/3D Grounding	Avg.	
Commercial Models												
GPT 6 Astra	93.3	83.3	78.1	84.1	86.3	76.9/83.8	90.9/78.1	74.2/67.2	70.0/100.0	—/60.2	76.9/69.2	77.6
GPT 5.6 Sol	91.4	76.7	71.9	85.7	83.6	76.9/91.9	81.8/71.9	74.2/59.0	100.0/92.3	—/53.4	80.0/64.2	74.6
GPT 5.6 Terra	91.4	83.3	71.9	74.6	81.7	69.2/86.5	72.7/81.3	58.1/49.2	60.0/92.3	—/55.0	63.1/63.1	71.3
Gemini 3.1 Pro	82.9	83.3	68.8	79.4	78.6	84.6/83.8	72.7/81.3	48.4/67.2	100.0/100.0	—/48.5	67.7/64.3	71.0
Gemini 3.8 Flash	91.4	80.0	76.6	81.0	84.0	76.9/83.8	81.8/84.4	54.8/67.2	100.0/100.0	—/64.1	70.8/71.8	77.1
Claude Opus 5	68.6	66.7	51.6	69.8	64.5	69.2/78.4	63.6/78.1	64.5/60.7	80.0/92.3	—/49.9	67.7/61.9	63.7
Kimi K3 2.8T	88.6	66.7	68.8	81.0	79.4	69.2/83.8	81.8/78.1	61.3/63.9	90.0/100.0	—/50.1	70.8/63.9	71.5
Open-Source Models												
Qwen-3.8 27B	80.0	66.7	51.6	69.8	69.1	61.5/75.7	63.6/71.9	51.6/52.5	50.0/84.6	—/52.8	55.4/59.8	63.4
Qwen-3.5 27B	81.9	70.0	56.2	77.8	73.3	69.2/70.3	54.5/75.0	64.5/54.1	80.0/76.9	—/46.1	66.2/56.4	64.9
4D / Planning Experts												
MomaGraph 7B (Ju et al., 2025)	56.2	56.7	46.9	73.0	58.0	53.8/73.0	27.3/71.9	48.4/54.1	80.0/84.6	—/31.3	50.8/50.0	53.6
SpatialVLM 13B (Chen et al., 2024)	49.5	30.0	23.4	41.3	38.9	46.2/59.5	36.4/81.2	48.4/41.0	30.0/76.9	—/27.6	43.1/44.1	41.7
DSR 7B (Zhou et al., 2026)	59.0	56.7	43.8	60.3	55.3	53.8/81.1	27.3/75.0	45.2/55.7	80.0/84.6	—/25.0	49.2/48.9	51.8
MLLM-4D 8B (Yin et al., 2026)	63.8	63.3	48.4	58.7	58.8	53.8/70.3	63.6/81.2	45.2/50.8	20.0/84.6	—/31.3	46.2/49.9	53.4
Baselines v.s. Ours												
Qwen-3.5 9B	74.3	60.0	39.1	61.9	61.1	61.5/73.0	54.5/75.0	54.8/52.5	70.0/84.6	—/32.0	58.5/50.3	56.0
ChronoGraphVLM 9B	89.5	86.7	65.6	90.5	83.6	61.5/97.3	51.5/100.0	74.2/78.7	80.0/100.0	—/85.4	67.7/88.0	83.8
Qwen-3.5 4B	55.2	56.7	35.9	58.7	51.5	69.2/75.7	36.4/81.2	51.6/47.5	40.0/84.6	—/38.5	50.8/53.2	52.2
ChronoGraphVLM 4B	86.7	83.3	57.8	84.1	78.6	61.5/91.9	63.6/96.9	71.0/65.6	90.0/100.0	—/91.0	70.8/86.4	81.2

Table 5: Human domain comparisons on HD-EPIC and HOI! data source.

Model	4D Interaction Understanding					Spatially Grounded Planning						Overall Avg.
	Action Recognition	Affordance Recognition	Spatial Reasoning	State Change Recognition	Avg.	Action Prediction	Affordance Prediction	Spatial Prediction	State Change Prediction	2D/3D Grounding	Avg.	
Commercial Models												
GPT 6 Astra	94.1	62.5	74.2	90.1	85.1	89.3/86.0	76.9/83.6	80.0/70.0	91.7/95.6	—/71.3	83.2/76.3	80.3
GPT 5.6 Sol	83.7	68.8	68.3	82.4	78.0	78.6/83.2	76.9/82.0	75.4/57.3	83.3/93.3	—/64.5	77.6/69.6	73.6
GPT 5.6 Terra	80.4	65.6	60.0	79.4	73.4	78.6/81.3	65.4/85.2	69.2/56.0	62.5/93.3	—/64.3	69.2/69.2	70.7
Gemini 3.1 Pro	86.9	65.6	63.3	84.0	78.0	71.4/86.9	73.1/86.9	76.9/62.0	83.3/100.0	—/66.4	76.2/73.1	75.2
Gemini 3.8 Flash	92.2	68.8	70.8	87.0	83.0	78.6/87.9	73.1/85.2	73.8/67.3	87.5/97.8	—/67.1	76.9/74.5	77.8
Claude Opus 5	72.5	46.9	55.8	70.2	65.4	67.9/81.3	61.5/85.2	60.0/56.0	62.5/93.3	—/52.4	62.2/64.1	64.3
Kimi K3 2.8T	79.7	75.0	61.7	84.0	75.7	67.9/86.9	69.2/78.7	72.3/56.7	79.2/91.1	—/65.8	72.0/70.2	72.4
Open-Source Models												
Qwen-3.8 27B	66.7	71.9	42.5	71.8	61.9	50.0/79.4	38.5/86.9	58.5/46.0	62.5/93.3	—/42.8	53.8/57.5	58.6
Qwen-3.5 27B	65.4	75.0	54.2	69.5	64.2	53.6/79.4	53.8/90.2	63.1/41.3	66.7/97.8	—/37.3	60.1/54.6	58.7
4D / Planning Experts												
MomaGraph 7B (Ju et al., 2025)	55.6	65.6	41.7	61.1	54.1	32.1/79.4	26.9/78.7	43.1/28.7	70.8/88.9	—/34.9	42.7/48.9	50.0
SpatialVLM 13B (Chen et al., 2024)	39.9	50.0	36.7	30.5	36.9	39.3/56.1	30.8/67.2	46.2/24.0	29.2/86.7	—/31.8	39.2/41.3	39.5
DSR 7B (Zhou et al., 2026)	51.0	59.4	46.7	64.1	54.4	35.7/74.8	26.9/80.9	47.7/24.0	79.2/82.2	—/34.7	46.9/46.6	49.4
MLLM-4D 8B (Yin et al., 2026)	54.2	68.8	38.3	66.4	54.6	50.0/75.7	50.0/88.5	43.1/45.3	50.0/86.7	—/34.5	46.9/52.8	52.7
Baselines v.s. Ours												
Qwen-3.5 9B	54.9	59.4	37.5	59.5	51.8	42.9/71.0	42.3/82.0	55.4/37.3	41.7/91.1	—/27.9	48.3/46.9	48.9
ChronoGraphVLM 9B	70.6	87.5	60.8	77.9	71.3	50.0/86.0	53.8/88.5	73.8/53.3	79.2/95.6	—/68.6	66.4/71.7	71.0
Qwen-3.5 4B	51.6	56.2	37.5	51.1	47.9	35.7/73.8	19.2/83.6	44.6/31.3	33.3/86.7	—/36.1	36.4/49.4	47.3
ChronoGraphVLM 4B	73.9	90.6	65.8	72.5	72.5	46.4/82.2	46.2/82.0	75.4/42.7	75.0/97.8	—/67.8	64.3/67.8	69.1

APPENDIX

A DETAILED COMPARISON RESULTS ON DIFFERENT DATA SOURCES

To determine whether the overall conclusions are driven by either source, we further disaggregate performance by data domain. Table 4 reports results on the robotic-interaction subset from RoboCasa (Nasiriany et al., 2026), while Table 5 reports results on the human-interaction subset from HD-EPIC (Perrett et al., 2025) and HOI! (Engelbracht et al., 2025).

The source-specific results exhibit the same overall trends as the aggregate comparison. Performance improvements are therefore not attributable to a single data source or interaction domain. Strong performance on both subsets indicates that the functional 4D scene graph provides a domain-general interface between interaction understanding and spatially grounded planning, rather than specializing to the visual characteristics of either human or robotic data.

B ANNOTATION EXAMPLES AND VISUALIZATIONS

We illustrate the detailed semantic and geometric annotations of functional 4D scene graph in ChronoGraphBench. Such graphs connect high-level task structure to the specific entities, interaction regions, and spatial configurations required for grounded understanding and planning. Listing 1 presents an example in which an agent opens a refrigerator, retrieves a stick of butter, and closes the refrigerator. The task-level field `ca_action` records the complete action description and goal. The remaining annotation alternates between graph states $\{s_0, \ldots, s_3\}$ and affordance-grounded actions $\{a_0, \ldots, a_2\}$. Each graph state contains task-relevant object and affordance nodes, their semantic states, functional relations, spatial relations, and a concise state description. Each action specifies an affordance verb, the identifier of its target node, its motion type, and natural-language descriptions of the local action and subgoal. This example also demonstrates how action-induced changes are represented. From s_0 to s_1, pulling the refrigerator handle changes the refrigerator state from `close` to `open`. From s_1 to s_2, picking up the butter removes its `inside` relation with the refrigerator. Finally, pushing the handle returns the refrigerator to the closed state in s_3. Node identifiers remain consistent throughout the trajectory, allowing semantic-state and spatial-relation changes to be traced across actions.

Listing 1: Example functional 4D scene graph annotation from ChronoGraphBench. The task is decomposed into four graph states and three affordance-grounded actions.


```
{
  "ca_action": {
    "action_description": "Open the fridge, retrieve the stick of butter from inside, and
     close the fridge.",
    "goal_description": "Get the stick of butter from the fridge.",
    "no_spoiler_goal": "Retrieve the stick of butter from the refrigerator."
  },
  "s0": {
    "nodes": [
      {
        "node_id": 0,
        "cls": "fridge",
        "state": {
          "type": "OpenCloseState",
          "value": "close"
        },
        "name": "fridge"
      },
      {
        "node_id": 1,
        "cls": "handle",
        "state": null,
        "name": "fridge handle"
      },
      {
        "node_id": 2,
        "cls": "c_food",
        "state": null,
        "name": "stick of butter"
      }
    ],
    "edges": {
      "functional": {
        "source": 1,
        "relation_type": "opens/closes",
        "target": 0
      },
      "spatial": [
        {
          "source": 1,
          "relation_type": "on_the_door_of",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "left_of",
          "target": 0
        },
        {
          "source": 2,
          "relation_type": "inside",
          "target": 0
        },
        {
```

```
          "source": 1,
          "relation_type": "upper_part_of",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "front_of",
          "target": 0
        }
      ]
    },
    "state_description": "The fridge is closed, with the stick of butter inside."
  },
  "a0": {
    "affordance_action": "pull",
    "affordance_node_id": 1,
    "action_description": "Pull the handle of the under-counter fridge at the front right to
     open it.",
    "goal_description": "Open the fridge.",
    "object_position_description": "fridge",
    "motion_type": "rotation"
  },
  "s1": {
    "nodes": [
      {
        "node_id": 0,
        "cls": "fridge",
        "state": {
          "type": "OpenCloseState",
          "value": "open"
        },
        "name": "fridge"
      },
      {
        "node_id": 1,
        "cls": "handle",
        "state": null,
        "name": "fridge handle"
      },
      {
        "node_id": 2,
        "cls": "c_food",
        "state": null,
        "name": "stick of butter"
      }
    ],
    "edges": {
      "functional": {
        "source": 1,
        "relation_type": "opens/closes",
        "target": 0
      },
      "spatial": [
        {
          "source": 1,
          "relation_type": "on_the_door_of",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "left_of",
          "target": 0
        },
        {
          "source": 2,
          "relation_type": "inside",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "upper_part_of",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "front_of",
          "target": 0
        }
      ]
    },
    "state_description": "The fridge is open, and the stick of butter is still inside."
```

```
  },
  "a1": {
    "affordance_action": "pick",
    "affordance_node_id": 2,
    "action_description": "Pick up the stick of butter from the middle shelf near the front
     left inside the fridge.",
    "goal_description": "Retrieve the stick of butter.",
    "object_position_description": "stick of butter",
    "motion_type": "free_movement"
  },
  "s2": {
    "nodes": [
      {
        "node_id": 0,
        "cls": "fridge",
        "state": {
          "type": "OpenCloseState",
          "value": "open"
        },
        "name": "fridge"
      },
      {
        "node_id": 1,
        "cls": "handle",
        "state": null,
        "name": "fridge handle"
      },
      {
        "node_id": 2,
        "cls": "c_food",
        "state": null,
        "name": "stick of butter"
      }
    ],
    "edges": {
      "functional": {
        "source": 1,
        "relation_type": "opens/closes",
        "target": 0
      },
      "spatial": [
        {
          "source": 1,
          "relation_type": "on_the_door_of",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "left_of",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "upper_part_of",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "front_of",
          "target": 0
        }
      ]
    },
    "state_description": "The fridge is open, and the stick of butter has been retrieved."
  },
  "a2": {
    "affordance_action": "push",
    "affordance_node_id": 1,
    "action_description": "Push the handle of the center-right refrigerator beneath the
     countertop to close it.",
    "goal_description": "Close the fridge.",
    "object_position_description": "fridge",
    "motion_type": "rotation"
  },
  "s3": {
    "nodes": [
      {
        "node_id": 0,
        "cls": "fridge",
        "state": {
          "type": "OpenCloseState",
```



Figure 4: Target-object and affordance masks in the pre-interaction frames of three sub-actions. From left to right, the frames correspond to opening the refrigerator, retrieving the butter, and closing the refrigerator.


```
          "value": "close"
        },
        "name": "fridge"
      },
      {
        "node_id": 1,
        "cls": "handle",
        "state": null,
        "name": "fridge handle"
      },
      {
        "node_id": 2,
        "cls": "c_food",
        "state": null,
        "name": "stick of butter"
      }
    ],
    "edges": {
      "functional": {
        "source": 1,
        "relation_type": "opens/closes",
        "target": 0
      },
      "spatial": [
        {
          "source": 1,
          "relation_type": "on_the_door_of",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "left_of",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "upper_part_of",
          "target": 0
        },
        {
          "source": 1,
          "relation_type": "front_of",
          "target": 0
        }
      ]
    },
    "state_description": "The fridge is closed, and the stick of butter is present."
  }
}
```


For every action, we annotate the target object or affordance in the pre-interaction frame immediately preceding execution. These masks provide pixel-level information for identifying where the next action should be applied and are used by the static-conditioned grounding tasks. Figure 4 visualizes the masks associated with the three actions in Listing 1.

We store precise geometric annotations in a companion file linked to the functional graph through shared state and node identifiers. As shown in Listing 2, each graph state contains an oriented 3D



Figure 5: Oriented 3D bounding boxes visualized across the four graph states s_0–s_3 of the example trajectory. The boxes are registered in a shared world coordinate frame and projected into each egocentric view using the corresponding camera pose, enabling consistent spatial reasoning across viewpoint and scene-state changes.

bounding box for every geometrically annotated node. A box is parameterized by its center, extent, and rotation in a shared world coordinate frame. We additionally retain the camera pose and corresponding frame index. These poses enable the world-coordinate annotations to be transformed into the camera coordinate system used for spatial supervision and grounding. Figure 5 projects the annotated 3D boxes onto the four frames of the example trajectory. Because the boxes are registered in a common coordinate frame, the annotation preserves object identity and geometry despite egocentric camera motion. It therefore supports reasoning about object location, scale, displacement, and action-induced changes across the complete interaction.

Listing 2: Companion geometry annotation for the same episode. Oriented 3D bounding boxes are stored in a shared world SLAM coordinate frame, together with the frame indices and camera poses of frames.


```
{
    "video_id": "P02-20240211-120927",
    "coord_frame": "world_slam",
    "bboxes": {
        "s0": {
            "0": {
                "center": [
                    -0.536388,
                    -1.3989725,
                    -1.1495945
                ],
                "extent": [
                    0.3000323515735966,
                    0.3162529440213807,
                    0.4070465
                ],
                "rotation": [
                    0.9907208613059781,
                    0.1359123797640993,
                    0.0
                ]
            },
            "1": {
                "center": [...],
                "extent": [...],
                "rotation": [...],
            },
            "2": {
                "center": [...],
                "extent": [...],
                "rotation": [...],
            }
        },
        "s1": {
            "0": {...},
            "1": {...},
            "2": {...},
        },
        "s2": {
            "0": {...},
            "1": {...},
            "2": {...},
        },
        "s3": {
            "0": {...},
            "1": {...},
```

```
            "2": {...},
        }
    },
    "camera_poses": {
        "states": {
            "s0": {
                "frame_index": 17854,
                "T_world_camera": [
                    [
                        0.9845765077205448,
                        0.17057703973833804,
                        0.03889182380554134,
                        -0.8887244076646096
                    ],
                    [
                        0.12134102841770068,
                        -0.8259173433950197,
                        0.5505786925606988,
                        -1.8895987623183907
                    ],
                    [
                        0.12603751531726884,
                        -0.5373676724491517,
                        -0.8338768070519956,
                        -0.14215577443889113
                    ],
                    [
                        0.0,
                        0.0,
                        0.0,
                        1.0
                    ]
                ]
            },
            "s1": {
                "frame_index": 17888,
                "T_world_camera": [...]
            },
            "s2": {
                "frame_index": 17943,
                "T_world_camera": [...]
            },
            "s3": {
                "frame_index": 17969,
                "T_world_camera": [...]
            }
        },
        "preinteraction": {
            "a0": {
                "frame_index": 17845,
                "T_world_camera": [...]
            },
            "a1": {
                "frame_index": 17879,
                "T_world_camera": [...]
            },
            "a2": {
                "frame_index": 17944,
                "T_world_camera": [...]
            }
        }
    }
}
```


C Detailed Dataset Statistics

Our dataset contains 1,669 interaction episodes, represented by 5,143 functional 4D graph states and 15,094 VQA pairs. As shown in Figure 6(a), the dataset combines 1,008 real-world human-interaction recordings from 33 households with 661 simulated robot episodes from RoboCasa. The human-domain data comprise 687 HD-EPIC episodes and 321 HOI! episodes, accounting for 60.4% of the dataset, while the RoboCasa portion covers 61 manipulation tasks. The robot episodes span 425 distinct combinations of environment layout and visual style, drawn from 60 layouts and 60 styles.

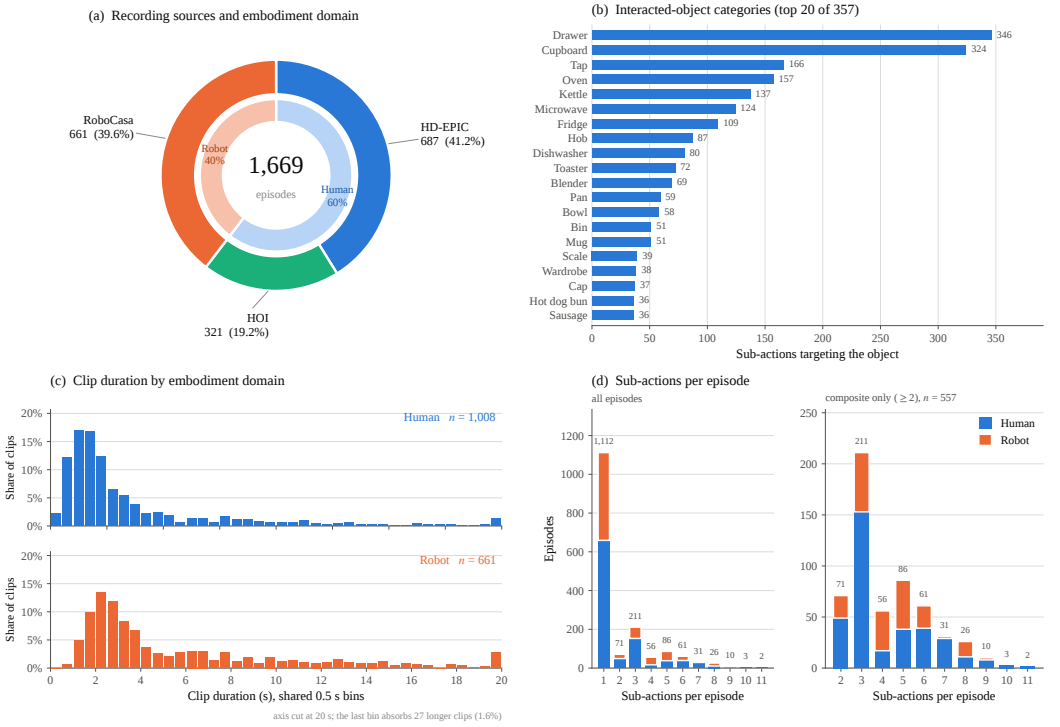

Figure 6: Dataset statistics. (a) Distribution across recording sources and embodiment domains. (b) The 20 most frequently interacted object categories among 357 categories. (c) Clip-duration distributions for the human and robot domains. (d) Number of sub-actions per episode, shown for all episodes and for the composite subset containing at least two sub-actions.

Temporal composition. Figure 6(c) shows that human recordings are concentrated among short interactions of approximately one to three seconds, but also exhibit a long tail of more extended activities. RoboCasa episodes are generally longer and have a broader duration distribution, reflecting the execution time required by simulated manipulation policies. To characterize compositional complexity, we additionally report the number of sub-actions per episode in Figure 6(d). The dataset contains 1,112 atomic episodes with a single sub-action and 557 composite episodes with at least two sub-actions. Although short sequences are most common, the composite subset extends to eleven sub-actions and therefore supports the evaluation of temporally extended understanding and planning.

Interacted-object coverage. The dataset covers 357 interacted-object categories. As illustrated in Figure 6(b), frequently manipulated objects include drawers, cupboards, taps, ovens, kettles, microwaves, and refrigerators. This distribution reflects the benchmark's emphasis on household activities involving articulated objects and functional parts. At the same time, the long-tailed object vocabulary includes appliances, containers, utensils, furniture, and task-specific tools, providing diverse object–affordance associations and interaction dynamics.

VQA distribution. Figure 7 summarizes the distribution of QA pairs across evaluation tracks and reasoning categories for both the full benchmark and the held-out test split. The full benchmark contains 6,259 dynamic-video-understanding questions, 7,018 static-conditioned-planning questions, and 1,817 history-conditioned-planning questions. The smaller history-conditioned subset is intentional: this setting requires a temporally stable sequence containing at least two completed sub-actions, for which both the interaction history and the subsequent action can be reliably annotated. Such long, coherent interaction sequences occur less frequently than atomic interactions or static planning instances. Because the resulting track sizes are imbalanced, our benchmark tables report track-specific results in addition to aggregate performance, preventing the larger tracks from obscuring performance on history-conditioned planning.

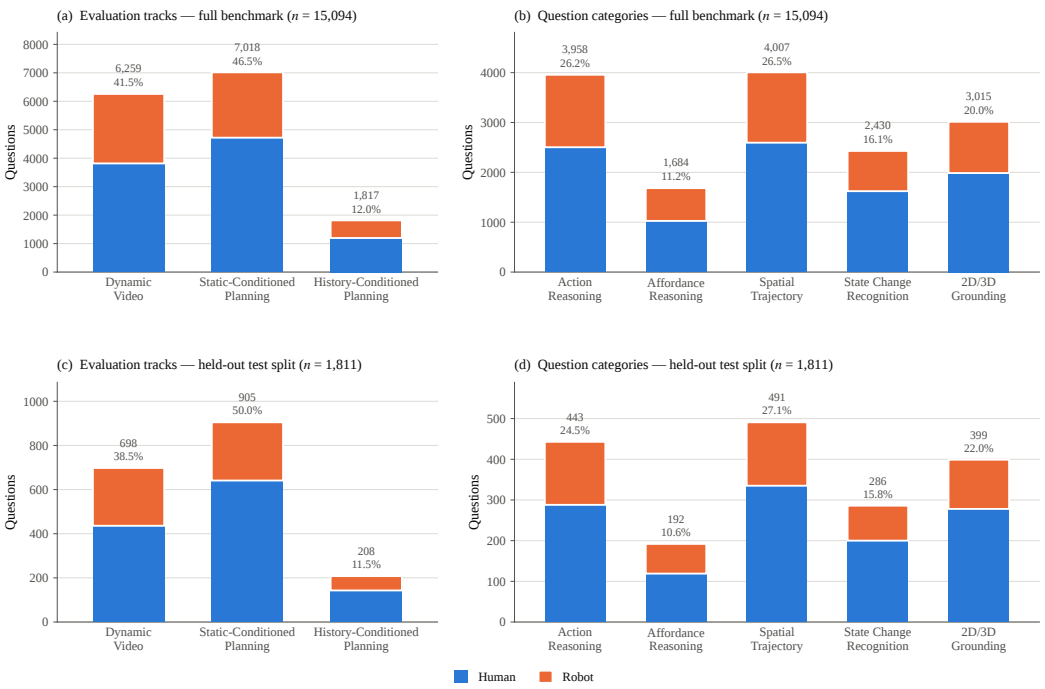

Figure 7: Distribution of VQA pairs by evaluation track and reasoning category. The top row reports statistics for the complete benchmark, and the bottom row reports the held-out test split. Colors distinguish questions derived from human-interaction recordings and robot-simulation episodes.

Across reasoning categories, action reasoning and spatial-trajectory reasoning constitute the largest subsets, followed by 2D/3D grounding, state-change recognition, and affordance reasoning. We deliberately filter near-duplicate affordance and state-change questions, which would otherwise overrepresent recurring concepts such as handles, knobs, and binary open/closed or on/off transitions. In contrast, we retain a broader range of action and spatial questions because they exhibit greater diversity in objects, motion patterns, spatial configurations, and linguistic vocabulary. These categories also most directly assess the capabilities required for downstream robot execution: selecting an action, identifying where it should be applied, and predicting its spatial consequences. The held-out split preserves a similar distribution across domains, tracks, and question categories.

D Detailed VQA Samples

We present example VQA questions in ChronoGraphBench respectively on 3 tracks: 4D interaction understanding (Figure 8), history-conditioned planning (Figure 9), and static-conditioned planning (Figure 10).

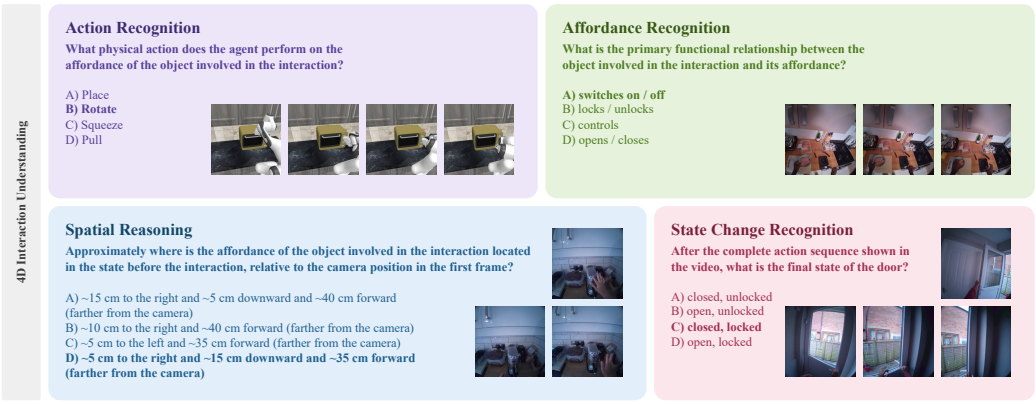

Figure 8: Example benchmark questions from 4D interaction understanding track.

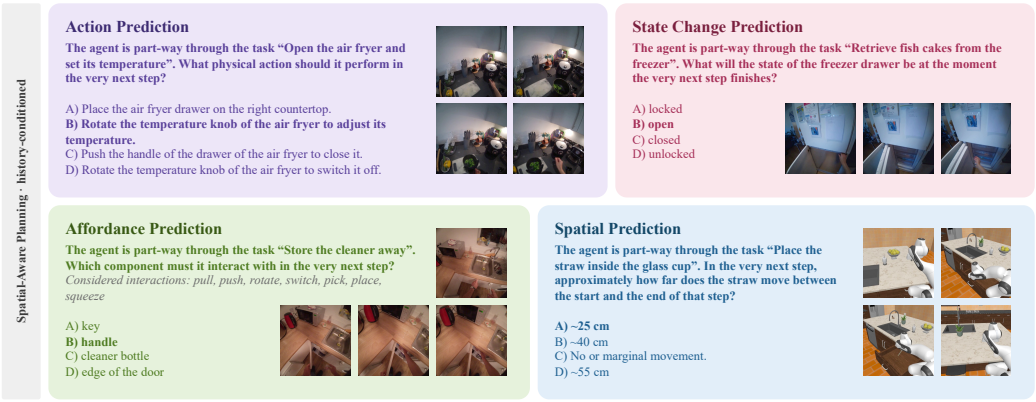

Figure 9: Example benchmark questions from history-conditioned planning track.

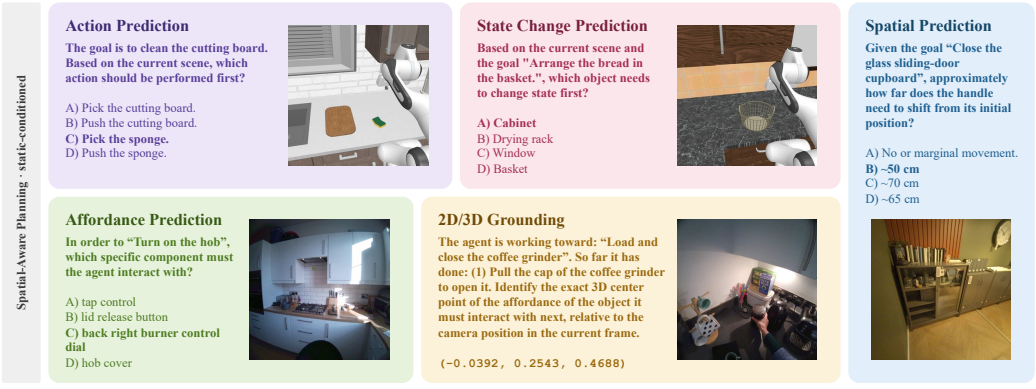

Figure 10: Example benchmark questions from static-conditioned planning track.

E Reward Details of the Reinforcement Learning Stage

Format reward. The policy must emit its scene graph as a JSON object inside an `<answer>...</answer>` block, and the format reward scores how far a completion satisfies that contract. Let $\mathbf{1}\{\text{open}\}$ indicate that the opening `<answer>` tag is present and $\mathbf{1}\{\text{close}\}$ that the block is also properly closed. $|S|$ and $|A|$ denote the numbers of graph state and action keys recovered from the results. The reward is

$$R_{\text{fmt}} = \min\Big\{1,\, 0.25\,\mathbf{1}\{\text{open}\} + 0.15\,\mathbf{1}\{\text{close}\} + r_{\text{parse}} + 0.15\,\mathbf{1}\{|S| \geq 1\} + 0.20\,\mathbf{1}\{\text{chain}\}\Big\}, \tag{8}$$

with $r_{\text{parse}} = 0.25$ for directly parseable JSON, 0.10 for JSON recovered by the repair pass, and 0 otherwise. The chain indicator requires state indices contiguous from zero and interleaved with actions, $|A| = \max(|S| - 1, 0)$.

Graph reward. After aligning the predicted and ground-truth state sequences, we evaluate both semantic-structural consistency and metric spatial accuracy within a single graph reward:

$$R_{\text{graph}} = 0.5\, r_{\text{sem}} + 0.5\, r_{\text{geo}}. \tag{9}$$

The semantic-structural component is

$$r_{\text{sem}} = \min\Big\{1,\, 0.10\, r_{\text{name}} + 0.05\, r_{\text{edge}}^{\text{func}} + 0.05\, r_{\text{edge}}^{\text{spat}} + 0.20\, r_{\text{state}} + 0.05\, r_{\text{count}} + 0.20\, r_{\text{change}} + 0.15\, r_{\text{shift}} + 0.20\, r_{\text{act}}\Big\}. \tag{10}$$

Here, r_{name}, $r_{\text{edge}}^{\text{func}}$, and $r_{\text{edge}}^{\text{spat}}$ are soft F1 scores over node names, functional edges, and symbolic spatial edges, respectively. The semantic state term r_{state} measures discrete state value agreement between name-matched predicted and ground-truth nodes.

The count term penalizes an incorrect number of graph states:

$$r_{\text{count}} = \max\left(0,\, 1 - \frac{\big||S_{\text{pred}}| - |S_{\text{gt}}|\big|}{\max(|S_{\text{gt}}|, 1)}\right). \tag{11}$$

It therefore penalizes both prematurely terminated trajectories and trajectories padded with redundant states.

The transition term r_{change} considers only entities whose semantic states differ between consecutive ground-truth states. A prediction receives credit only if it produces the correct new value and its predicted state also changes. Similarly, r_{shift} compares the predicted change between consecutive edge sets with the corresponding ground-truth change. These transition-sensitive terms discourage the degenerate strategy of copying the initial graph across the entire trajectory, which would otherwise receive substantial credit because most nodes and relations legitimately persist over time.

We score the affordance-grounded action sequence encoded in the graph:

$$r_{\text{act}} = \frac{1}{|A_{\text{gt}}|} \sum_{t=0}^{|A_{\text{gt}}|-1} \left[\frac{1}{2}\mathbf{1}[\hat{\nu}_t = \nu_t \wedge \nu_t \in \mathcal{V}] + \frac{1}{2}\mathbf{1}[\text{norm}(\hat{n}_t) = \text{norm}(n_t)]\right], \tag{12}$$

where $\mathcal{V}$ is the affordance-verb vocabulary and n_t is the name of the target node. Because action t connects graph states t and $t+1$, it is evaluated only when the state alignment maps the corresponding ground-truth states to adjacent predicted states. Scoring actions explicitly prevents an incorrect action from being diluted by the many correctly predicted nodes and relations in a long trajectory.

The geometric component R_{geo} measures both absolute node localization and action-induced motion. For each ground-truth node with 3D center $\mathbf{p}_{i,s}$, we locate the name-matched predicted node and compute

$$r_{\text{center}} = \exp\left[-\frac{1}{2}\left(\frac{d_{uv}}{\sigma_{uv}}\right)^2 - \frac{1}{2}\left(\frac{d_z}{\sigma_z}\right)^2\right], \tag{13}$$

where d_{uv} is the Euclidean error in normalized image coordinates, d_z is the absolute depth error, $\sigma_{uv} = 0.08$, and $\sigma_z = 0.20\,\mathrm{m}$. Separate scales are necessary because the image coordinates are normalized and can be localized relatively precisely, whereas monocular depth is measured in metric units and is inherently more ambiguous.

To evaluate motion, we track each matched node across the aligned state pair associated with an action. Let $\Delta\mathbf{c} = \mathbf{c}_{s+1} - \mathbf{c}_s$ denote its center displacement. The direction score is

$$r_{\mathrm{dir}} = \mathrm{clip}\left(\frac{\Delta\mathbf{c}_{\mathrm{gt}}^{\top}\Delta\mathbf{c}_{\mathrm{pred}}}{\|\Delta\mathbf{c}_{\mathrm{gt}}\|_2 \, \|\Delta\mathbf{c}_{\mathrm{pred}}\|_2}, \, 0, \, 1 \right). \tag{14}$$

We exclude pairs satisfying $\|\Delta\mathbf{c}_{\mathrm{gt}}\|_2 < 0.02$, because a nearly static node does not provide a meaningful motion direction. Predictions with zero displacement receive zero directional credit.

Finally, we aggregate the available geometric terms as

$$r_{\mathrm{geo}} = \frac{0.75\, I_{\mathrm{center}}\, \bar{r}_{\mathrm{center}} + 0.25\, I_{\mathrm{dir}}\, \bar{r}_{\mathrm{dir}}}{0.75\, I_{\mathrm{center}} + 0.25\, I_{\mathrm{dir}}}, \tag{15}$$

where $\bar{r}_{\mathrm{center}}$ and $\bar{r}_{\mathrm{dir}}$ denote the mean scores over valid node matches, and each indicator I specifies whether the corresponding annotation is available. Thus, unavailable geometric components abstain rather than being treated as incorrect.

Answer rewards. Multiple-choice answers are scored by exact option match, comparing the leading character after normalisation,

$$R_{\mathrm{mc}} = \mathbf{1}[\hat{a}_{\mathrm{mc}} = a_{\mathrm{mc}}]\,. \tag{16}$$

Grounding uses

$$R_{\mathrm{gnd}} = \exp(-\alpha d), \tag{17}$$

where d is the distance from the predicted point to the target box surface and is zero for any point inside. In 2D the target is the axis-aligned bounding box of the target mask and $\alpha = 8$ in normalised coordinates. In 3D, given the predicted camera-frame point $\mathbf{x}$,

$$d = \left\| \max\!\left(\mathbf{0},\ \left| R_{\mathrm{box}}^{\top}(\mathbf{x} - \mathbf{c}_{\mathrm{box}}) \right| - \tfrac{\mathbf{e}}{2} \right) \right\|_2, \tag{18}$$

with R_{box}, $\mathbf{c}_{\mathrm{box}}$ and $\mathbf{e}$ the box orientation, centre and side lengths, and $\alpha = 5$ for distances in metres.

Composite objective.

$$R = \lambda_{\mathrm{fmt}} R_{\mathrm{fmt}} + \lambda_{\mathrm{graph}} R_{\mathrm{graph}} + \lambda_{\mathrm{mc}} R_{\mathrm{mc}} + \lambda_{\mathrm{gnd}} R_{\mathrm{gnd}}, \tag{19}$$

with $\lambda = (0.05,\ 0.30,\ 0.42,\ 0.23)$

Rewards as performance metrics for the graph generation quality. Since our reinforcement learning rewards directly evaluate the quality of the generated graph versus the ground truth one, we report some statistics here to demonstrate the graph quality inferred by ChronoGraphVLM: during the whole reinforcement learning stage, R_{graph} is moved from 0.557 to 0.620, where specifically r_{sem} is from 0.713 to 0.772 while r_{geo} is from 0.402 to 0.462. These show the overall generation accuracy and moving trend in the reinforcement learning stage.

F Qualitative Analysis of Generated Graphs

Figure 11 qualitatively compares the graph trajectories generated by our model with the ground-truth annotations across all three evaluation settings. For visualization, we project each node into the corresponding ground-truth state frame using its annotated or predicted 3D center and the associated camera pose. Each state is displayed as a pair: ground-truth nodes are shown in green on the left, while predicted nodes are shown in blue on the right.

In the interaction-understanding example, the generated graph recovers the task-relevant object and affordance nodes from the observed transition and places them consistently across the pre- and post-interaction states. In the static-conditioned examples, the model must go beyond recognition: from a

single initial observation and task goal, it preserves entity identity while predicting the future states produced by successive actions. The history-conditioned example further shows that the model can incorporate an observed interaction prefix and continue the graph trajectory into unobserved future states. Overall, the predicted nodes closely follow the ground-truth annotations despite changes in viewpoint, articulation, and object position, with only minor localization deviations, thanks to the consistent 3D geometric information. More importantly, the graph traces remain structurally consistent across understanding and planning: observed states are reconstructed using the same representation with which future states are predicted. These examples support the central role of the functional 4D scene graph as a unified interface for jointly learning interaction understanding and spatially grounded planning.

G Implementation Details

For supervised fine-tuning, we initialize from a pretrained VLM, *e.g.,* Qwen-3.5 4B and Qwen-3.5 9B, and fine-tune all language model parameters while keeping the vision encoder frozen. We train for one epoch using a learning rate of 1×10^{-5}. This deliberately short supervised fine-tuning stage primarily teaches the model the structured graph evidence output format and provides a stable initialization for reinforcement learning. It also reduces overfitting to the supervised trajectories and better preserves the general visual and linguistic capabilities of the pretrained model (Chu et al., 2025). The reinforcement learning stage is initialized from the fine-tuned checkpoint. We sample 8 generations per prompt and use 32 prompts, corresponding to 256 completions, per optimization step. We set the learning rate to 1×10^{-6} and the KL coefficient to $\beta = 0.04$.

H Discussion of the Training Allocation

Our default training schedule deliberately uses a short supervised fine-tuning stage followed by a longer reinforcement learning stage. This choice reflects the different roles of the two stages: SFT teaches the model the output structure and establishes the initial graph-conditioned reasoning behavior, whereas RL optimizes semantic, geometric, and answer-level consistency without requiring the model to imitate a fixed target sequence. This design is also consistent with prior evidence that SFT tends to fit patterns observed in the training data, whereas RL can provide stronger out-of-distribution generalization (Chu et al., 2025).

During development, we increased the SFT allocation to 50% of the training data (44% of all data), resulting in approximately 420 optimization steps. The held-out validation loss was 0.1345, 0.1326, 0.1423, and 0.1501 at steps 100, 200, 300, and 420, respectively. The loss reached its minimum near step 200 and increased thereafter, indicating that extending SFT beyond this point leads to overfitting. We therefore allocate approximately 20% of all data (about 23% of the training data) to SFT, corresponding to roughly 190 optimization steps near this minimum, and assign the remaining training data and computation to RL.

Table 6 compares results under our default allocation and a balanced 50%/50% SFT–RL allocation. The balanced allocation generally improves ChronoGraphBench performance, but at the cost of zero-shot transfer, as VLM4D accuracy decreases after both stages. After SFT, it raises the overall average by 4.6 but lowers VLM4D accuracy by 1.5%. For the complete model, it improves the overall average by only 1.4 while reducing VLM4D accuracy by 3.2%, and it slightly lowers static-conditioned planning (75.6 vs. 76.5). The subsequent RL stage also contributes less under this allocation, improving the overall average by 2.1 and VLM4D accuracy by 1.2%, compared with 5.3 and 2.9% under the default allocation. We also qualitatively observe less reliable behavior under this allocation in the real-world robot experiments. Together with the validation loss trend, these results suggest that prolonged SFT increasingly specializes the model to the annotation and question distributions of ChronoGraphBench and leaves less room for RL to improve generalization. We therefore adopt the short-SFT, long-RL schedule reported in the main paper, accepting a small reduction in in-domain accuracy in exchange for stronger cross-benchmark and embodied generalization.

Table 6: Comparison under the default and balanced SFT–RL data allocations. For each trainable configuration, the *Default* and *50/50* rows use the same architecture, supervision, and held-out evaluation set, differing only in the allocation of training data. Planning averages are reported as *history-conditioned / static-conditioned*. Training-free prompting baselines are shared across the two settings.

Method	Allocation	Understanding Avg.	Planning Avg.	Overall Avg.	VLM4D Acc. (%)
Graph-Guided Reasoning					
Qwen-3.5 9B	Shared	55.3	51.4/47.9	51.2	60.5
Qwen-3.5 9B Naive Think	Shared	63.8	49.5/50.6	55.8	62.2
Qwen-3.5 9B Graph-as-CoT	Shared	63.8	53.3/56.7	59.0	63.9
Supervised Fine-Tuning					
Ours: SFT	Default	69.8	61.5/72.0	69.9	64.2
	50/50	78.5	64.9/73.6	74.5	62.7
Reinforcement Learning					
Ours	Default	75.9	66.8/76.5	75.2	67.1
	50/50	80.5	67.3/75.6	76.6	63.9

I AI DISCLOSURE

In this work, we used generative AI tools (LLMs) to aid and polish the writing to improve readability and clarity. Meanwhile, it is used in our automatic data engine, with human verification on the generated content quality. We have not used generative AI tools for developing theoretical models, formulating or proving mathematical claims, proposing hypotheses, designing experiments, or interpreting results. All AI-assisted text was thoroughly reviewed, verified, and edited by the authors to ensure accuracy and fidelity to the original research. The authors take full responsibility for the final content and claims presented in this paper.

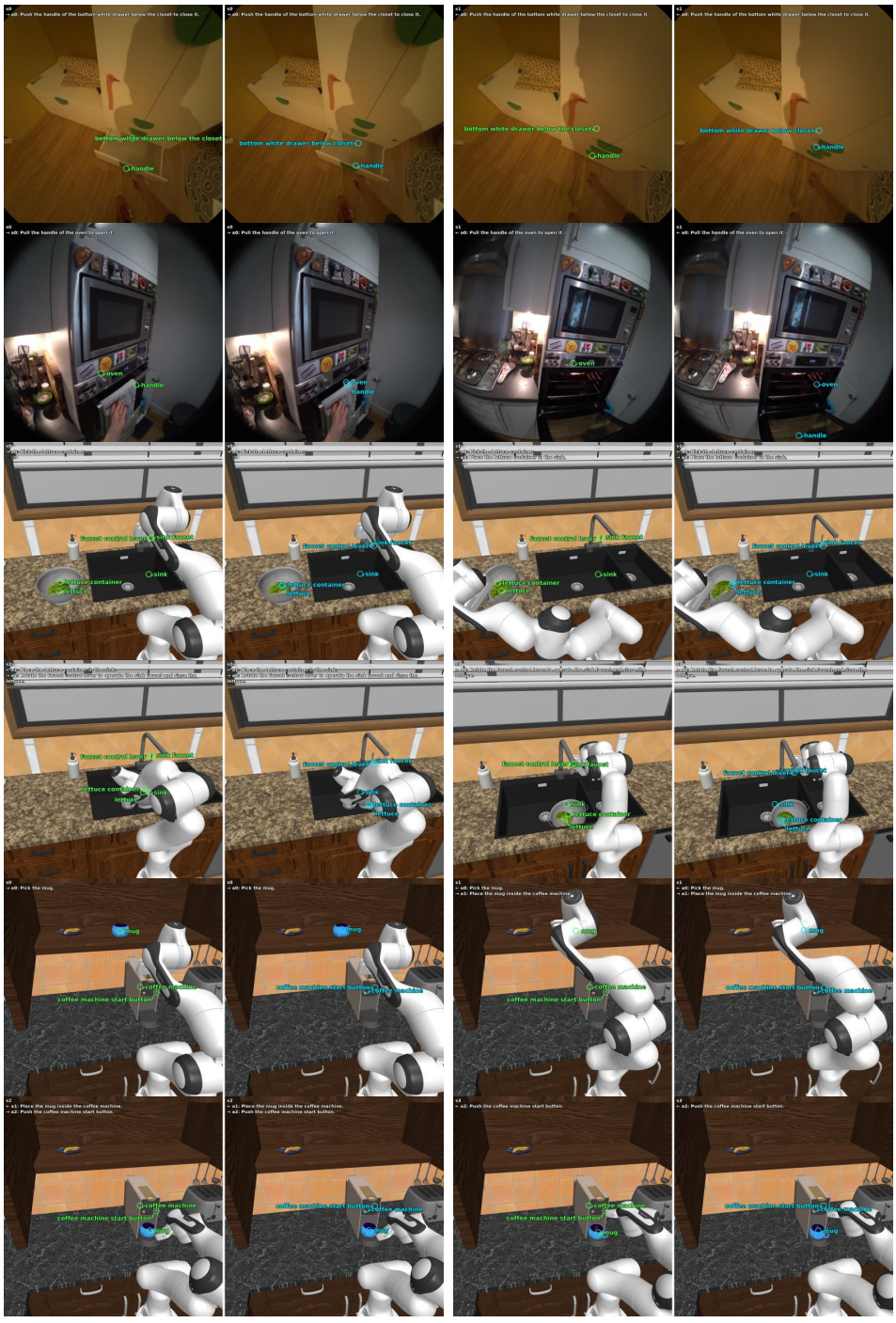

Figure 11: **Qualitative comparison of generated and ground-truth functional 4D scene graphs.** Each graph state is visualized by projecting its nodes into the corresponding ground-truth frame using the associated camera pose. Green nodes in the left image of each pair denote ground-truth annotations, and blue nodes in the right image denote model predictions. Scene 1 shows 4D interaction understanding, for which states s_0 and s_1 are observed and reconstructed. Scenes 2 and 3 show static-conditioned planning: only the initial frame is provided, and the subsequent graph states are inferred. Scene 4 shows history-conditioned planning, in which s_0 and s_1 are observed while s_2 and s_3 are predicted. Across the three settings, the same graph representation is used to reconstruct observed interactions and forecast future action-conditioned scene states.