Learning to Evolve Scenes: Reasoning about Human Activities with Scene Graphs

Francesca Pistilli
Politecnico di Torino
Torino, Italy
francesca.pistilli@polito.it

Simone Alberto Peirone
Politecnico di Torino
Torino, Italy
simone.peirone@polito.it

Giuseppe Averta
Politecnico di Torino
Torino, Italy
giuseppe.averta@polito.it

Abstract

Understanding human behavior while interacting with the surrounding world is crucial for many applications of embodied AI. First-person videos are particularly informative for this problem, as they well capture how activities reshape the scene over time. However, existing approaches often rely on implicit visual or language-aligned representations, disregarding structured reasoning over the scene dynamic. We argue that explicit, compositional and editable representations of human-environment interactions can play a crucial role for rich grounded activity understanding. To this end, we introduce SG-Ego, a large scale annotation set extending Ego4D with spatio-temporal scene graphs, where relations triplets are consolidated over time into explicit time-evolving descriptions of the scene state. To reason over this representation, we propose GLEN, a graph-based model that operates over scene graph sequences to both align them with textual actions and model their temporal evolution. In addition, we formulate the activity-conditioned graph-edit forecasting (A-GEF) problem, a novel task that casts scene dynamics as a sequence of structured transformations conditioned on ongoing actions, enabling explicit reasoning about how scenes change over time. We validate our approach across multiple downstream tasks, spanning retrieval benchmarks as EgoMCQ and EgoCVR, as well as long-horizon reasoning benchmarks as EXPLORE-Bench and the newly introduced A-GEF. GLEN achieves strong results compared to raw video baselines and it excels in reasoning settings, typically addressed only with MLLMs, while enabling controllable and structured predictions of scene dynamics driven by human activities. We believe our results establish spatio-temporal scene graphs, together with models that reason over them, as strong compositional and interpretable representations for video understanding and potentially beyond. Project page with code at https://francescapistilli.github.io/GLEN.

1 Introduction

Despite rapid progress in video understanding, modeling how scenes evolve under human actions remains a fundamental challenge. Reasoning over these transformations is crucial for embodied AI systems that have to plan, forecast and act in shared dynamic environments [1, 2].

First-person videos [3] are particularly suited to support this kind of reasoning, thanks to their interaction-centered focus on how actions reshape the scene status over time. Despite significant

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

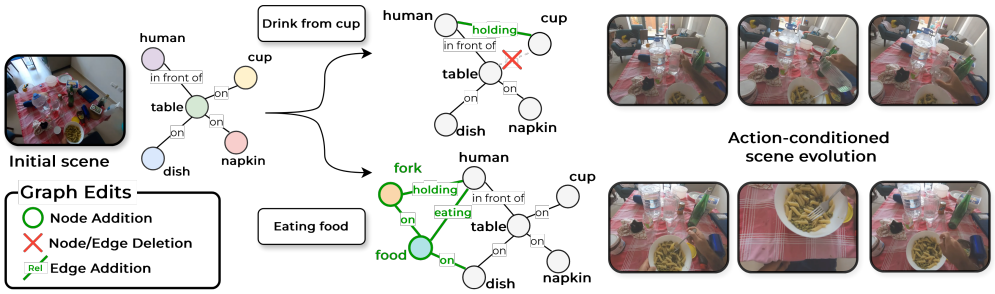


Figure 1: **Reasoning over evolving scenes.** Different activities induce structured changes in a scene over time. We formulate this process as *activity-conditioned graph edit forecasting* (A-GEF), where the temporal evolution of a scene is modeled as graph edits conditioned on an activity description.

interest of the community on this topic [4–7], current approaches typically handle egocentric video understanding similarly to other computer vision tasks, relying on large-scale trainings to learn compact latent representations from raw pixels or dense visual features. While effective in many practical cases, such approaches entangle objects, interactions, and their temporal dynamics into a single embedding space, making it difficult to reason about how and why a scene changes over time. Approaches that include textual narrations like [5, 6] move from purely visual to visual and semantic features, but still rely on implicit representations. The core limitation of these works is not merely a lack of semantics, but rather the absence of an explicit, editable and compositional representation of the human-scene interaction. Without such structure, it remains challenging to attribute changes in the scene to specific actions or to reason about how the scene evolves under different activities.

We argue that this can be addressed by structured representations in the form of sequences of scene graphs, where objects and the human are represented as nodes and their functional or spatial relations as edges. This recasts video understanding tasks from a passive feature extraction to an explicit activity-conditioned reasoning over scene structure, with temporal dynamics expressed as structured transformations of scene state rather than implicit shifts in latent embeddings. In other words, sequences of scene graphs provide an interpretable, compositional and editable representation that naturally supports reasoning about *what* changed, *why*, and *how the scene evolves over time*.
While scene graphs have been studied in other contexts like generation from image or video [8–10] or future scene anticipation [11–13], prior works remain fundamentally passive and limited to existing benchmarks, disregarding the activity that generates the scene change. In this work we go beyond existing paradigms and introduce *activity-conditioned graph edit forecasting*, a novel approach for activity understanding that maps the dynamic evolution of a scene conditioned by human activities as edits on a scene graph, see Fig. 1. First, we introduce SG-Ego, a large-scale annotation set for Ego4D of spatio-temporal scene graphs, where relational triplets are automatically annotated and then consolidated over time to deliver time-controlled evolving scene descriptions. Then, we propose GLEN (Graph-Language Edit Network), a graph neural network trained for graph-language alignment and activity-conditioned graph edits predictions to model scene evolution. Finally, we introduce *Activity-conditioned Graph Edit Forecasting* (A-GEF), a novel formulation that models scene dynamics as a sequence of structured transformations over scene graphs conditioned on the ongoing actions, enabling explicit and interpretable reasoning about how scenes evolve over time. We validate the quality and completeness of our scene graph annotations on EgoSchema [7], showing that scene graph sequences provide rich semantic information. Then, we demonstrate that such representations can be exploited with minimal loss of information compared to raw videos, achieving competitive or state-of-the-art performance on retrieval tasks such as EgoMCQ [5] and EgoCVR [14], a challenging compositional video retrieval benchmark. Finally, GLEN achieves state-of-the-art results on reasoning tasks, including EXPLORE-Bench [15] and A-GEF, showing that explicit structured representations can effectively replace or complement black-box MLLMs.

Our results highlight how structured representations improve both language grounding and temporal reasoning, while enabling a new class of tasks centered on predicting and manipulating scene dynamics. More broadly, our work advocates for a shift toward structured, explicit representations in egocentric video understanding - moving from describing what is observed to reasoning about how actions transform the scene state.

2 Related Works

Video Scene Graph. Scene Graph Generation (SGG) aims to represent visual scenes as graphs where objects are the nodes and their relationships are the edges. Recent progress has been driven by transformer-based architectures, which enable end-to-end relational reasoning and have become a dominant paradigm in image-based SGG [16, 17]. Alongside architectural advances, substantial effort has addressed the long-tailed distribution of predicates through debiasing and knowledge-aware methods [18] and on scaling structured supervision. Synthetic Visual Genome [19] and its video extension [20] enable large-scale extraction of scene graphs from images and videos, improving data availability for relational learning. In parallel, large language models have been leveraged to improve weakly supervised SGG via better triplet extraction and alignment [21]. Extending SGG to videos introduces the challenge of modeling temporal dynamics. Action Genome [8] formalized video scene graph generation (VidSGG), followed by approaches using spatio-temporal transformers and relational reasoning for dynamic interactions [9, 10]. Recent work further improves robustness and expressivity through bias mitigation [22] and richer structures such as hierarchical and hypergraph representations [11, 23]. Beyond generation, scene graph anticipation methods predict future graphs from past observations, modeling temporal evolution directly in graph space [11, 12]. However, these approaches remain unconditioned, relying solely on past visual input and failing to model the human activity driving scene changes, limiting controllability over future outcomes. Recently, scene graphs have been adopted as structured representations for downstream video understanding tasks. For long-horizon reasoning tasks, graph-based representations enable more effective compositional and relational reasoning [24–27]. Recent approaches further explore graph-based memory and reasoning mechanisms for long-video understanding [11, 28, 29]. Despite their success, these methods primarily use scene graphs as auxiliary structures for reasoning or question answering, rather than as a unified representation of scene dynamics. In contrast, we treat scene graphs as a controllable and evolving state, introducing an activity-conditioned formulation where temporal changes are explicitly modeled through graph edit operations.

Graph Edit Models. Graph Edit Distance (GED) [30] is a classical framework for measuring dissimilarity between graphs by computing the minimum-cost sequence of edit operations, as node/edge insertions, deletions, and substitutions, required to transform one graph into another. Since the exact computation of GED is NP-hard, a range of approximations have been proposed, including heuristics strategies [31, 32], bipartite matching [33, 34], and learning-based methods [35–37]. Recently, Graph Edit Networks (GENs) [37] have reformulated graph transformation as a learning problem, where neural networks are trained to predict sequences of edit operations between graph pairs. In particular, GENs employ a linear output layer over node representations to produce local edit decisions, including node-level insertions, deletions, and attribute modifications. This formulation enables efficient inference of graph transformations through independent per-node predictions, and leverages learned representations to parameterize edit operations in a data-driven manner. However, existing GEN formulations primarily address unconditional graph-to-graph transformation, where edits are inferred solely from the source and target graphs. In contrast, our setting considers conditioned graph evolution, where graph transformations are explicitly influenced by high-level contextual information, requiring the model to adapt its edit decisions based on external conditions.

3 SG-Ego

Notation. We define a spatio-temporal scene graph as a directed multi-edge graph $G_i = (V_i, E_i, \mathbf{X}_i, \mathbf{Y}_i)$, where V_i is the set of object nodes and $E_i \subseteq V_i \times V_i$ is the set of directed edges representing pairwise relations. Each node $v \in V_i$ corresponds to an object instance in the scene, to which it is associated a set of attributes $\mathbf{X}_i \in \mathbb{R}^{|V_i| \times d}$, including its visual features (extracted from the object crop), the corresponding bounding box, and a semantic label describing the category of the object. Each edge $(u, v) \in E_i$ captures a spatial or functional relation from u to v and it is associated with a set of attributes $\mathbf{Y}_i \in \mathbb{R}^{|E_i| \times r}$ comprising the semantic label describing the relation. Each scene graph G_i covers a variable temporal window, summarizing all the relations within the window. In particular, we define G_t as the spatial scene graphs associated with frame t and $G_{t:t+T}$ as the spatio-temporal scene graph associated with the temporal window from frame t to $t+T$.

Dataset Collection. We collect SG-Ego, a large scale annotation set of spatio-temporal scene graphs from videos of human activities, based on the Ego4D [3] dataset, consisting of 3.8M spatio-temporal scene graphs annotated through a *training-free* pipeline. We define a set of $N_{obj} = 1480$ node labels and $N_{rel} = 387$ relations labels. Each sample spans a temporal window $[t, t+T]$ and is associated with the initial spatial scene graph G_t, a spatio-temporal scene graph over the entire temporal window $G_{t:t+T}$ and a textual narration act_t. For Graph-Text Alignment, we define a training split consisting of 3.8M spatio-temporal scene graphs (*SG-Ego Align*). For A-GEF, we define a training set and a validation set of approximately 360k and 7.2k tuples $(G_t, act_t, G_{t:t+T})$. We name this split *SG-Ego Edit*. These samples are taken from a broad set of videos, following the same scenario distribution of the Ego4D dataset to ensure diversity.

3.1 Annotation Pipeline

SG-Ego implements a *training-free* VidSGG strategy divided into three stages. Our pipeline maps a sequence of T frames into a single temporally consistent spatio-temporal scene graphs $G_{t:t+T}$, capturing all the spatial and functional relations in the video. We follow a bottom-up approach to extract a rich set of frame-level relations, ground them with a open vocabulary detection model and build a temporally consistent scene graph of the temporal window. Here, we provide a description of the annotation process, see Sec. A for further implementation details and qualitative visualizations.

Stage 1: Frame-level relations extraction. Given a video $\mathcal{V}$, we first extract frames at a constant sampling rate $\tau = 5$ fps. Then, we prompt a MLLM (Qwen3.5-9B [38]) to describe the spatial and functional relations appearing in the frame, avoiding unnecessary details related to the appearance of the object or the environments. We instruct the model to directly produce *(subject_x, relation, object_y)* triplets, where the *_x* and *_y* track the specific instances of the entities. Then, we use rule-based filtering to remove malformed or duplicated triplets. For each frame, the captioning pipeline outputs a list of candidate *frame-level* triplets $\mathcal{T}_i$.

Stage 2: Frame-level relations grounding. We use GroundingDINO [39] to ground each candidate frame-level triplet $\mathcal{T}_i$ in the image. Specifically, we concatenate the subject, predicate, and object into a single sentence and feed it to the detector. We then extract the queries aligned with the textual spans corresponding to the subject and object, and use them to obtain candidate detections for each entity in the triplet. We use the tracking suffixes from the captions to merge multiple detections from the same object, and apply a set of heuristics to drop duplicates and filter invalid spatial relations (*e.g.*, *to the right*, *inside*, etc...). The resulting objects and relations define the nodes and edges of the frame-level graph, respectively. From the valid detections, we extract the node attributes consisting of the bounding box and semantic label of the object. Objects and relations from the captions are mapped to a fixed set of $N_{obj} = 1480$ object and $N_{rel} = 387$ classes respectively. The output of the grounding stage at frame t is a frame-level spatial scene graph $G_t = (V_t, E_t, \mathbf{X}_t, \mathbf{Y}_t)$.

Stage 3: Graph consolidation across frames. Each frame-level scene graph provides a partial view of the ongoing interactions in the video, due to occlusions or omitted triplets by the captioner. Given a sequence $G_t, \ldots, G_{t+T}$, the consolidation process $\mathcal{C}$ merge them into a single spatio-temporal scene graph that includes all nodes and relations observed during the timespan [40]. Objects correspondences are established via a semantic tracking pipeline based on SAM2 [41] and DINOv2 [42]. More specifically, starting from a pair of graphs G_t and G_{t+1}, we extract object masks with SAM2 and propagate them from G_t to G_{t+1}. We compute IoU between propagated and detected masks at each frame, and use Hungarian matching to associate tracked and detected objects with IoU > 0.5. To improve robustness, we additionally match objects based on DINOv2 feature similarity when tracking fails. The merged graph includes all matched and unmatched nodes and inherits relations from both graphs. This process is repeated sequentially over time to obtain the final consolidated graph $G_{t:t+T}$.

4 Method

We formulate the dynamic evolution of a scene as a novel setting called *activity-conditioned graph edit forecasting* (A-GEF), in which we ask a model to forecast how a spatial scene graph G_t will change based on the activity described in act_t. To do this, we consider a *conditional* formulation in

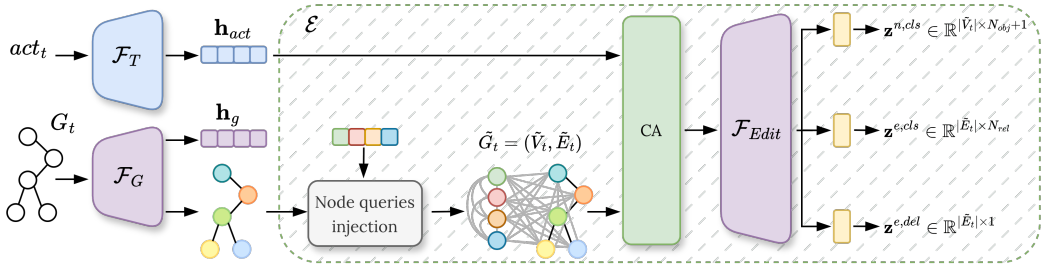


Figure 2: **The GLEN architecture for Activity-conditioned Graph Edit Forecasting (A-GEF).** The model takes a spatial graph G_t and a conditioning action act_t and predicts the necessary edit to evolve the scene graph into the consolidated $G_{t:t+T}$. For Graph-Text Alignment (GTA), we extract a graph embedding from $G_{t:t+T}$ and align it with the text embedding of act_t.

which future graph edits are predicted given both the current graph G_t and the upcoming activity act_t. Our approach Graph-Language Edit Network (GLEN), see Fig. 2, consists of three main components. The **Graph Encoder** $\mathcal{F}_G : \mathcal{G} \mapsto \mathbb{R}^d$ maps from the space of scene graphs $\mathcal{G}_t$ to d-dimensional embeddings, while the **Text Encoder** $\mathcal{F}_T : \mathcal{T} \mapsto \mathbb{R}^d$ maps act_t from the text space $\mathcal{T}$ into the same embedding space. By leveraging a Graph-Text Alignment (GTA) objective, the model can learn a versatile representation that summarizes the state of the scene and enable spatially grounded reasoning about the objects and relations in the scene. Finally, the **Graph Edit Model** $\mathcal{E} : (\mathcal{G}, \mathcal{T}) \mapsto \mathcal{G}$ defines text-conditioned transformations over the scene graph $\mathcal{G}_t$, allowing structured modifications based on the given activity description.
We represent an action as a triplet $(G_t, act_t, G_{t:t+T})$, where G_t is the initial scene graph at timestamp t, act_t is a textual description of the activity starting at t, and $G_{t:t+T}$ is the final consolidated scene graph after the activity concludes. Given such triplets, GLEN is trained to (i) align the embedding of $G_{t:t+T}$ with the embedding of the action textual description act_t via an alignment objective (GTA) or (ii) transform G_t into the final graph $G_{t:t+T}$ through the Graph Edit Model $\mathcal{E}$ (A-GEF).

4.1 Graph-Text Alignment

We formulate the Graph-Text Alignment task as the problem of learning a joint embedding space in which a spatio-temporal scene graph is aligned with the textual description of the activities occurring within the corresponding temporal window. We train GLEN with *Graph-Text Contrastive Alignment (GTCA)* and *Graph-Text Matching (GTM)*. Our goal is to learn a versatile representation that summarizes the scene graph in a single text-aligned embedding, enabling grounded reasoning about the spatial and functional relations in the scene.

Embeddings extraction. We define a Graph Encoder $\mathcal{F}_G : \mathcal{G} \mapsto \mathbb{R}^d$ that maps from the space of scene graphs $\mathcal{G}$ into d-dimensional embeddings, and a Text Encoder $\mathcal{F}_T : \mathcal{T} \mapsto \mathbb{R}^d$ that maps the textual narrations into the same embedding space. For a scene-graph activity pair $(G_{t:t+T}, act_t)$, we define their corresponding embeddings as $\mathbf{h}_g$ and $\mathbf{h}_{act}$. $\mathcal{F}_G$ is composed of L TripletGCN layers [43], where each layer at depth l updates the node embeddings $\mathbf{x}_j^{(l)}$ through message passing:

$$\mathbf{x}_j^{(l+1)} = \mathbf{x}_j + \phi_2(\underset{i \in \text{Neigh}(j)}{\text{mean}} \phi_1(\mathbf{x}_i || \mathbf{y}_{i \to j} || \mathbf{x}_j)), \tag{1}$$

where ϕ_1 and ϕ_2 are MLPs and Neigh(j) is the set of neighbors of node j. A similar update rule applies to the edge embeddings $\mathbf{y}_i$. By jointly updating node and edge features, the Graph Encoder learns context-aware representations that capture the spatial and semantic structure of the current scene. We alternate message passing and cross attention layers CA to update the node embeddings $\mathbf{X}^{(l)}$ with the text tokens $\mathbf{H}_{act}$ from the narrations:

$$\tilde{\mathbf{X}}^{(l+1)} = LN\left[\mathbf{X}^{(l+1)} + \alpha \text{CA}(\mathbf{X}^{(l+1)}, \mathbf{H}_{act})\right], \tag{2}$$

where LN is the layer norm operator and α is a learnable parameter that controls the contribution of the CA. Finally, the encoder outputs a single graph embedding $\mathbf{h}_{g,i}$ by average pooling all the node $\mathbf{X}^{(L)}$ and edge $\mathbf{Y}^{(L)}$ embeddings at the final layer, followed by a MLP projection. Depending on the task, we enable or disable the cross attention modules. For Graph-Text Alignment, we train only the Graph Encoder $\mathcal{F}_G$ to align the graph embedding of $G_{t:t+T}$ to the textual embedding of act_t.

Graph-Text Contrastive Alignment (GTCA). Given a batch $\mathcal{B} = \{(G, act)_i\}_{i=1}^{|\mathcal{B}|}$, consisting of $|\mathcal{B}|$ scene graphs paired with their corresponding textual narrations, we push the model to map close in the embedding space paired graph-text samples, while pushing unpaired samples further apart. We define the batches such that there are at least two samples from the same video to make the model more robust to subtle variations in the scene graphs extracted from the same video. Formally, we define the graph-to-text contrastive alignment loss $\mathcal{L}_{g2t}$ as:

$$\mathcal{L}_{g2t} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{1}{|\mathcal{P}_i|} \sum_{k \in \mathcal{P}_i} \log \frac{\exp(\mathbf{h}_{g,i}^T \mathbf{h}_{act,k}/\tau)}{\sum_{n \in \mathcal{N}_i} \exp(\mathbf{h}_{g,i}^T \mathbf{h}_{act,n}/\tau)}, \tag{3}$$

where $\mathcal{P}_i$ is the set of positive samples for the anchor sample i, defined as the samples from the batch that share at least one verb and noun labels from the narration act_i, and $\mathcal{N}_i$ is the set of negative samples from the same video as sample i. We define the symmetric *text-to-graph* contribution $\mathcal{L}_{t2g}$ in a similar way. For GTCA, we disable the Cross Attention layers, keeping the two encoders separate.

Graph-Text Matching (GTM). To encourage the network to distinguish subtle differences in the scene graphs, we introduce an additional Graph-Text Matching objective [6, 44]. Specifically, we mine hard positive pairs and negative pairs from the batch based on the similarity of their graph and text embeddings, enable the CA layers, and feed them to the encoders. The resulting embeddings are then passed to an MLP to predict whether the pair is positive or negative.

4.2 Activity-conditioned Graph Edit Forecasting (A-GEF)

Problem formulation. Given a sequence of graphs $\{G_t, G_{t+1}, \ldots, G_{t+T}\}$, a Graph Edit Network takes as input a spatial graph G_t and predict a set of edits that transform it into G_{t+1} [37]. A graph edit function $\delta : \mathcal{G} \to \mathcal{G}$ is defined as a composition of elementary edit operations which includes node deletion (removing node i and its incident edges), node insertion (adding a new node with attribute x), node replacement (updating the attribute of node i to x), edge deletion (removing edge (i, j)), and edge insertion (adding edge (i, j)). These operations form the basis of the graph edit formulation. In our setting, we extend this formulation to *activity-conditioned graph edit forecasting*. Given the initial scene graph G_t and a textual description of the activity act_t starting at timestamp t, the goal is to predict the final scene graph $G_{t:t+T}$. Unlike standard graph edit networks [37], which model graph evolution purely as a function of the current graph, in our case the dynamic evolution of the graph is conditioned by the activity. This conditioning is crucial in our setting, as scene evolution is inherently driven by human actions and the same scene graph will likely evolve differently depending on the activity being performed. Differently from the standard formulation, we do not explicitly model node replacement operations. Instead, we restrict the edit space to node deletions and node insertions, and express changes in node attributes through a delete-and-insert mechanism.

The Graph Edit Model. We design a Graph Edit Model $\mathcal{E} : (\mathcal{G}, \mathcal{T}) \mapsto \mathcal{G}$ that predicts graph transformations explicitly conditioned on the future activity. We first encode the initial graph G_t using the Graph Encoder $\mathcal{F}_G$ described in Sec. 4.1, using Eq. 2 with $\alpha = 0$, obtaining node $\mathbf{Z}_t^n$ and edge $\mathbf{Z}_t^e$ embeddings. For the editing objective, the cross-attention layers between node embeddings and text tokens and the graph pooling are disabled; therefore, both node and edge representations correspond directly to the output of the L-layer TripletGCN module. We then augment the graph structure by adding a fixed number K of learnable node queries, which act as candidate slots for potential future objects. These queries are initialized as learnable embeddings and appended to the set of node features, taking inspiration from [17]. Next, we construct a fully connected graph over this augmented node set. Edges involving the newly introduced query nodes and not previously connected old nodes are initialized with noisy features, providing no prior structural bias and allowing the model to freely infer their connectivity and semantics. After this step, we obtain a noised graph $\tilde{G}_t = (\tilde{V}_t, \tilde{E}_t, \tilde{\mathbf{Z}}_t^n, \tilde{\mathbf{Z}}_t^e)$, with $\tilde{V}_t = |V_t| + K$ nodes and $|\tilde{V}_t| \times |\tilde{V}_t|$ edges.

We condition the augmented graph on the text embedding of the action $\mathbf{h}_{act} = \mathcal{F}_T(act_t)$ through a Cross-Attention layer, allowing both node and edge features to be conditioned by the action:

$$\tilde{\mathbf{Z}}_t^{n,cond} = \tilde{\mathbf{Z}}_t^n + CA(\tilde{\mathbf{Z}}_t^n, \mathbf{h}_{act}), \qquad \tilde{\mathbf{Z}}_t^{e,cond} = \tilde{\mathbf{Z}}_t^e + CA(\tilde{\mathbf{Z}}_t^e, \mathbf{h}_{act}). \tag{4}$$

This allows the model to modulate the entire graph based on the conditioning action, effectively injecting activity-dependent signals into the relational structure. The conditioned graph is then

processed by a second graph encoder, called Graph Edit Encoder $\mathcal{F}_{Edit}$, built as a stack of TripletGCN layers, which propagates the activity-aware information via edge-aware message passing. The graph encoder $\mathcal{F}_{Edit}$ maps $\tilde{G}_t$ into a set of graph edits. Specifically, the node embeddings at the last layer of the model $\mathbf{Z}_t^n$ are fed to a *node update head*, which outputs a object category score $y_i \in \mathbb{R}^{N_{obj}+1}$. This score assigns each node to one of N_{obj} categories or to a special *no object* class. Assigning a node from G_t to the *no object* class corresponds to a *node deletion*, while assigning a valid class to a query node corresponds to node insertion with a specific node label. Edge edits are processed through two separate heads to predict the edge deletion score $z_i^{del} \in \mathcal{R}^1$ and its semantic labels $z_i^{cls} \in \mathcal{R}^{N_{rel}}$, possibly more than one. This jointly captures *edge deletion* (by suppressing existing edges) and *edge insertion* (by activating edges involving query nodes with predicted relation labels).

We train the network by directly supervising graph edit operations. Ground-truth correspondences are established between nodes in G_t and $G_{t:t+T}$ using visual features and node labels with a one-to-one mapping, while the additional K query nodes are matched via Hungarian assignment based on label costs. Unmatched predictions are assigned to the *no object* class. Based on these matches, we supervise edge updates and deletions by comparing predicted edges with ground-truth edges between matched node pairs. Given directed multi-edge graphs, edge classification is modeled as a multi-label task over N_{rel} classes per node pair. Edge deletion is formulated as a binary task indicating whether at least one edge exists between two nodes. We use cross-entropy for node classification and binary cross-entropy for edge update and deletion.

5 Experiments

We validate SG-Ego and GLEN on several benchmarks. On EgoSchema [7], we demonstrate that the proposed scene graph annotations provide a structured and complete representation of the scene that can support long range reasoning about human activities. On EgoMCQ [5] and EgoCVR [14], we show that the graph embeddings produced by GLEN are well aligned with the textual description of the corresponding ongoing human activities. Finally, on A-GEF and EXPLORE-Bench, we assess the capability of GLEN to predict the dynamic evolution of a scene given an activity.

Implementation details. For the Graph-Text Alignment tasks (EgoMCQ and EgoCVR), we extract the visual and semantic embeddings of nodes and edges using the Perception-Encoder [45] backbone. For A-GEF and EXPLORE-Bench, we use DINOv2 [42] for the visual embeddings and MiniLM [46] for the semantic embeddings. We initialize the Text Encoder $\mathcal{F}_T$ from the EgoVLP [5] text encoder and keep it frozen. We use three and two TripletGCN layers for $\mathcal{F}_G$, and $\mathcal{F}_{Edit}$, respectively. The embedding size is set to $d = 512$ for the $\mathcal{F}_G$ and $d = 256$ for $\mathcal{F}_{Edit}$. For GTA, we mine $n = 3$ negative samples for each anchor in the batch. For A-GEF, we set $K = 128$ node queries. Ablations are reported in Sec. B. The backbones used to initialize the node and edge embeddings are kept frozen and we train only the Graph Encoders and a small projection head on top of the Text Encoder. We train our models on a single A100 GPU for approx. 24 hours.

5.1 Long-Range Reasoning and Summarization on EgoSchema

The scene graphs produced by SG-Ego capture the relevant spatial and functional interactions in the scene and provide a compact, compositional and editable summary of the ongoing activities happening in the video. We validate the quality and completeness of our scene graphs on EgoSchema [7], a diagnostic benchmark used to evaluate long-horizon temporal, causal, and intent reasoning over first-person video sequences, by providing a Large Language Model (Qwen3.5-9B [38]) uniformly sampled scene graphs from the video, results as shown in Table 1(left). Specifically, we convert the graph triplets into textual form and prompt the model to answer questions based solely on the scene state described by these graphs, either using only object and relation labels (*SG-Ego semantic*) or also providing the visual crop of each node (*SG-Ego full*). Semantic graphs alone already reach 64.6%, and the full graphs achieve performance close to the model with full access to the raw frames (71.6% vs. 72.8%). This is particularly notable since full graphs (*SG-Ego full*) retain only localized object crops rather than the complete visual scene, making questions that require detailed visual recognition challenging. When combined with frames, scene graphs further improve over the video-only setting (74.0%, +1.2%). These results show that scene graphs are not a replacement for raw visual input, but rather a structured complementary representation that preserves useful semantic and visual information while explicitly modeling objects and interactions.

Table 1: **Long-Range Reasoning on EgoSchema [7] and Graph-Text Alignment on EgoMCQ [5].** On EgoSchema, we validate the quality and completeness of SG-Ego. On EgoMCQ, we show that the graph embeddings from GLEN are well-aligned with the caption of the ongoing activities.

Method	EgoSchema (val) Acc.
EgoThinker [47]	71.8
Qwen3.5-9B (Blind)	38.2
Qwen3.5-9B (Frames)	72.8
Qwen3.5-9B (SG-Ego semantic)	64.6
Qwen3.5-9B (SG-Ego full)	71.6
Qwen3.5-9B (Frames + SG-Ego semantic)	73.2
Qwen3.5-9B (Frames + SG-Ego full)	**74.0**

Method	Modalities	EgoMCQ Acc.	
		Inter	**Intra**
EgoVLP [5]	Video / Text	90.6	57.2
HierVL [48]	Video / Text	90.5	52.4
EgoVLPv2 [6]	Video / Text	91.0	**60.9**
HiERO (EgoVLP) [49]	Video / Text	**91.6**	<u>59.6</u>
GLEN (Perc. Enc.)	Graph / Text	<u>91.2</u>	56.2

Table 2: **Composed Video Retrieval on EgoCVR**, a retrieval benchmark in which each sample includes a query clip, an instruction with the desired modification, and a set of candidate clips to choose from. We measure performance in terms of Recall@k. Candidate clips are either from the same (*local*) or different videos (*global*) as the query.

Method	Global			Local		
	R@1	R@5	R@10	R@1	R@2	R@3
Random	0.01	0.05	0.1	25.3	38.2	50.7
CLIP	7.4	33.2	55.3	26.1	43.4	57.7
BLIP$_{CoVR}$ [50]†	5.4	15.2	24.3	33.1	49.5	62.9
BLIP$_{CoVR-ECDE}$ [51]†	6.0	14.8	24.3	33.4	49.3	63.0
CIReVL [52]	2.0	6.8	10.2	21.6	35.1	46.0
TFR-CVR [14]	14.1	39.5	54.4	44.2	61.0	73.2
Thawakar *et al.* [53]†	14.6	**41.3**	54.9	44.8	61.7	74.0
GLEN	**16.0**	40.1	**56.9**	**46.1**	**64.6**	**76.7**

† Finetuned on WebVid-CoVR or Dense-WebVid-CoVR for CVR.

5.2 Graph-Text Alignment on EgoMCQ

EgoMCQ [5] consists of 39k multiple-choice *text-to-video* questions where, given a textual narration, the model has to select the correct video clip between five candidates. We report Inter-Video accuracy, where clips are from different videos, and Intra-Video accuracy, where they are from the same video in Table 1(right). We achieve performance comparable to video-language models *end-to-end* fine-tuned for multimodal contrastive alignment, despite relying on a significantly more compact and structured representation of the video input.

5.3 Composed Video Retrieval on EgoCVR

In Composed Video Retrieval (CVR), the model is tasked to retrieve a video that matches a reference clip after applying a textual modification, *i.e.*, given a source video and a query describing a change in its content, the model must identify the correct video among a set of candidates that matches the modified scene. We approach this task by directly leveraging the graph–text alignment branch of GLEN. Unlike EgoMCQ, which is focused on action-level dynamics, CVR requires more fine-grained reasoning about the compositional structure of the scene and how the instruction query changes it. Notably, our approach is zero-shot as we do not train or fine-tune for this downstream task.

We validate GLEN on EgoCVR [14], a CVR benchmark with 2295 samples, each consisting of a *query clip*, an *instruction* representing the intended modification and a set of *candidate clips* to choose from. We follow the original EgoCVR formulation, extracting a text embedding from the query clip caption augmented with the modification instructions. Then, we seek for the best match among the available candidates through cosine similarity with their graph embeddings. To preserve visual consistency between the query and candidate clips, which can otherwise be overshadowed by semantic similarity, we follow TFR-CVR [14] and introduce a preliminary visual filtering step that selects the top-n_c candidates ($n_c = 15$) based on similarity to the reference clip. We chose to implement it with CLIP features, ensuring generality without introducing task-specific bias. Results in Table 2 show that GLEN achieves state-of-the-art results, outperforming even methods specifically trained for CVR. We argue that the scene graph based structure of GLEN allows the model to better reason about the spatial and functional relations in the input clips, which is crucial for reasoning about the fine-grained modifications induced by the query.

Table 3: **Action-conditioned Graph Edit Forecasting (A-GEF) on SG-Ego**. We compare GLEN with Qwen3.5 by providing the model with the textual triplets list and asking it to predict the future triplets, and with a static baseline that replicates G_t as the final $\hat{G}_{t:t+T}$ prediction. Our approach outperforms both Qwen and the static baseline, demonstrating that graph edits are an effective way to capture scene evolution. LoRA: fine-tuned on the A-GEF training set.

Method	Triplet Recall		
	R@20	R@50	R@100
Qwen3.5-9B (text only) [38]	9.19	9.19	9.19
Qwen3.5-9B (text + start frame) [38]	15.30	15.30	15.30
Qwen3.5-9B (LoRA, text + start frame) [38]	22.70	22.78	22.78
G_t (*static*)	25.94	25.94	25.94
GLEN	**36.69**	**45.39**	**49.91**

5.4 Scene Evolution on A-GEF benchmark

We evaluate the scene evolution capabilities of GLEN on A-GEF built on SG-Ego, see Sec. 4.2. Following standard scene graph evaluation protocols, we evaluate the quality of the predicted graph $\hat{G}_{t:t+T}$ using triplet $Recall@K$ over $(subject, relation, object)$ tuples. We introduce an entropy-based filtering strategy to handle uncertainty in node generation, where we remove nodes for which predicted class entropy exceeds a threshold of 0.5. This reduces noise from ambiguous object assignments and provides a more reliable evaluation on confident predictions. Following standard scene graph evaluation protocols, we evaluate the quality of the predicted graph $\hat{G}_{t:t+T}$ using triplet $Recall@K$. We introduce an entropy-based filtering strategy to handle uncertainty in node generation, where we remove nodes for which predicted class entropy exceeds a threshold of 0.5. This reduces noise from ambiguous object assignments and provides a more reliable evaluation on confident predictions. We compare GLEN with two families of baselines: a *static* approach that replicates G_t as the $\hat{G}_{t:t+T}$ prediction, and baselines based on Qwen3.5-9B [38]. For the latter, we provide the model with the list of triplets of the start graph in textual form, the narration of the activity and the lists of possible objects and relations, and ask it to predict the triplets of the target scene.
We evaluate three settings: zero-shot with text only, zero-shot with the start frame as an additional visual input, and with the start frame after LoRA fine-tuning on the A-GEF training set. As shown in Table 3, the start frame helps the LLM (15.30 vs. 9.19 R@20) and fine-tuning further improves it (22.70), but all the LLM settings remain below the static baseline. In all settings the LLM predicts few triplets, which leaves the recall almost unchanged at different K values. GLEN consistently outperforms all the baselines, including the fine-tuned LLM (36.69 vs. 22.70 R@20), demonstrating the effectiveness of explicitly modeling activity-conditioned graph edits for capturing scene evolution. Notably, the gap with the static baseline highlights that a large portion of scene dynamics cannot be explained by persistence alone, but requires modeling how objects and relations change under actions. At the same time, the improvement over the LLM baselines, even when trained on the same data, suggests that operating in a structured graph space provides a stronger inductive bias than purely language-driven predictions. Overall, these results validate our formulation of scene evolution as a sequence of activity-conditioned graph transformations, enabling more accurate and interpretable predictions of future scene structure.

5.5 Long Horizon Reasoning on EXPLORE-Bench

We evaluate the generalization capabilities of our activity-conditioned graph edit forecasting framework on EXPLORE-Bench [15], a benchmark designed to assess long-horizon reasoning over sequences of actions. The task requires predicting the final scene resulting from the sequential application of atomic actions to an initial state. We cast this problem within our framework by first extracting a scene graph from the initial scene (Sec. 3.1), and conditioning GLEN on the sequence of actions to predict the resulting scene graph, which serves as a structured representation of the final state. Notably, we do not train our model on this benchmark. Instead, we directly apply the model trained on A-GEF, demonstrating its ability to transfer to long-horizon reasoning without any task-specific supervision. This is particularly significant because such long-horizon compositional reasoning tasks are typically addressed using MLLMs only. In contrast, we show that a structured, graph-based formulation alone is sufficient to model long multi-step scene evolution under actions.
We follow the evaluation protocol of [15], measuring both object and relation level accuracy. For

Table 4: **Long horizon reasoning on EXPLORE-Bench.** Starting from an initial frame, the model must predict the final scene after a sequence of actions. Object predictions are evaluated in terms of recall ($\boldsymbol{S_{obj}}$), while relations are evaluated with an LLM-based scoring function ($\boldsymbol{S_{rel}}$). † Reproduced results using the official protocol.

Methods	Short		Medium		Long		Full	
	$\boldsymbol{S_{obj}}$	$\boldsymbol{S_{rel}}$	$\boldsymbol{S_{obj}}$	$\boldsymbol{S_{rel}}$	$\boldsymbol{S_{obj}}$	$\boldsymbol{S_{rel}}$	$\boldsymbol{S_{obj}}$	$\boldsymbol{S_{rel}}$
GPT-5.2-Chat	59.91	2.70	59.88	2.65	58.06	2.61	59.69	2.67
Gemini-3-Pro	61.29	2.77	60.99	2.74	59.17	2.70	60.94	2.75
LLaVA-OneVision-1.5-8B	53.25	2.51	51.21	2.44	47.62	2.41	51.87	2.47
Qwen3-VL-8B-Instruct	61.34	2.84	60.78	**2.81**	56.83	**2.71**	60.63	**2.82**
Qwen3-VL-8B-Thinking	63.77	**2.85**	62.61	2.78	58.02	2.63	62.70	2.80
Qwen3.5-9B-Instruct†	58.73	2.74	56.20	2.63	51.21	2.55	56.95	2.68
GLEN	**66.12**	2.67	**66.71**	2.73	**59.37**	2.67	**65.59**	2.69

object evaluation, Sentence-BERT embeddings are used to compute a similarity matrix between predicted and ground-truth objects, yielding a soft coverage score $\boldsymbol{S_{obj}}$. For relations, a standardized prompting strategy is used to guide a large language model in scoring predicted relations on a 0–5 scale against ground-truth annotations, resulting in the $\boldsymbol{S_{rel}}$ metric.

As shown in Table 4, our method achieves state-of-the-art performance on $\boldsymbol{S_{obj}}$ and competitive results on $\boldsymbol{S_{rel}}$ compared to MLLMs baselines. Since Qwen3.5-9B [38] is used in the first stage of our pipeline (Sec. 3.1) for scene captioning and description, we also evaluate it directly on EXPLORE-Bench. GLEN improves S_{obj} over Qwen3.5-9B by 8.6 points on the full benchmark (65.59 vs. 56.95), with comparable S_{rel} (2.69 vs. 2.68), showing that the gains do not come from the choice of the underlying MLLM, but from explicit reasoning over a structured representation. These results highlight the strong generalization ability of our structured, activity-conditioned formulation for modeling long-horizon scene evolution, and demonstrate that graph-based reasoning provides a viable alternative to language-based approaches for compositional temporal understanding.

6 Future directions

Our formulation models evolving scenes as editable graphs, enabling explicit reasoning over how actions transform the state of the environment. While this is valuable for human activity understanding, a particularly promising direction lies in embodied-AI and especially robotics. By extending GLEN with spatial prediction, it naturally becomes a structured scene transition model particularly suited to let a robotic agent reason not only about *what* changes, but also about *where* such changes occur and *why*. This provides a direct bridge between perception and action, supporting grounded planning and control through explicit predictions of future scene states, and may also serve as rich intermediate representation for visual policy learning trained from graph-based expert demonstrations.

7 Conclusion

We introduce a new paradigm for egocentric video understanding by modeling scenes as structured graphs and their evolution as activity-conditioned graph edits. This enables explicit reasoning over how actions transform the environment, unifying representation learning, language alignment, and temporal prediction within a single framework. Our results show that operating in structured space improves language grounding and temporal reasoning, while providing interpretability and controllability. More broadly, this work highlights editable, structured representations as a promising foundation bridging perception and action.

Acknowledgments

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support.

References

[1] Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. Remembr: Building and reasoning over long-horizon spatio-temporal memory for robot navigation. In *ICRA*, 2025.

[2] Yue Fan, Xiaojian Ma, Rongpeng Su, Jun Guo, Rujie Wu, Xi Chen, and Qing Li. Embodied videoagent: Persistent memory from egocentric videos and embodied sensors enables dynamic scene understanding. In *ICCV*, 2025.

[3] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

[4] Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. Egotaskqa: Understanding human tasks in egocentric videos. In *NeurIPS*, 2022.

[5] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z XU, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. In *NeurIPS*, 2022.

[6] Shraman Pramanick, Yale Song, Sayan Nag, Kevin Qinghong Lin, Hardik Shah, Mike Zheng Shou, Rama Chellappa, and Pengchuan Zhang. Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. In *ICCV*, 2023.

[7] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023.

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

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

[10] Jingkang Yang, Wenxuan Peng, Xiangtai Li, Zujin Guo, Liangyu Chen, Bo Li, Zheng Ma, Kaiyang Zhou, Wayne Zhang, Chen Change Loy, et al. Panoptic video scene graph generation. In *CVPR*, 2023.

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

[12] Rohith Peddi, Saksham Singh, Saurabh, Parag Singla, and Vibhav Gogate. Towards scene graph anticipation. In *ECCV*, 2024.

[13] Antonio Alliegro, Francesca Pistilli, Tatiana Tommasi, and Giuseppe Averta. Forescene: Forecasting human activity via latent scene graphs diffusion. *arXiv preprint arXiv:2503.06182*, 2025.

[14] Thomas Hummel, Shyamgopal Karthik, Mariana-Iuliana Georgescu, and Zeynep Akata. Egocvr: An egocentric benchmark for fine-grained composed video retrieval. In *ECCV*, 2024.

[15] Chengjun Yu, Xuhan Zhu, Chaoqun Du, Pengfei Yu, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Explore-bench: Egocentric scene prediction with long-horizon reasoning. *arXiv preprint arXiv:2603.09731*, 2026.

[16] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *CVPR*, 2022.

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

[18] Zhanwen Chen, Saed Rezayi, and Sheng Li. More knowledge, less bias: Unbiasing scene graph generation with explicit ontological adjustment. In *WACV*, 2023.

[19] Jae Sung Park, Zixian Ma, Linjie Li, Chenhao Zheng, Cheng-Yu Hsieh, Ximing Lu, Khyathi Chandu, Quan Kong, Norimasa Kobori, Ali Farhadi, et al. Synthetic visual genome. In *CVPR*, 2025.

[20] Ziqi Gao, Jieyu Zhang, Wisdom Oluchi Ikezogwo, Jae Sung Park, Tario G You, Daniel Ogbu, Chenhao Zheng, Weikai Huang, Yinuo Yang, Winson Han, et al. Synthetic visual genome 2: Extracting large-scale spatio-temporal scene graphs from videos. *arXiv preprint arXiv:2602.23543*, 2026.

[21] 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. In *CVPR*, 2024.

[22] Sayak Nag, Kyle Min, Subarna Tripathi, and Amit K Roy-Chowdhury. Unbiased scene graph generation in videos. In *CVPR*, 2023.

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

[24] Jianguo Mao, Wenbin Jiang, Xiangdong Wang, Zhifan Feng, Yajuan Lyu, Hong Liu, and Yong Zhu. Dynamic multistep reasoning based on video scene graph for video question answering. In *HLT-NAACL*, 2022.

[25] Ivan Rodin, Antonino Furnari, Kyle Min, Subarna Tripathi, and Giovanni Maria Farinella. Action scene graphs for long-form understanding of egocentric videos. In *CVPR*, 2024.

[26] Haiyi Qiu, Minghe Gao, Long Qian, Kaihang Pan, Qifan Yu, Juncheng Li, Wenjie Wang, Siliang Tang, Yueting Zhuang, and Tat-Seng Chua. Step: Enhancing video-llms' compositional reasoning by spatio-temporal graph-guided self-training. In *CVPR*, 2025.

[27] Linlin Zong, Xinyu Zhai, Xinyue Liu, Wenxin Liang, Xianchao Zhang, and Bo Xu. Structuring video semantics with temporal triplets for zero-shot video question answering. In *CIKM*, 2025.

[28] Sergey Linok, Vadim Semenov, Anastasia Trunova, Oleg Bulichev, and Dmitry Yudin. Dygenc: Encoding a sequence of textual scene graphs to reason and answer questions in dynamic scenes. *Technologies*, 2026.

[29] Tatiana Zemskova, Solomon Andryushenko, Ilya Obrubov, Viktoriia Khoruzhaia, Ekaterina Eroshenko, Ekaterina Derevyanka, and Dmitry Yudin. Focusgraph: Graph-structured frame selection for embodied long video question answering. *arXiv preprint arXiv:2603.04349*, 2026.

[30] Alberto Sanfeliu and King-Sun Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 1983.

[31] Michel Neuhaus, Kaspar Riesen, and Horst Bunke. Fast suboptimal algorithms for the computation of graph edit distance. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 2006.

[32] Kaspar Riesen, Stefan Fankhauser, and Horst Bunke. Speeding up graph edit distance computation with a bipartite heuristic. In *MLG*, 2007.

[33] Kaspar Riesen and Horst Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision computing*, 2009.

[34] Francesc Serratosa. Fast computation of bipartite graph matching. *Pattern Recognition Letters*, 2014.

[35] Yunsheng Bai, Hao Ding, Song Bian, Ting Chen, Yizhou Sun, and Wei Wang. Simgnn: A neural network approach to fast graph similarity computation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, 2019.

[36] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In *ICML*, 2019.

[37] Benjamin Paassen, Daniele Grattarola, Daniele Zambon, Cesare Alippi, and Barbara Hammer. Graph edit networks. In *ICLR*, 2021.

[38] Qwen3.5: Towards native multimodal agents, 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

[39] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, 2024.

[40] Sanghyeok Chu, Seonguk Seo, and Bohyung Han. Fine-grained captioning of long videos through scene graph consolidation. *arXiv preprint arXiv:2502.16427*, 2025.

[41] 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 Dollar, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *ICLR*, 2025.

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[43] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions. In *CVPR*, 2020.

[44] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[45] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Abdul Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Shang-Wen Li, Piotr Dollar, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network. In *NeurIPS*, 2025.

[46] Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021.

[47] Baoqi Pei, Yifei Huang, Jilan Xu, Yuping He, Guo Chen, Fei Wu, Yu Qiao, and Jiangmiao Pang. Egothinker: Unveiling egocentric reasoning with spatio-temporal cot. *arXiv preprint arXiv:2510.23569*, 2025.

[48] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. Hiervl: Learning hierarchical video-language embeddings. In *CVPR*, 2023.

[49] Simone Alberto Peirone, Francesca Pistilli, and Giuseppe Averta. Hiero: understanding the hierarchy of human behavior enhances reasoning on egocentric videos. In *ICCV*, 2025.

[50] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. CoVR: Learning composed video retrieval from web video captions. In *AAAI*, 2024.

[51] Omkar Thawakar, Muzammal Naseer, Rao Muhammad Anwer, Salman Khan, Michael Felsberg, Mubarak Shah, and Fahad Shahbaz Khan. Composed video retrieval via enriched context and discriminative embeddings. In *CVPR*, 2024.

[52] Shyamgopal Karthik, Karsten Roth, Massimiliano Mancini, and Zeynep Akata. Vision-by-language for training-free compositional image retrieval. In *ICLR*, 2024.

[53] Omkar Thawakar, Dmitry Demidov, Ritesh Thawkar, Rao Muhammad Anwer, Mubarak Shah, Fahad Shahbaz Khan, and Salman Khan. Beyond simple edits: Composed video retrieval with dense modifications. In *ICCV*, 2025.

Appendix

A The SG-Ego dataset

SG-Ego Examples. We show some examples from our dataset in Figure 3, Figure 4 and Figure 5, reporting the spatial scene graph at the start of the action (G_t), the textual narration describing the human action and the final consolidated spatio-temporal scene graph over the entire temporal window $G_{t:t+T}$.

SG-Ego-Align. The Align dataset consists of approx. 3.8M spatio-temporal graphs from 7297 unique videos. Each video is paired with its corresponding textual narration from Ego4D. We leverage the EgoClip [5] annotations to define a temporal window around each annotated action. These graphs have 10.97 ± 6.58 nodes and 16.06 ± 12.24 edges on average.

SG-Ego-Edit. The Edit dataset consists of 360k training and 7.2k validation samples. Samples are taken from 7025 and 182 unique videos for the train and validation splits, respectively. In the **train split**, the start graphs have 5.44 ± 3.42 nodes and 4.35 ± 3.33 edges on average. The spatio-temporal graphs consolidated over the entire window have 11.68 ± 6.50 nodes and 17.28 ± 12.05 edges on average. In the **validation split**, the start graphs have 5.10 ± 3.35 nodes and 4.05 ± 3.24 edges on average. The spatio-temporal graphs consolidated over the entire window have 12.33 ± 7.19 nodes and 20.36 ± 15.49 edges on average.

Object and relation vocabularies. We define a closed set vocabulary of node and edge labels. Specifically, we mine node labels from the Ego4D nouns taxonomy and remove synonyms by clustering the corresponding text embeddings. Similarly, relations are taken from the Ego4D verbs taxonomy and from the relations labels of PVSG [10]. Following this process, we obtain a set of $N_{obj} = 1480$ objects and $N_{rel} = 387$ relations. We report node and edges distributions in Figure 6. Note that we label the node relative to the camera wearer, *i.e.*, the human recording the video and performing the main actions, as the "main actor", in order to differentiate it with other possible people in the scene, labeled as "person".

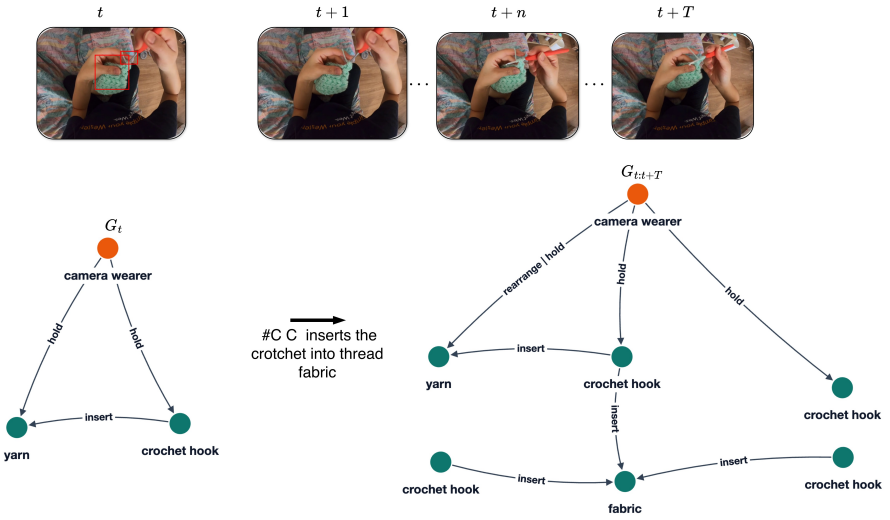


Figure 3: **Qualitative examples from the SG-Ego dataset**. Video UID: `a1b43a0c-cfaa-44df-adcc-531697ce1f8c`, from second 70.4 to 71.6. Notice that in $G_{t:t+T}$, multiple crochet hook nodes appear because the graph tracks the main actor while knitting, and the crochet hook bounding box changes significantly over time as the action progresses.

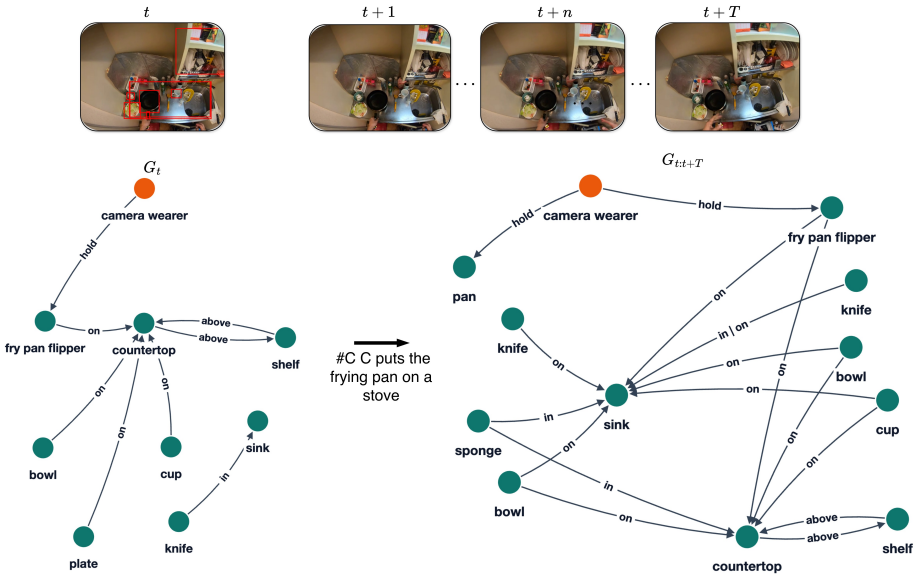


Figure 4: **Qualitative examples from the SG-Ego dataset**. Video UID: `727ffce8-20ec-4111-af26-698eb306e8c7`, from second 74.6 to 75. Interestingly, by aggregating information across the temporal action window, $G_{t:t+T}$ captures a richer representation of the scene than the frame-level graph G_t. For example, the two *knife* nodes correspond to distinct knives both present in the environment but not captured in a single frame-level observation.

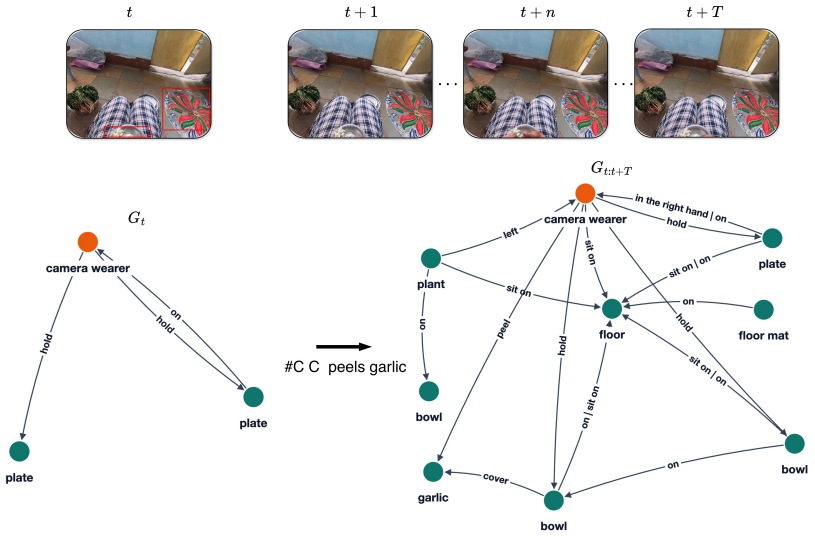


Figure 5: **Qualitative examples from the SG-Ego dataset**. Video UID: `31d6fe77-da70-42da-8f47-66bb79b9285b`, from second 89.8 to 90.8. Notice that the temporal aggregation in $G_{t:t+T}$ enriches the scene representation by incorporating objects and relations observed at different moments of the action sequence. While the initial graph G_t only captures a limited subset of the environment, the consolidated spatio-temporal graph reveals additional objects (*e.g.*, bowls, garlic and plant) together with evolving interactions involving the main actor.

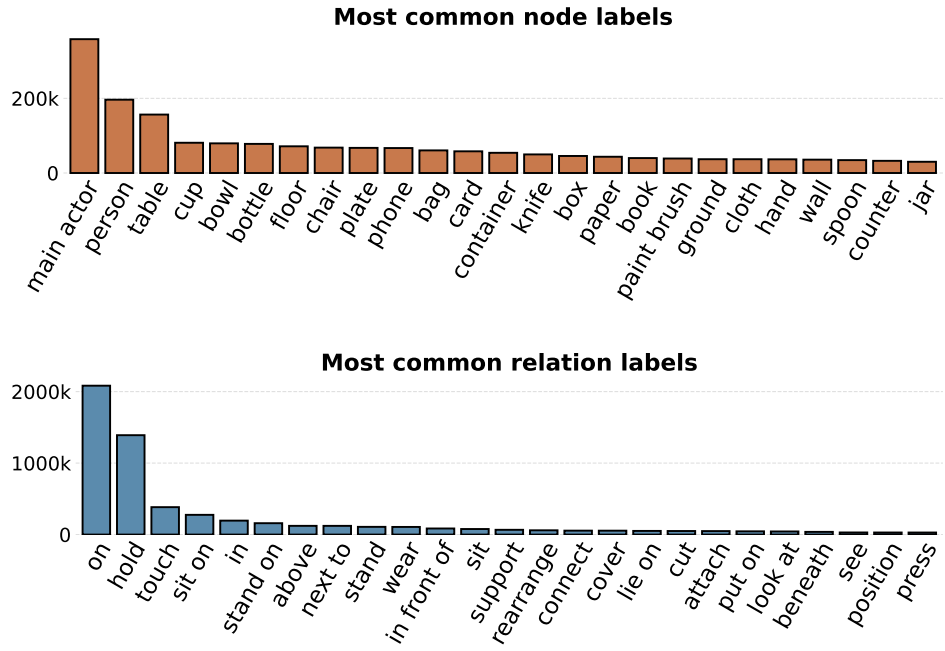


Figure 6: **Node and edge label distribution in SG-Ego-Edit.** We define a vocabulary of object and relation labels by mining the Ego4D taxonomy and the PVSG relation labels. Following this process, we obtain a set of $N_{obj} = 1480$ objects and $N_{rel} = 387$ relations.

A.1 Annotation Pipeline

For frame-level triplets extraction, we use Qwen3.5-9B [38]. We prompt the model with one frame at the time, disable thinking and set the generation temperature to $t = 0.7$. We report the full prompt used to extract the frame-level triplets in the box "Triplets extraction prompt".

Triplets extraction prompt

You are an expert in human-object and object-object interaction analysis.
You are given an image corresponding to a frame coming from an egocentric video and your final goal is to generate a good description of the interactions in the image.

Output the relations as a list of triplets "subject, predicate, object", one per line.

Indications YOU MUST FOLLOW:
- Strictly follow the output format required and only include human-object or object-object interactions. For example, do not output "(wall, white)" because it's not an object-object interaction.
- Be exhaustive about all the spatial and functional interactions in the scene.
- Be specific in the relation types and avoid vague predicates like 'is', 'has' or 'near'.
- Refer to the human wearing the egocentric camera as "camera_wearer", and use "human_x" for any other people in the scene.
- DO NOT repeat the same triplet more than once.
- DO NOT include attributes about the objects, such as their color, size, or other visual details. Focus only on the interactions between them.
- Include spatial relations between human and objects and between objects (such as "cup_1, on, table_1").
- Include functional relations only between the human and the objects in the scene (such as "human_1, holding, cup_2").
- DO NOT include functional relations between objects in the scene (such as "cup_1, used for,

drinking"). Use only spatial relations for object-object interactions.
- Keep track of multiple instances of the same object category in the scene using numerical indices (e.g., "cup_1", "cup_2") and use these indices consistently across all triplets to refer to the same instance.
- For objects that are not involved in any interaction, do not describe them.
- DO NOT describe what the human is wearing.

B GLEN Ablations and extended results

B.1 Graph-Text Alignment

Table 5: **Ablations on EgoMCQ.** We compare different configurations of GLEN using the Graph-Text Contrastive Alignment (**GTCA**) and Graph-Text Matching (**GTM**) objectives. For prediction, we can use either the GTCA head (trained with the contrastive objective), the GTM head (trained for graph-text matching) or an ensemble of the two heads. We indicate with n the number of negative samples for each anchor in the batch.

Configuration	EgoMCQ Inter	EgoMCQ Intra
GTCA ($n = 1$)	89.3	53.2
GTCA ($n = 3$)	90.6	53.9
GTCA ($n = 1$) + GTM	89.6	53.3
GTCA ($n = 3$) + GTM	**91.0**	54.4
GTCA ($n = 1$) + GTM†	89.4	53.7
GTCA ($n = 3$) + GTM†	**91.0**	**54.9**

Mode	EgoMCQ Inter	EgoMCQ Intra
GTCA head	91.0	54.9
GTM† head	80.8	54.0
GTCA + GTM† heads	**91.2**	**56.2**

We report in Table 5 (left) the impact of different training objectives for the Graph-Text Alignment task on EgoMCQ. Here, n denotes the number of negatives sampled for each anchor within the batch. We observe that hard negative mining significantly improves graph-text alignment performance. Since scene graphs provide a high-level abstraction of the evolving scene, this strategy encourages the model to distinguish between actions occurring in visually similar scenes while capturing different underlying actions. The GTM objective further helps the model better distinguish between similar scene graphs. The GTM† variant further restricts sampling to the same video.

For the evaluation in Table 5 (right), we use either the cross-modal similarity between the graph and text embeddings (GTCA head), the output of the GTM head or a combination of the two.

B.2 Activity-conditioned graph edit forecasting

We show in Table 6 the impact of the number of node queries on the A-GEF task. Increasing the number of queries allows the model to specialize each query on different objects which is particularly helpful on the large consolidated graphs from SG-Ego-Edit.

B.3 Long-Horizon Reasoning on EXPLORE-Bench

We present the full results for Explore-Bench in Table 7.

Table 6: **Activity-conditioned graph edit forecasting (A-GEF) on SG-Ego**.

Num queries	Triplet Recall		
	R@20	R@50	R@100
G_t (*static*)	25.94	25.94	25.94
64	36.84	44.06	46.61
72	36.93	44.63	47.55
100	36.73	45.30	49.38
128	36.69	45.39	49.91

Table 7: **Evaluation results on EXPLORE-Bench.** `Short`, `Medium` and `Long` mean the subsets with short (11-99), medium (100-199) and long (200-694) atomic-action sequences, respectively. † Reproduced results using the official protocol. # and ∗ represent the non-thinking and thinking mode, respectively.

Methods	`Short`				`Medium`				`Long`				`Full`			
	S_{obj}	S_{att}	S_{rel}	S_{uni}	S_{obj}	S_{att}	S_{rel}	S_{uni}	S_{obj}	S_{att}	S_{rel}	S_{uni}	S_{obj}	S_{att}	S_{rel}	S_{uni}
Proprietary Multimodal Foundation Models (API)																
GPT-5.2-Chat	59.91	1.74	2.70	48.71	59.88	1.67	2.65	47.85	58.06	1.65	2.61	46.91	59.69	1.70	2.67	48.19
Gemini-3-Flash	60.31	1.84	2.78	50.27	59.33	1.76	2.71	48.84	58.27	1.72	2.69	48.18	59.72	1.80	2.75	49.51
Gemini-3-Pro	61.29	1.81	2.77	50.11	60.99	1.75	2.74	49.44	59.17	1.70	2.70	48.31	60.94	1.77	2.75	49.66
Open-source Non-thinking Multimodal Foundation Models																
Qwen2.5-VL-3B-Instruct	48.17	1.36	2.27	39.70	44.19	1.22	2.17	36.92	39.92	1.09	2.08	34.28	45.78	1.28	2.21	38.07
Qwen2-VL-7B-Instruct	47.64	1.34	2.23	39.14	46.04	1.27	2.16	37.73	43.79	1.22	2.15	36.73	46.62	1.30	2.20	38.35
Keye-VL-1.5-8B#	49.70	1.34	2.24	39.78	49.49	1.34	2.24	39.69	46.15	1.25	2.17	37.68	49.23	1.33	2.23	39.51
LLaVA-OneVision-1.5-4B	51.15	1.42	2.33	41.37	49.17	1.33	2.28	39.85	46.35	1.28	2.26	38.64	49.88	1.37	2.31	40.50
Qwen3-VL-2B-Instruct	52.18	1.48	2.55	43.77	49.03	1.37	2.45	41.38	43.27	1.18	2.28	37.34	50.02	1.40	2.48	42.17
Qwen2.5-VL-7B-Instruct	52.80	1.52	2.46	43.58	51.25	1.46	2.41	42.32	47.78	1.34	2.36	40.22	51.67	1.48	2.43	42.74
LLaVA-OneVision-1.5-8B	53.25	1.54	2.51	44.13	51.21	1.47	2.44	42.63	47.62	1.38	2.41	40.83	51.87	1.49	2.47	43.21
Ovis2.5-2B#	55.65	1.60	2.56	45.64	51.70	1.44	2.43	42.49	47.96	1.33	2.35	40.13	53.33	1.51	2.49	43.86
InternVL3.5-2B	57.51	1.64	2.62	46.80	55.35	1.53	2.50	44.60	52.40	1.42	2.41	42.32	56.14	1.58	2.55	45.48
InternVL3.5-8B	57.85	1.65	2.62	47.00	55.77	1.56	2.55	45.30	53.30	1.51	2.51	43.99	56.57	1.60	2.58	46.04
Qwen3.5-9B-Instruct†	58.73	-	2.74	-	56.20	-	2.63	-	51.21	-	2.55	-	56.95	-	2.68	-
MiMo-VL-7B-RL-2508#	56.13	1.67	2.60	46.53	55.82	1.64	2.59	46.14	53.42	1.58	2.54	44.68	55.71	1.65	2.59	46.18
MiniCPM-V-4.5 (8B)#	58.86	1.70	2.63	47.67	57.45	1.64	2.58	46.49	54.69	1.56	2.55	44.95	57.87	1.66	2.60	46.93
Ovis2.5-9B#	59.51	1.74	2.72	48.85	57.84	1.65	2.63	47.01	53.83	1.55	2.50	44.32	58.26	1.68	2.66	47.66
Qwen3-VL-8B-Instruct	61.34	1.88	2.84	51.23	60.78	1.85	2.81	50.64	56.83	1.73	2.71	48.00	60.63	1.85	2.82	50.65
Open-source Thinking Multimodal Foundation Models																
Ovis2.5-2B*	51.10	1.40	2.43	42.02	44.52	1.17	2.20	36.89	40.70	1.04	2.10	34.25	47.50	1.28	2.31	39.25
Qwen3-VL-2B-Thinking	59.23	1.47	2.63	46.13	57.02	1.19	2.53	42.79	48.82	1.02	2.30	37.81	57.26	1.32	2.56	43.97
Ovis2.5-9B*	55.92	1.61	2.59	45.95	53.79	1.51	2.50	44.03	49.71	1.39	2.41	41.48	54.44	1.55	2.54	44.74
Keye-VL-1.5-8B*	56.58	1.58	2.55	45.64	56.54	1.54	2.53	45.14	53.13	1.62	2.43	44.09	56.18	1.57	2.53	45.28
MiMo-VL-7B-RL-2508*	57.32	1.64	2.56	46.26	56.85	1.59	2.53	45.63	56.58	1.57	2.50	45.17	57.06	1.61	2.54	45.90
MiniCPM-V-4.5 (8B)*	58.87	1.74	2.67	48.25	57.79	1.69	2.63	47.33	54.16	1.54	2.52	44.46	57.95	1.70	2.64	47.49
GLM-4.6V-Flash (9B)	60.70	1.78	2.71	49.32	58.44	1.70	2.65	47.67	54.95	1.60	2.60	45.75	59.22	1.73	2.67	48.31
Step3-VL-10B	61.32	1.76	2.78	49.87	60.06	1.62	2.71	48.01	59.11	1.60	2.64	47.09	60.61	1.69	2.74	48.87
Qwen3-VL-8B-Thinking	63.77	1.91	2.85	52.08	62.61	1.81	2.78	50.56	58.02	1.64	2.63	47.07	62.70	1.84	2.80	50.96
Embodied/Egocentric Multimodal Models																
Embodied-Reasoner (7B)	39.74	1.00	1.96	32.65	39.69	0.99	1.96	32.56	37.22	0.97	2.02	32.23	39.44	1.00	1.97	32.57
EgoThinker (7B)	48.22	1.36	2.23	39.39	45.88	1.28	2.19	37.89	42.42	1.16	2.08	35.36	46.71	1.31	2.20	38.38
Graph Edits																
GLEN	**66.12**	-	2.67	-	**66.71**	-	2.73	-	59.37	-	2.67	-	**65.59**	-	2.69	-

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction reflect the contributions of the papers in terms of method and experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A "Future directions" section is reported in the paper, in which there is a discussion of the current limitations of the proposed method and possible future developments.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: We do not have theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Details on how the proposed annotation set is created are reported in the paper, as well as implementation details of the proposed model. More information are also reported in the Appendix. We will also release the code upon acceptance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We are not currently submitting the code and the dataset according to the NeurIPS guidelines. Nonetheless, we plan to publicly release the dataset annotation and model upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We reported implementation details on the training and evaluation processes in the experimental section of the paper.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Error bars are not reported for computationally complexity reasons.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include information related to the computational resources in the implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics and we confirm we have conducted our research in accordance to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: These safeguards do not apply to our research.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available datasets and models with proper citation.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We plan to release the code and the dataset annotations upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.