ESCA: Contextualizing Embodied Agents via Scene-Graph Generation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Multi-modal large language models (MLLMs) are making rapid progress toward general-purpose embodied agents. However, current training pipelines primarily rely on high-level vision-sound-text pairs and lack fine-grained, structured alignment between pixel-level visual content and textual semantics. To overcome this challenge, we propose ESCA, a new framework for contextualizing embodied agents through structured spatial-temporal understanding. At its core is SGClip, a novel CLIP-based, open-domain, and promptable model for generating scene graphs. SGClip is trained on 87K+ open-domain videos via a neurosymbolic learning pipeline, which harnesses model-driven self-supervision from video-caption pairs and structured reasoning, thereby eliminating the need for human-labeled scene graph annotations. We demonstrate that SGClip supports both prompt-based inference and task-specific fine-tuning, excelling in scene graph generation and action localization benchmarks. ESCA with SGClip consistently improves both open-source and commercial MLLMs, achieving state-of-the-art performance across two embodied environments. Notably, it significantly reduces agent perception errors and enables open-source models to surpass proprietary baselines.

1 Introduction

Recent advances in large-scale pretraining have enabled foundation models to assist with a wide range of tasks, from language and vision understanding [2, 14, 47, 77] to mathematical problem solving [17, 90] and code generation [20, 50, 56]. However, it remains an open challenge to realize embodied agents that are capable of doing household chores, training alongside humans in physical activities, or providing care for the aging [12, 36]. A critical first step toward this goal is equipping agents with fine-grained perception to ground abstract goals in physical interactions.

Despite progress, multi-modal large language models (MLLMs) still struggle to build spatially and temporally grounded world models. Their inability to reliably ground visual features with spatial-temporal relations creates a disconnect between conceptual semantics and pixel-level observations [55, 83]. This lack of structured, fine-grained scene understanding severely limits their effectiveness in embodied environments. In fact, our empirical analysis shows that up to 69% of agent failures stem from perception errors, highlighting the need for frameworks that can bridge this gap.

To address this challenge, we propose integrating structured scene graphs into the perception, reasoning, and planning pipelines of MLLM-based embodied agents. While prior work has explored enhancing MLLMs with external visual grounding modules, these approaches typically rely on open-domain object detection models such as Grounding DINO [51] and YOLO [26]. However, these models are primarily designed for object identification and often overlook semantic attributes, inter-object relationships, and temporal consistency.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Figure 1: An overview of embodied agent pipeline augmented with ESCA. In each cycle, the agent takes in an instruction and the environmental feedback and outputs a concrete executable action, through a sequence of perception, reasoning, and planning. The action is then executed and the environment will provide the next state. Notably, ESCA contextualizes the task planner with grounded visual features represented as a scene graph.

In this work, we introduce ESCA (Embodied and Scene-Graph Contextualized Agent), a framework designed to contextualize MLLMs through open-domain scene graph generation (Figure 1). Much like the bioluminescent lure of a deep-sea angler sh, which illuminates its surroundings to reveal otherwise hidden prey, ESCA provides structured visual grounding that helps MLLMs make sense of complex and ambiguous sensory environments. A key feature of ESCA is selective grounding: rather than injecting full scene graphs, which may degrade performance, the MLLM rst identi es the subset of objects, attributes, and relations most pertinent to the instruction, then determines the essential entities for task completion. This mechanism is supported by our transfer protocol, which performs probabilistic reasoning over object names, attributes, and spatial relations to construct prompts enriched with the most relevant scene elements. At its core is SGClip, a CLIP-based model that captures semantic visual features, including entity classes, physical attributes, actions, interactions, and inter-object relations.

Through experiments on four challenging embodied environments, we demonstrate that ESCA consistently improves the performance of all evaluated MLLMs, including both open-source and proprietary models. By providing structured and grounded scene graphs, ESCA significantly reduces perception errors, laying the foundation for more reliable reasoning and planning. Beyond its integration with MLLM-based agents, we show that SGClip, when evaluated independently, exhibits strong zero-shot generalization, is promptable for task-specific scene understanding, and remains ne-tunable for downstream tasks such as action recognition.

In summary, our contributions are as follows: (1) we present ESCA, a general framework for contextualizing MLLM-based embodied agents through selective scene graph generation; (2) we introduce the transfer protocol for enriching prompts with probabilistically inferred scene-specific information for diverse embodied benchmarks; (3) we introduce SGClip, a generalizable and ne-grained scene graph generation model, along with ESCA-Video-87K, an MLLM annotated dataset; (4) we conduct extensive evaluations demonstrating the effectiveness and versatility of ESCA and SGClip across both embodied agent and scene understanding tasks.

2 ESCA: A Framework for Embodied Agents

2.1 Background

Embodied Environments. Recent research on embodied agents has been accelerated by the availability of simulated environments such as VisualAgentBench [52] and EmbodiedBench [85], which provide rich, multimodal task suites covering navigation, manipulation, and interaction across diverse scenarios. These benchmarks pose challenges that require agents to operate in both a) low-level action spaces such as continuous control signals over robot joints, and b) high-level action spaces such as programmatic instructions and skills that abstract over low-level actions.

Figure 2: A detailed illustration of the visual description module, which involves concept extraction, object identification, scene graph prediction, and visual summarization. We also illustrate sample MLLM prompts used in a kitchen environment for the concept extraction and summarization steps.

These tasks are typically modeled as Partially Observable Markov Decision Processes (POMDPs), represented as 7-tuple (S, A, T, O, L_{ins}, R), where S is the unobservable state space, A is the task-specific action space, O is the visual perception space where $o_t \in O$ is an image frame at time t, $T : S \times A \to S$ is the state transition dynamics, $O : S \to O$ relates the underlying state to the observations, L_{ins} is the natural language instruction for the agent, and $R : S \to \{0, 1\}$ is the reward function indicating whether the task has been completed.

Embodied Agents and MLLM-Based Embodied Agents. A typical embodied agent interacts with the environment by maintaining a history of observations and actions $h_t = (I_0, a_0, I_1, \ldots, a_{t-1}, I_t)$, where a_t is the action taken by the agent at time t. The agent selects the next action by conditioning on the instruction L_{ins} and history h_t via a policy $\pi(a_t \mid L_{ins}, h_t)$. An MLLM-based embodied agent realizes such a policy by leveraging a multi-modal model that processes both: a) image data I_t, and b) textual data, including the instruction L_{ins} and textual representations of actions $a \in A$.

Established by [35], recent MLLM-based agent architectures often decompose the policy evaluation process into structured stages inspired by cognitive reasoning workflows (Figure 1): 1) *Visual Description* extracting and summarizing visual inputs; 2) *Memorization* integrating observations with historical context to build situational awareness; 3) *Reasoning* inferring task-relevant insights based on the combined context; 4) *Language Plan* generating high-level plans or action sequences in natural language; 5) *Executable Plan* translating plans into concrete actions executable by the agent.

Challenges of MLLM-Based Embodied Agents. Despite recent progress, MLLM-based embodied agents remain fragile in complex environments due to compounded errors across perception, reasoning, and planning [12, 36]. *Perception errors* include hallucinated objects, misrecognized entities or actions, and incorrect spatial relationships. *Reasoning errors* arise when agents fail to correctly infer spatial relations or recognize task termination states. These issues propagate into *planning*, where agents may skip critical steps or generate invalid plans due to inaccurate state estimation.

2.2 The ESCA Framework

At the core of ESCA is a simple yet effective idea: contextualizing MLLM-based agents with grounded, structured scene-graph information to improve visual description. Specifically, given a language instruction L_{ins} and interaction history h_t, the agent generates a multi-modal message sequence of images and text. While existing approaches rely on end-to-end MLLMs to implicitly perform this step, ESCA decomposes the process into four modular stages below (Figure 2).

Selective Concept Extraction. This module extracts concepts with MLLM guided by carefully designed prompts based on the instruction L_{ins} and the history h_t. Rather than producing only free-form natural descriptions, ESCA requires the MLLM to explicitly extract structured concepts that are most relevant to the query, which can be classified as follows: a) *entity classes* such as (car, knife, person, cup); b) *attributes* including physical properties (red, small, broken) and semantic states (close-by, far, moving, sitting); c) *relations* covering spatial relations (behind, above) and interactions (cutting, cooking).

Concept extraction is guided by two signals: 1) the instruction L_{ins}, which highlights target entities, attributes, or relations the agent should focus on, and 2) the visual feedback I_t, which provides spatial context such as environmental structures (e.g., barriers, pathways) and dynamic interactions (e.g.,

objects being manipulated). Overall, this MLLM-generated structured output provides a targeted concepts set $\mathcal{C} = \{c_1, c_2, \ldots\}$ for subsequent object identification and scene graph generation.

Object Identification. Given the extracted concepts and the agent's visual feedback, the second module grounds these concepts to specific image segments $\mathcal{S} = \{s_1, s_2, \ldots\}$ each represented by a bit-mask. This step isolates visual elements in the frame that correspond to classified entities or attributes, enabling downstream semantic inference over localized visual units rather than raw pixels.

The object identification module is implemented using a multi-stage pipeline. First, Grounding DINO (GD) [51], a vision-language detection model, takes the extracted concepts and the visual input to predict bounding boxes for the mentioned entities. These bounding boxes are further refined into precise segmentation masks using SAM2 [72], a segmentation model that produces pixel-accurate regions. Leveraging the GD + SAM2 pipeline improves computational efficiency and offers better generalization to both attribute and relational concepts, as further detailed in the Appendix.

Scene Graph Prediction. We then construct a scene graph (SG) [31, 35, 81] by grounding the extracted concepts into structured visual elements. Specifically, the SG contains two types of probabilistic facts: 1) unary facts of the form $p :: c_i(s_j)$, each describing an entity s_j with their classes, attributes, or states represented by c_i; 2) relational facts of the form $p :: c_i(s_j, s_k)$, each representing that a relation or an interaction c_i between a pair of grounded entities s_j and s_k. Notably, each predicted fact is associated with a confidence score denoted by p. This probabilistic formulation allows the SG to capture uncertainty and distributions over possible scene interpretations.

Implementation-wise, we build SGClip, a CLIP-based model [66] fine-tuned for open-domain scene graph prediction. The design of SGClip is guided by three aforementioned key desiderata. First, it supports open-domain concept coverage, enabling it to generalize beyond fixed taxonomies. Second, it is adaptable to extracting diverse types of information, including entity classes, attributes, and inter-object relations. Third, it produces probabilistic predictions, allowing the model to capture uncertainties. We provide a more detailed discussion of SGClip and its training in Section 3.

Visual Summarization and Validation. This module distills these multi-modal signals into a list of messages to contextualize the agent's reasoner and planner. Concretely, the summarizer takes a prompt and is responsible for transforming the structured scene graph into natural descriptions, while also validating the consistency between the visual feedback and the underlying structured scene graph. While the summary is customizable via our transfer protocol, the set of messages include 1) the current view (and potentially historic ones) augmented with visualized bounding boxes served as markers, 2) image segments corresponding to key entities, 3) the textual description of scene graph and segments, and 4) an analysis of history actions and how they caused the current scene.

2.3 Transfer Protocol

To enable ESCA to generalize across different downstream tasks, we define a general transfer protocol based on the customization of two prompt templates, positioned at the entry and exit points of the entire visual description module. The goal of this unified transfer protocol is to maximize adaptability of ESCA across tasks with diverse planning strategies, action spaces, and reasoning requirements, while maintaining a consistent interface.

Specifically, the first prompt, the Concept Extraction Prompt, specifies the required JSON output format and indicates, or enumerates if possible, task-specific concepts that the agent should focus on. The second prompt, the Visual Summarization Prompt, guides the model to produce a contextualized summary that integrates both grounded image segments and task-specific textual elements, such as target objects, desired states, and environmental constraints. Together, these prompts provide a principled way to adapt ESCA to diverse embodied AI tasks without retraining the core system. We present a case study of the transfer protocol in Figure 2 and provide further details in Section 4.

3 SGClip Model

To enable the generation of spatial-temporal scene graphs in an open-domain setting, especially the embodied environments, we develop SGClip, a CLIP-based foundation model for [structured] scene understanding. SGClip is designed to operate in an open-domain fashion, recognizing a wide and extensible set of concepts. Secondly, it must adapt to different types of concepts, while be

(a) Entity classes (b) Attributes (c) Binary relations

Figure 3: Illustration of the inference modes of SGClip for three types of concepts: entity classes, attributes, and binary relations. While the model stays the same, the three inference modes perform different pre- and post-processing for a more accurate semantic estimation of probabilities.

capable of generalizing to unseen visual and textual domains. Finally, it must produce probabilistic predictions to capture uncertainty.

To meet these goals, SGClip builds on CLIP's vision-language architecture, which naturally supports joint reasoning over images and textual phrases. However, deploying CLIP directly is insuf cient, as it lacks specialization for structured scene graph prediction. To bridge this gap, we ne-tune SGClip to balance adaptability and generalizability. In this section, we describe 1) how to handle different types of concepts through inference time adaptation (Section 3.1), 2) how to overcome the lack of data by collecting a model-driven self-supervision dataset (Section 3.2), and 3) how to learn without relying on human annotations via a self-supervised neurosymbolic learning pipeline (Section 3.3).

3.1 Model Architecture and Inference Time Adaptation

At its core, SGClip (Scene Graph CLIP) is a single CLIP-based model designed to score concept relevance within an image. Formally, it operates as $\text{SGClip}(\ ; c) \in \mathbb{R}^{|c|}$, where is the input image and c is the a set of candidate concepts, producing a logit score for each concept that re ects the model's con dence in its presence. SGClip supports three distinct inference modes to handle different types of concepts: entity classes c_{class}, attributes c_{attr}, and binary relations c_{rela}. Illustrated in Figure 3, each mode requires a specialized formulation of the input concepts and scoring process, effectively allowing SGClip to operate exibly across these concept types:

Entity classes. In this setting, SGClip is used to identify the most likely entity class presented in an image segment. Let c_{class} denote the list of candidate entity classes. Since an entity is typically assumed to belong to a single class, we apply softmax normalization over the logit scores produced by SGClip for these candidates: $\text{softmax}(\text{SGClip}(\ ; c_{\text{class}}))$.

Attributes. To estimate the likelihood that a speci c segment possesses a particular attribute c, we construct a binary contrast between the attribute and its negation by evaluating $\text{softmax}(\text{SGClip}(\ ; [c; \text{not } c]))$, where not c denotes the negated textual phrase (e.g., "not red" for the attribute "red"). The rst element of the resulting probability distribution corresponds to the model's estimated likelihood. To improve computational ef ciency, we perform batched evaluation by merging all attribute-contradiction pairs into a single concept set $c_{\text{attr}} = \bigcup_{c \in c_{\text{attr}}} [c, \text{not } c]$.

Binary relations. For binary relation prediction, the goal is to determine whether a relation holds between two segments $_i$ and $_j$. To do this, we rst compute a bounding region $_{ij}$ that tightly encloses both segments. Within this region, we apply distinct color tinting to $_i$ and $_j$ to indicate their directional roles (subject and object). To provide additional relational context, especially for interactions like "cutting," we augment the relation phrase by including the predicted entity classes of the subject and object, generating prompts like "(knife, cutting, cabbage)". Relation predictions are thus conditioned on the classes of both the subject and the object. Speci cally, for each segment $_i$, we first compute its most likely class c_i by selecting the top prediction from the entity class:

$$c_i = c_{\text{class}u}, \quad \text{where } u = \operatorname{argmax}_{u \in 1:|c_{\text{class}}|} \text{SGClip}(\ _i; c_{\text{class}})_u.$$

We then form the augmented relation phrase (c_i, c, c_j). Similar to attribute prediction, we contrast the candidate relation with a special token <norel> denoting "no relation," and compute $\text{softmax}(\text{SGClip}(\ _{ij}; [f(\ _i; c; \ _j); \text{<norel>}]))$, to obtain the probability of whether the relation c holds between object i and j. In practice, this process is batched over all segment pairs and relation concepts to maximize ef ciency. $c_{\text{rela}} = \{f(\ _i; c; \ _j) \mid i, j \in 1:|\ |, j \neq i, c \in c_{\text{rela}}\} \cup \{\text{<norel>}\}$.

Figure 4: Illustration of the construction of ESCA-Video-87K dataset and the model-driven self-supervised fine-tuning pipeline of our SGClip model. In addition to videos and their natural language captions, ESCA-Video-87K includes object traces, open-domain concepts, and programmatic specifications for 87K video-caption pairs. The dataset is then used to train SGClip via LASER, a neurosymbolic learning procedure based on spatial-temporal alignment.

3.2 ESCA-Video-87K Dataset

We adopt the neurosymbolic weak-supervision pipeline introduced in LASER [38], which enables learning fine-grained STSGs from weak supervision signals derived from spatial-temporal programmatic specifications, eliminating the need for costly manual annotations. While the details of this learning pipeline are provided in Section 3.3, we begin by introducing the ESCA-Video-87K dataset, the dataset we curate and use to train SGClip.

The ESCA-Video-87K dataset is constructed on top of the publicly available LLaVA-Video-178K dataset [91], and consists of 87K short video clips, each paired with natural language captions generated by GPT-4o [32]. As illustrated in Figure 4, these captions are first processed to extract relevant concepts which are then fed into GDINO [51] and SAM2 [62] to obtain object traces, which are sequences of object segmentations that evolve across multiple video frames. In addition, these concepts are also used to assist in generating spatial-temporal programmatic specifications, expressed in a linear temporal logic-based language. To construct these specifications, we develop a semantic parsing pipeline, again leveraging GPT-4, which converts high-level captions into structured temporal statements. These specifications formally describe how the semantics of object traces evolve, capturing temporal relations using operators such as “until”, “finally”, or “always”.

In summary, each data point in ESCA-Video-87K is represented as a 5-tuple $(I, L_{cap}, C, \Gamma, \phi)$, where $I = \{I_1, I_2, \ldots\}$ is the video, L_{cap} is the associated natural language caption, $\Gamma = \{\gamma_1, \gamma_2, \ldots\}$ is the set of object traces, $C = \{c_1, c_2, \ldots\}$ is the set of extracted concepts, and ϕ is the spatial-temporal programmatic specification. This rich, multi-level annotation enables training models like SGClip without requiring manual scene graph labeling. We defer additional details about the dataset construction process and data statistics to the Appendix.

3.3 Neurosymbolic Learning Pipeline

Given the ESCA-Video-87K, our goal is to fine-tune the SGClip model using the provided object traces, concepts, and spatial-temporal programmatic specifications. This is achieved by aligning the scene graphs generated by SGClip with the expected specifications [38], where the degree of alignment serves as the learning signal (Figure 4). To perform this alignment in a differentiable manner, we leverage the Scallop programming language [46], enabling symbolic alignment checks to be integrated into end-to-end gradient-based learning.

Specifically, the alignment loss computation mirrors the inference-time adaptation procedure described in Section 3.1, where different types of concepts are processed differently but unified under the same model. The pipeline is further enhanced with semantic losses, derived from evaluating

common-sense and temporal constraint satisfaction, as well as a contrastive loss that encourages the model to distinguish between matched and unmatched scene graph–specification pairs. Additional details of the training process are provided in the Appendix.

4 Embodied Environments and Transfer Protocol Setup

We evaluate our approach on EmbodiedBench [42], a benchmark suite designed to assess MLLM-based embodied agents. We focus on two environments: EB-Navigation and EB-Manipulation, each of which requires different levels of perception, reasoning, and control, and may benefit from a contextualized visual description. To adapt ESCA to these tasks, we apply our transfer protocol by designing two specialized prompts for each environment. Full prompt templates are provided in the Appendix; here, we summarize the core challenges and how ESCA addresses them.

EB-Navigation is built on AI2-THOR [39] and focuses on visual navigation tasks where the agent must locate target objects based on language instructions, such as "navigate to the laptop." The agent relies solely on egocentric visual input and textual feedback, navigating through a space using eight low-level movement and rotation actions. Existing models often fail by generating correct high-level plans but producing incorrect low-level actions due to poor spatial grounding from an egocentric perspective. ESCA addresses this by generating accurate scene graphs that capture spatial relations and object positions. With the scene graphs, we design prompts that incorporate numerical bounding box data and temporal movement cues, helping the agent to localize targets more precisely.

EB-Manipulation extends VLMBench [13] for evaluating low-level robotic manipulation. The agent controls a 7-DoF robotic arm using discretized action spaces, with additional signals such as YOLO bounding boxes [26] and global 3D object pose estimates to assist manipulation. Existing models struggle to ground object concepts into actionable spatial representations, leading to perception failures that disrupt downstream planning and control. ESCA improves this by grounding target features into precise visual segments, enabling prompts that describe object attributes, semantic relations, and 3D spatial coordinates, giving the agent more reliable geometric context.

EB-Habitat builds upon Language Rearrangement tasks [32], simulated via Habitat 2.0 [71], and primarily evaluates high-level task decomposition and planning capabilities. The action space is limited to atomic high-level actions, such as navigate/pick/place/open/close, from which the agent is instructed to complete tasks such as "Find a toy airplane and move it to the right counter". Common failure cases of existing models include invalid actions arising from failure to identify and remember object displacements in the scene. Our model improves this by managing scene graphs of the current and desired state of the target object, which improves awareness of task progression and limits the number of focus objects.

5 Empirical Evaluation

Our experiments are designed to address two key research questions: (1) How effectively does ESCA, together with SGClip, improve embodied agent performance through structured scene graph generation? and (2) How generalizable and adaptable is SGClip when evaluated independently on open-domain, zero-shot, and downstream transfer tasks? We now detail our experimental setup and present empirical results addressing both questions.

Experimental Setup. We evaluate ESCA in EB-Navigation and EB-Manipulation, two environments that demand fine-grained perception to support low-level control. To assess the general applicability of ESCA, we integrate it with four diverse MLLMs: InternVL-2.5-38B-MPO [13], Qwen2.5-VL-72B-Ins [3], Gemini-2.0-flash [57], and GPT-4o [32]. For each MLLM experiment, we use the model for both the concept extraction and visual summarization steps (Figure 2). To further benchmark ESCA's impact, we compare to performance of MLLMs augmented with existing visual grounding modules, including Grounding DINO [51] and Ultralytics-YOLO11 [26].

For evaluating SGClip independently, we consider out-of-domain scene graph benchmarks, including OpenPVSG [84], Action Genome [33], and VidVRD [65], comparing SGClip against strong baselines such as CLIP [60], InternVL-6B [14], BIKE [79], and Text4Vis [78]. To assess SGClip's downstream adaptability beyond structured scene graph prediction, we further test the fine-tunability on the ActivityNet action recognition dataset by applying a transfer protocol.

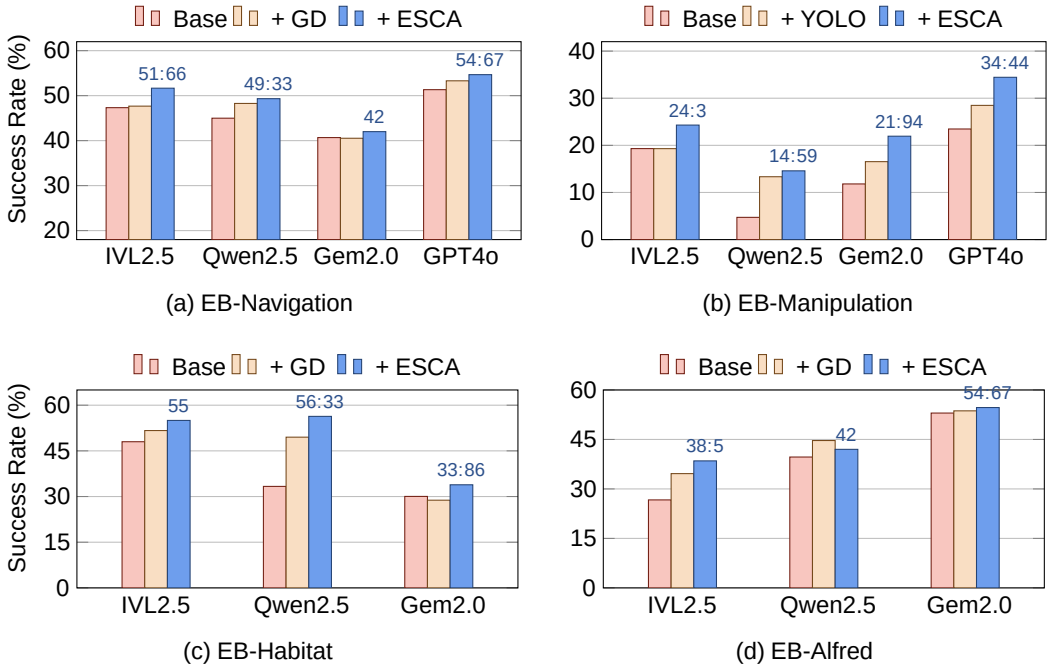


Figure 5: The overall performance on EB-Navigation and EB-Manipulation environments. We show the performance of four base models, InternVL-2.5-38B-MPO (IVL2.5), Genimi-2.0- ash (Gem2.0), Qwen2.5-VL-72B-Instruct (Qwen2.5), and GPT-4o (GPT4o), as well as their performance when accompanied with baseline visual grounding modules such as Grounding DINO (GD) or YOLO. With ESCA and SGClip, all models consistently outperform the baselines.

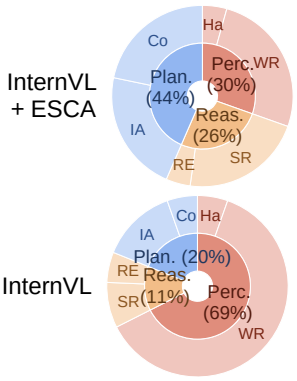
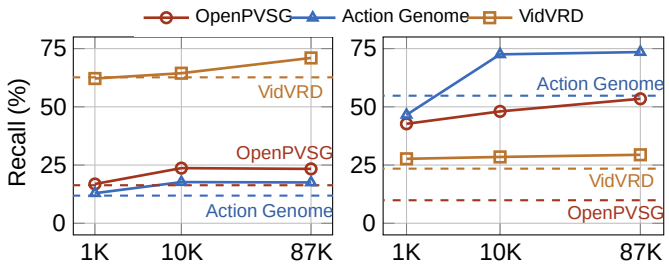


Figure 6: Two traces by InternVL with and without ESCA, on the task of "navigate to the kettle on the stove". With ESCA, the kettle can be marked in the image and textual description, encouraging the agent to decisively move towards the goal.

Figure 7: Error decomposition[1] of InternVL with or without ESCA, manually inspected on 60 EB-Navigation tasks.



(a) Entity class prediction R@1(b) Binary relation prediction R@10

Figure 9: The zero-shot performance of SGClip compared to CLIP (shown in dashed lines) on OpenPVSG, Action Genome, and VidVRD datasets. We showcase the Recall@1 metrics on entity class prediction, as well as the Recall@10 metrics on binary relation prediction. To illustrate data-ef ciency, we include the performance of checkpoints of SGClip when trained on 1K, 10K, or 87K (full) portion of ESCA-Video-87K.

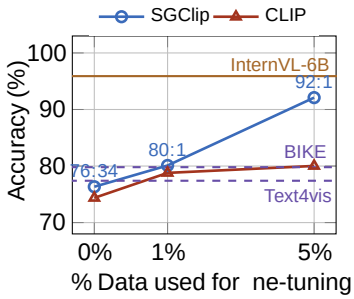


Figure 10: Down-stream ne-tunability on action recognition, evaluated on ActivityNet dataset. We also illustrate zero-shot baselines (BIKE and Text4vis) as well as a fully-supervised baseline (InternVL-6B).

ESCA for Embodied Agents. As shown in Figure 5, ESCA-augmented MLLMs consistently outperform their non-contextualized baselines across both EB-Navigation and EB-Manipulation. Remarkably, on EB-Navigation, even the open-source InternVL-2.5, when augmented with ESCA, surpasses the base performance of the proprietary GPT-4o model. While integrating Grounding DINO or YOLO improves baseline models, ESCA provides additional, substantial gains. For example, Gemini-2.0 with ESCA achieves over 10% improvement, while GPT-4o, already boosted by YOLO, still bene ts from an additional 6% performance gain on EB-Manipulation.

Through qualitative analysis of end-to-end agent behaviors, we observe that ESCA consistently improves the agent's perceptual grounding, leading to more effective task execution. As illustrated in Figure 6, an agent powered by InternVL with ESCA successfully identi es the kettle early in the episode and navigates directly toward it. In contrast, the base InternVL model fails to recognize the target and ultimately collapses onto the wall. This observation is further supported by the error decomposition analysis shown in Figure 7, where we nd that ESCA reduces the overall perception error rate from 69% to 30%. We provide additional detailed experimental results in the Appendix.

Generalizability and Adaptability of SGClip. Evaluating SGClip's zero-shot generalization, Figure 9 shows that SGClip trained on ESCA-Video-87K consistently outperforms CLIP on OpenPVSG, Action Genome, and VidVRD, demonstrating strong out-of-domain robustness. Further, SGClip shows strong adaptability, achieving notable improvements when ne-tuned on VidVRD (details provided in the Appendix). Beyond scene graph tasks, Figure 10 highlights SGClip's downstream transferability to action recognition on ActivityNet. Fine-tuned with only 1% of the training data, SGClip outperforms state-of-the-art zero-shot video recognition baselines. With 5% of the data (approximately 800 videos), SGClip achieves 92.10% accuracy, approaching the performance of InternVideo2-6B with end-to-end netuning on the ActivityNet dataset.

6 Related Works

Scene Graph in planning. Scene graphs [35, 81] are symbolic representations that encode the semantic structure of an image or video by identifying objects as nodes and their relationships as edges [40, 53]. They play a central role in a variety of vision-related tasks, including visual question answering [27, 41, 59], image captioning [86, 94], and image generation [25, 43]. More recently, scene graphs have been increasingly adopted in the domain of robotic planning, for enhanced robustness [29, 75], or veri able planning [15, 34, 61]. To enable seamless integration with multimodal large language model (MLLM) agents, ESCA constructs scene graphs from 2D image inputs and dynamically updates them through embodied interaction with the environment.

[1]The three top-level error types are Perception, Reasoning, and Planning. The second-level errors are Hallucination, Wrong Recognition, Spatial Understanding, Spatial Reasoning, Re ection Error, Inaccurate Action, and Collision. For clarity, the gure uses these acronyms to label the different error types.

Embodied Agents. Embodied agents [21, 92] are autonomous systems that perceive, reason, and interact within physical or simulated environments. To evaluate their capabilities, various benchmarks [42, 80] have been proposed, ranging from vision-language navigation [4, 16, 63] and object manipulation [18, 48] to interactive instruction following and long-horizon planning [19, 67, 68]. Recent advances in large language models (LLMs) [3, 7, 32, 73, 74] and multimodal large language models (MLLMs) [2, 14, 47, 77] are driving progress toward general-purpose embodied agents [1, 5, 8, 30, 54, 88]. Furthermore, recent MLLMs have been trained end-to-end to directly generate low-level numerical control commands [6, 9]. ESCA introduces a general framework for augmenting vision-driven, MLLM-based agents with structured scene graph information.

Neurosymbolic Methods with LLM and MLLM. A growing trend for enhancing the reasoning capabilities and robustness of large language models (LLMs) is to incorporate structured representations and leverage symbolic algorithms to reason over them [22, 37, 44, 45, 87]. These efforts span diverse domains, including code generation [20, 50, 56], mathematical problem solving [17, 90], and verifiable planning [11, 49, 70, 89]. Recent work has extended this paradigm to the low level control domain, training MLLMs with structured scene graphs in an end-to-end manner [6, 58, 66]. ESCA follows this neurosymbolic direction by introducing SGClip, which is trained in MLLM-augmented self-supervised manner enabled by neuro-symbolic methodology, using structured scene graphs as an intermediate representation to guide learning.

7 Conclusion and Limitations

We introduced ESCA, a framework for contextualizing embodied agents through scene graph generation, powered by SGClip, a promptable, open-domain scene graph model. Through a general transfer protocol, ESCA adapts to diverse tasks and consistently improves agent performance across multiple environments and MLLMs. Beyond embodied tasks, SGClip demonstrates strong generalization and adaptability on open-domain scene graph and action recognition benchmarks.

Limitations. Despite its strong performance, our framework has several limitations. First, the use of large language models for high-level planning introduces latency, making it unsuitable for real-time low-level control. Second, the system relies on 2D visual inputs and lacks support for 3D representations like point clouds, limiting depth-aware reasoning and spatial precision. Finally, while ESCA leverages MLLMs to generate coherent plans, it lacks formal mechanisms for verifying intermediate and final states during execution.

References

[1] Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. Advances in Neural Information Processing Systems, 36:22304–22325, 2023.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. arXiv preprint arXiv:2308.12966, 1(2):3, 2023.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[4] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. arXiv preprint arXiv:2006.13171, 2020.

[5] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. arXiv preprint arXiv:2403.01823, 2024.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv. org/abs/2410.24164, 2024.

[7] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258, 2021.

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818, 2023.

[9] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817, 2022.

[10] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the ieee conference on computer vision and pattern recognition, pages 961–970, 2015.

[11] Alessio Capitanelli and Fulvio Mastrogiovanni. A framework for neurosymbolic robot action planning using large language models. Frontiers in Neurorobotics, 18:1342786, 2024.

[12] Yanan Chen, Ali Pesaranghader, Tanmana Sadhu, and Dong Hoon Yi. Can we rely on llm agents to draft long-horizon plans? let's take travelplanner as an example, 2024.

[13] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025.

[14] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 24185–24198, 2024.

[15] 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, 2024.

[16] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 3164–3174, 2020.

[17] Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Jimenez Rezende, Yoshua Bengio, Michael C Mozer, and Sanjeev Arora. Metacognitive capabilities of llms: An exploration in mathematical problem solving. Advances in Neural Information Processing Systems, 37:19783–19812, 2024.

[18] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4497–4506, 2021.

[19] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. Advances in Neural Information Processing Systems, 35:18343–18362, 2022.

[20] James Finnie-Ansley, Paul Denny, Brett A Becker, Andrew Luxton-Reilly, and James Prather. The robots are coming: Exploring the implications of openai codex on introductory programming. In Proceedings of the 24th Australasian computing education conference, pages 10–19, 2022.

[21] Stan Franklin. Autonomous agents as embodied ai. Cybernetics & Systems, 28(6):499–520, 1997.

[22] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In International Conference on Machine Learning, pages 10764–10799. PMLR, 2023.

[23] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In Proceedings of the IEEE international conference on computer vision, pages 5842–5850, 2017.

[24] 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 Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 18995–19012, 2022.

[25] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16, pages 210–227. Springer, 2020.

[26] Priyanto Hidayatullah, Nurjannah Syakrani, Muhammad Rizqi Sholahuddin, Trisna Gelar, and Refdinal Tubagus. Yolov8 to yolo11: A comprehensive architecture in-depth comparative review, 2025.

[27] Marcel Hildebrandt, Hang Li, Rajat Koner, Volker Tresp, and Stephan Günnemann. Scene graph reasoning for visual question answering. arXiv preprint arXiv:2007.01072, 2020.

[28] Matthew Honnibal, Ines Montani, So e Van Landeghem, Adriane Boyd, et al. spacy: Industrial-strength natural language processing in python. 2020.

[29] Joy Hsu, Jiayuan Mao, Josh Tenenbaum, and Jiajun Wu. What's left? concept grounding with logic-enhanced foundation models. Advances in Neural Information Processing Systems, 36:38798–38814, 2023.

[30] Cassie Huang and Li Zhang. On the limit of language models as planning formalizers, 2025.

[31] Jiani Huang, Ziyang Li, Mayur Naik, and Ser-Nam Lim. LASER: A neuro-symbolic framework for learning spatio-temporal scene graphs with weak supervision. In The Thirteenth International Conference on Learning Representations, 2025.

[32] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. arXiv preprint arXiv:2410.21276, 2024.

[33] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10236–10247, 2020.

[34] Ziyuan Jiao, Yida Niu, Zeyu Zhang, Song-Chun Zhu, Yixin Zhu, and Hangxin Liu. Sequential manipulation planning on scene graph. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 8203–8210. IEEE, 2022.

[35] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3668–3678, 2015.

[36] Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. Llms can't plan, but can help planning in llm-modulo frameworks, 2024.

[37] Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. Beliefbank: Adding memory to a pre-trained language model for a systematic notion of belief. arXiv preprint arXiv:2109.14723, 2021.

[38] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.

[39] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: an interactive 3d environment for visual AI. CoRR, abs/1712.05474, 2017.

[40] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision, 123:32–73, 2017.

[41] Soohyeong Lee, Ju-Whan Kim, Youngmin Oh, and Joo Hyuk Jeon. Visual question answering over scene graph. In 2019 First International Conference on Graph Computing (GC), pages 45–50. IEEE, 2019.

[42] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. Advances in Neural Information Processing Systems, 37:100428–100534, 2024.

[43] Yikang Li, Tao Ma, Yeqi Bai, Nan Duan, Sining Wei, and Xiaogang Wang. Pastegan: A semi-parametric method to generate image from scene graph. Advances in Neural Information Processing Systems, 32, 2019.

[44] Ziyang Li, Saikat Dutta, and Mayur Naik. Iris: Llm-assisted static analysis for detecting security vulnerabilities. In The Thirteenth International Conference on Learning Representations, 2025.

[45] Ziyang Li, Jiani Huang, Jason Liu, Felix Zhu, Eric Zhao, William Dodds, Neelay Velingker, Rajeev Alur, and Mayur Naik. Relational programming with foundational models. In Proceedings of the AAAI Conference on Arti cial Intelligence, volume 38, pages 10635–10644, 2024.

[46] Ziyang Li, Jiani Huang, and Mayur Naik. Scallop: A language for neurosymbolic programming, 2023.

[47] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. arXiv preprint arXiv:2311.10122, 2023.

[48] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In Conference on Robot Learning, pages 432–448. PMLR, 2021.

[49] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+p: Empowering large language models with optimal planning pro ciency, 2023.

[50] Fang Liu, Yang Liu, Lin Shi, Houkun Huang, Ruifeng Wang, Zhen Yang, Li Zhang, Zhongqi Li, and Yuchi Ma. Exploring and evaluating hallucinations in llm-powered code generation. arXiv preprint arXiv:2404.00971, 2024.

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

[52] Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Yifan Xu, Xixuan Song, Shudan Zhang, Hanyu Lai, Xinyi Liu, Hanlin Zhao, Jiadai Sun, Xinyue Yang, Yu Yang, Zehan Qi, Shuntian Yao, Xueqiao Sun, Siyi Cheng, Qinkai Zheng, Hao Yu, Hanchen Zhang, Wenyi Hong, Ming Ding, Lihang Pan, Xiaotao Gu, Aohan Zeng, Zhengxiao Du, Chan Hee Song, Yu Su, Yuxiao Dong, and Jie Tang. Visualagentbench: Towards large multimodal models as visual foundation agents, 2024.

[53] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pages 852–869. Springer, 2016.

[54] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. ICLR, 2024.

[55] Damiano Marsili, Rohun Agrawal, Yisong Yue, and Georgia Gkioxari. Visual agentic ai for spatial reasoning with a dynamic api, 2025.

[56] Theo X Olausson, Alex Gu, Benjamin Lipkin, Cedegao E Zhang, Armando Solar-Lezama, Joshua B Tenenbaum, and Roger Levy. Linc: A neurosymbolic approach for logical reasoning by combining language models with rst-order logic provers. arXiv preprint arXiv:2310.15164, 2023.

[57] Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. Introducing gemini 2.0: Our new ai model for the agentic era, Dec 2024.

[58] Jianing Qian, Yunshuang Li, Bernadette Bucher, and Dinesh Jayaraman. Task-oriented hierarchical object decomposition for visuomotor control. arXiv preprint arXiv:2411.01284, 2024.

[59] Tianwen Qian, Jingjing Chen, Shaoxiang Chen, Bo Wu, and Yu-Gang Jiang. Scene graph re nement network for visual question answering. IEEE Transactions on Multimedia, 25:3950–3961, 2022.

[60] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. CoRR, abs/2103.00020, 2021.

[61] 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.

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

[63] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In Proceedings of the IEEE/CVF international conference on computer vision, pages 9339–9347, 2019.

[64] Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xun Yang, and Tat-Seng Chua. Annotating objects and relations in user-generated videos. In Proceedings of the 2019 on International Conference on Multimedia Retrieval, pages 279–287. ACM, 2019.

[65] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. Video visual relation detection. In Proceedings of the 25th ACM International Conference on Multimedia, pages 1300–1308, 2017.

[66] Junyao Shi, Jianing Qian, Yecheng Jason Ma, and Dinesh Jayaraman. Composing pre-trained object-centric representations for robotics from" what" and" where" foundation models. 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 15424–15432. IEEE, 2024.

[67] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10740–10749, 2020.

[68] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. arXiv preprint arXiv:2010.03768, 2020.

[69] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pages 510–526. Springer, 2016.

[70] Tom Silver, Varun Hariprasad, Reece S Shuttleworth, Nishanth Kumar, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Pddl planning with pretrained large language models. In NeurIPS 2022 foundation models for decision making workshop, 2022.

[71] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. Advances in Neural Information Processing Systems, 34:251–266, 2021.

[72] Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazoure, Rin Metcalf, Walter Talbott, Natalie Mackraz, R Devon Hjelm, and Alexander T Toshev. Large language models as generalizable policies for embodied tasks. In ICLR, 2024.

[73] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.

[74] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and ef cient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

[75] Renhao Wang, Jiayuan Mao, Joy Hsu, Hang Zhao, Jiajun Wu, and Yang Gao. Programmatically grounded, compositionally generalizable robotic manipulation. arXiv preprint arXiv:2304.13826, 2023.

[76] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. arXiv preprint arXiv:2307.06942, 2023.

[77] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. arXiv preprint arXiv:2303.04671, 2023.

[78] Wenhao Wu, Zhun Sun, and Wanli Ouyang. Revisiting classi er: Transferring vision-language models for video recognition, 2023.

[79] Wenhao Wu, Xiaohan Wang, Haipeng Luo, Jingdong Wang, Yi Yang, and Wanli Ouyang. Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6620–6630, 2023.

[80] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 9068–9079, 2018.

[81] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5410–5419, 2017.

[82] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5036–5045, 2022.

[83] 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, 2024.

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

[85] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. arXiv preprint arXiv:2502.09560, 2025.

[86] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10685–10694, 2019.

[87] Hanlin Zhang, Jiani Huang, Ziyang Li, Mayur Naik, and Eric Xing. Improved logical reasoning of language models via differentiable symbolic programming. arXiv preprint arXiv:2305.03742, 2023.

[88] Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. arXiv preprint arXiv:2307.02485, 2023.

[89] Li Zhang, Peter Jansen, Tianyi Zhang, Peter Clark, Chris Callison-Burch, and Niket Tandon. Pddlego: Iterative planning in textual environments. arXiv preprint arXiv:2405.19793, 2024.

[90] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In European Conference on Computer Vision, pages 169–186. Springer, 2024.

[91] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024.

[92] Zhonghan Zhao, Wenhao Chai, Xuan Wang, Boyi Li, Shengyu Hao, Shidong Cao, Tian Ye, and Gaoang Wang. See and think: Embodied agent in virtual environment. In European Conference on Computer Vision, pages 187–204. Springer, 2024.

[93] Kaizhi Zheng, Xiaotong Chen, Odest Chadwicke Jenkins, and Xin Eric Wang. Vlmbench: A compositional benchmark for vision-and-language manipulation, 2022.

[94] Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. Comprehensive image captioning via scene graph decomposition. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, pages 211–229. Springer, 2020.

[95] Luowei Zhou, Chenliang Xu, and Jason J Corso. Procnets: Learning to segment procedures in untrimmed and unconstrained videos. arXiv preprint arXiv:1703.09788, 2(6):7, 2017.

[96] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Wancai Zhang, Zhifeng Li, Wei Liu, and Li Yuan. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment, 2024.

8 Checklist

1. Claims

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

   Answer: [Yes]

   Justi cation: We have made four claims in the introduction. All of them are supported by our pipeline and experimental results.

   Guidelines:

   - The answer NA 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 NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and re ect how much the results can be expected to generalize to other settings.
   - It is ne 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] .

   Justi cation: We have an explicit section dedicated to conclusion, limitation, and future works.

   Guidelines:

   - The answer NA 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-speci cation, asymptotic approximations only holding locally). The authors should re ect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should re ect 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 re ect on the factors that in uence 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 ef ciency 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 speci cally 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: [NA] .

Justi cation: We do not include theoretical results.

Guidelines:

- The answer NA 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] .

   Justi cation: We include the experimental details in the appendix, and our full code base in supplementary material.

   Guidelines:

   - The answer NA 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 veri able.
   - 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 suf ce, or if the contribution is a speci c 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 suf cient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes] .

Justi cation: In the supplementary material, we provide our codebase for (1) preprocessing the ESCA-Video-87K dataset and (2) implementing ESCA pipeline and transfer protocols. We will release the dataset and open-source the code upon paper acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes] .

Justi cation: We include these speci cations in the appendix.

Guidelines:

- The answer NA 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 signi cance

Question: Does the paper report error bars suitably and correctly de ned or other appropriate information about the statistical signi cance of the experiments?

Answer: [No] .

Justi cation: We evaluate large language models with the temperature set to 0 to reduce output variance. Moreover, querying these models multiple times with the same prompt is neither environmentally sustainable nor economically feasible.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, con dence intervals, or statistical signi cance 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 veri ed.
- For asymmetric distributions, the authors should be careful not to show in tables or gures 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 gures or tables in the text.

8. Experiments compute resources

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

   Answer: [Yes] .

   Justi cation: We include the experimental details in the appendix.

   Guidelines:

   - The answer NA 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 Ethicshttps://neurips.cc/public/EthicsGuidelines ?

   Answer: [Yes] .

   Justi cation: No human objectives are involved in our research. Our dataset is created on the top of a fully open source dataset with Apache License 2.0.

   Guidelines:

   - The answer NA 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: [NA] .

    Justi cation: This work focuses on evaluating embodied agent performance in simulated environments, with accuracy measured against prede ned, human-authored criteria. As such, the study does not directly engage with real-world deployment or societal impact, making a discussion of positive or negative societal effects not applicable in this context.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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 pro les, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact speci c 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 ef ciency 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA] .

    Justi cation: Our paper poses no such risks.

    Guidelines:

    - The answer NA 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 lters.
    - 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] .

    Justi cation: The LLava-Video-178K dataset is under Apache 2.0 license.

    Guidelines:

    - The answer NA 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: [Yes] .

    Justi cation: we provide the whole repository that generates our dataset.

    Guidelines:

    - The answer NA 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 le.

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: [NA] .

    Justi cation: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is ne, 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: [NA] .

    Justi cation: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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 signi cantly 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, scienti c rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes] .

    Justi cation: ESCA is a new method augmenting MLLMs with scene graphs.

    Guidelines:

    - The answer NA 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 (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Algorithm 1: Video Mask Propagation with New Object Discovery

```
Input: Video V = {I_0, ..., I_T}, mask generator G, propagation model P, state S
Input: Batch size B, thresholds: τ_iou, τ_score, τ_inner, strides
Output: Mask dictionary video_segments : Dict[Frame][ObjectID] = Mask
video_segments ← {}, now_frame ← 0;
while not saturated do
    I ← V[now_frame]; // Load current image frame
    M ← G(I); // Generate masks for current frame
    // Find new object masks not yet propagated from this frame
    M_new ← filter(M, video_segements[now_frame], τ_iou, τ_score, τ_inner);
    // Register the new masks as prompts
    P_buf.update(M_new);
    // Prompt the mask generator with updated masks as initialize condition
    P.reset_state(S);
    foreach (oid, frame_id, mask) in P_buf do
        P.add_new_mask_prompt(S, frame_id, o, m);
    end
    // Propagate the mask prompts through the whole video
    foreach (f, {o_i}, {ℓ_i}) ∈ P.propagate(S) do
        if f ∉ video_segments then
            video_segments[f] ← {};
        end
        foreach i do
            m_i ← binarize(ℓ_i);
            video_segments[I_f][o_i] ← m_i;
        end
    end
    // Find the next frame to process with least mask coverage
    Compute coverage_f over f ∈ {now_frame + 1, ..., T};
    saturated, now_frame ← check_saturation();
end
return video_segments
```

A ESCA-Video-87K

A.1 Video and Caption Source

We build upon the LLaVA-Video-178K dataset [91], which comprises diverse video sources and GPT-generated, detailed video descriptions.

ESCA-Video-87K contains a total of 87,045 datapoints, each consisting of a video clip ranging from 0 to 30 seconds in length, along with its corresponding caption. The videos are drawn from ten primary sources, spanning a wide range of content—including egocentric and household activities, as well as crowd-sourced videos from YouTube. Specifically, these ten sources are HD-VILA-100M [82], InternVid-10M [76], VidOR [64], VIDAL (YouTube Shorts) [96], YouCook2 [95], Charades [69], ActivityNet [10], Kinetics-700 [38], Something-Something v2 [23], and Ego4d [24]. The captions were generated using GPT-4 with a dense frame sampling technique [91].

The two main contributions of ESCA-Video-87K are the inclusion of object trajectories and executable programmatic specifications. We leverage SAM 2 [62] to generate object trajectories and design a custom algorithm to handle objects that do not appear in the first frame. To obtain the specifications, we use GPT-4 and develop a transcompiler that converts them into executable programs.

These two additional components enable fine-grained spatio-temporal alignment between the video content and the concepts expressed in the specifications [31]. Notably, the only source of supervision is the video itself; all other annotations are derived using multimodal large language models (MLLMs). We refer to this approach as *model-driven self-supervision*.

Algorithm 2: Bounding Box Prompt Buffer Class

```
Output: Prompt dictionary for propagation; object-to-class mapping
Initialize fields: oid, frame2bboxes, frame2masks, valid_prompts,
  overlapped_prompts;
// Add predicted bounding box and mask
Function AddPrompt(frame_id, box, mask,)
    if not valid(m) then
        return
    end
    frame2bboxes[frame_id].append(box); frame2masks[frame_id].append(mask)
end
Function PopMostDensePromptFrame()
    // Find frame f with largest mask area
    f = arg max_f sum(frame2masks[f]);
    // Assign object IDs starting from oid for each bbox in f
    foreach box in frame2bboxes[f] do
        valid_prompts[f][oid] = box; oid += 1;
    end
    // Remove prompts from to process list
    frame2bboxes.pop(f); frame2masks.pop(f);
    return f; valid_prompts[f]
end
Function RemoveDuplicatePrompts(video_segments)
    foreach frame_id in video_segments do
        if frame_id not in frame2masks then
            continue
        end
        Compute IoU between predicted and prompt masks for frame_id;
        // Filter prompts already covered
        Remove overlapping segments from frame2bboxes and frame2masks
    end
end
```

A.2 Object Trajectory Generation

We aim to generate object trajectories from videos using Segment Anything 2 (SAM2). A key challenge is discovering new objects that do not appear in the first frame, which is not natively supported by SAM2. To address this, we design an iterative algorithm that identifies the next frame most likely to contain a new object and propagates its mask throughout the video, as illustrated in Algorithm 1. To further leverage concepts generated by GPT-4, we extend this algorithm to incorporate arbitrary frame-based bounding box generators, such as Grounding DINO [21] and YOLO [26]. This extended version uses a prompt scheduler that prioritizes frames containing the highest number of grounded objects and iteratively propagates them across the video, as shown in Algorithm 2 and Algorithm 3.

Algorithm 3: Mask Generation via Frame-wise Bounding Box Grounding

Input: Grounding model G, SAM predictor P, SAM mask generator M, video frames $\{I_0, \ldots, I_T\}$, label set C
Input: Box and text thresholds τ_b, τ_t, target FPS, max propagation steps k
Output: Per-frame mask segments video_segments, object-to-class mapping cls_pred

Initialize prompt_memory ← PromptBuffer();
// Step 1: Grounding-based object detection
foreach frame I_f in video **do**
  bboxes = $G(I_f, C)$;
  video_boxes[f] ← bboxes;
end
// Step 2: Generate masks for all bounding boxes
foreach $(f, \text{bboxes}) \in$ video_boxes **do**
  masks = $M(I_f, \text{bboxes})$;
  foreach (box, mask) in zip(bboxes, masks) **do**
    prompt_memory.add_prompt(f, box, mask)
  end
end
// Step 3: Iterative propagation from dense prompt frame
f_{start}, prompt ← prompt_memory.pop_most_dense_fid_prompt();
video_segments ← {}, $t \leftarrow 0$;
while prompt $\neq$ None and $f_{\text{start}} \neq -1$ and $t < k$ **do**
  Reset predictor state $S \leftarrow S_0$;
  foreach $(f, \text{frame\_prompt}) \in$ prompt_memory.valid_prompts **do**
    foreach $(o, b) \in$ frame_prompt **do**
      Add prompt(o, b) at frame f into P with state S;
    end
  end
  // Propagate forward
  foreach $(f, \{o_i\}, \{\ell_i\}) \in P$.propagate($S$) **do**
    video_segments[f] ← binarized masks $\{\ell_i > 0\}$ mapped to o_i
  end
  // Propagate backward
  foreach $(f, \{o_i\}, \{\ell_i\}) \in P$.propagate($S$, reverse = True) **do**
    video_segments[f] ← binarized masks $\{\ell_i > 0\}$ mapped to o_i
  end
  prompt_memory.remove_dup_prompt(video_segments);
  f_{start}, prompt ← prompt_memory.pop_most_dense_fid_prompt();
  $t \leftarrow t + 1$;
end
return video_segments

A.3 Concept Generation

To construct low-level supervision from high-level captions, we leverage GPT-4 to generate spatio-temporal specifications. To mitigate potential hallucinations from the language model, we design a compiler that verifies the validity of the generated programs. We use a few-shot prompt with three examples.

Prompt 1: Role Definition

```
You are a super user in logic programming.
You are also an expert at structured data extraction.
```

Prompt 2: General Task Instruction

```
You will describe the event length and location in
both natural language and fraction of the video.
```

```
The natural language description of the locations in the video
    can be: early, mid, late.
The natural language description of the durations of the event
    can be: long, medium, short
Examples of precise video locations: [1/4, 1/2], [2/3, 1].
Examples of event durations: 1/4, 2/3, 1.
```

Prompt 3: Few-shot Examples

Listing 1: "One of the few-shot example for concept extraction"

```
Caption: A man carries a child and walks to the left from
    behind a woman holding another child.
Video_ID: "0be30efe",
Action json:
{"video_id": "0be30efe",
 "sequential descriptions": [ "man A carry child B, women C
     hold child D, man A is behind women C", "man A walk", "man
      A at left" ],
  "time stamps":
    { "1": { "description": [ "man A carry child B", "women C
        hold child D", "man A is behind women C" ], "
        programmatic": [ "carrying(A, B)", "name(A, man)", "
        name(B, child)", "holding(C, D)", "name(C, women)", "
        name(D, man)", "behind(A, C)",  ], "duration": "short",
         "duration precise": "1/4", "video location": "early",
         "video location precise": "[0, 1/4]" }, "2": { "
        description": [ "man A walk" ], "programmatic": [ "walk
        (A)", ], "duration": "medium", "duration precise": "1/2
        ", "video location": "mid", "video location precise":
        "[1/4, 3/4]" }, "3": { "description": [ "man A at left"
         ], "programmatic": [ "left(A)" ], "duration": "short",
         "duration precise": "1/4", "video location": "late", "
        video location precise": "[3/4, 1]"}}}
```

Prompt 4: Concept De nition

```
Note all the relations are name, unary or binary.
A name relation takes in two arguments, the first is always a
    variable, and the second argument could be noun ("apple"),
    noun phrase ("ancient_building"), location("dark_forest"),
    etc. For example "name(A, "apple")" means the variable A
    refers to an apple. Please ensure no space occur in the
    second argument.
A unary relation takes in one variable as its argument. For
    example, close(A) means A is close to the camera.
A binary relation takes in two variables as its arguments. For
    example, above(A, B) means A is above B.
The entity in the binary and unary relation are variables in
    the form of capitalized letters (A, B).
The predicate of the unary relation can be adjectives, verbs,
    and name.
The predicate of the binary relation can be preposition, and
    verb.
Please include the name relation any time if applicable.
Please make the preposition and adjectives into separate two
    relation.
```

For each time stamp, please only describe the events that are happening at the same time, if any sequential events occur, put them into multiple time stamps.
For example, instead of 'person A enter from left, person A walk to center, person A move to couch' in one time stamp, put it into three different time stamps: "person A enter from left", "person A walk to center", "person A move to couch".
Please only describe one single event in sequential description per time stamp.
Please use as many relations as possible to precisely describe the action.
Please generate the action json programs for the following captions in the following format:
{"actions": {caption_id: action json programs}}
IMPORTANT: Please REMOVE all new line characters and extra spaces in the generated json!

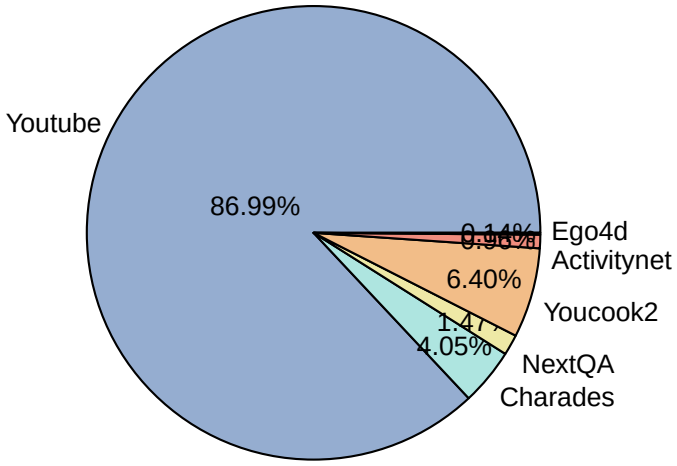


Figure 11: Video sources for ESCA-Video-87K.

A.4 Dataset Statistics

We present the statistics of ESCA-Video-87K through three perspectives: the composition of video sources, the complexity of extracted speci cations, and the word clouds of associated concepts. We selected 0–30 second clips from the VAV-V IDEO-178K dataset CITO, with the resulting source distribution shown in Figure 11. Our extracted spatio-temporal speci cations exhibit considerable complexity. As illustrated in Figure 12, a single speci cation can contain multiple names, actions, relations, and events. In addition, we visualize the vocabulary diversity using word clouds for names, actions, and relations. In total, our dataset includes 220,905 unique names, 57,930 unique actions, and 35,415 unique relation keywords.

(a) Name Count

(b) Action Count

(c) Relation Count

(d) Event Count

Figure 12: Distribution of Names, Attributes, Relations, and Events