We present the following details that are not included in the main manuscript:

- **Datasets**: We provide detailed information on the datasets that we used for the experiments.
- **Prompt Design**: We present detailed information of prompt design for generating semantic state graphs (SSGs) and contextual QA.
- **Validity and Robustness of SSG**: We discuss the verification measures taken to ensure the validity of the generated graphs and explain the mechanism that secures robustness against potential structural noise.
- **Detailed Formulation of Visual-State Alignment**: We provide the detailed mathematical formulation of the contrastive and matching objectives used in the Visual-State Alignment (VSA) loss.
- **Extended Qualitative Analysis**: We provide a deeper analysis of failure cases in baseline models, discussing how structured representations effectively resolve visual grounding ambiguities in unstructured text.

A. Datasets

In this section, we provide a more detailed overview of the three benchmark datasets used in our experiments: COIN, CrossTask, and NIV.

A.1. CrossTask

The CrossTask dataset [33] was specifically created to evaluate the ability of models to learn from instructional videos and generalize across related tasks. It features videos from various domains, including cooking, car maintenance, and crafting. The dataset includes 4,700 videos covering 83 different tasks, which are divided into 18 primary tasks with full temporal annotations and 65 related tasks with weaker supervision. A key feature of CrossTask is the significant overlap in actions and objects across different tasks (e.g., "cutting" and "mixing" appear in many cooking tasks), making it an excellent testbed for evaluating a model's ability to understand fine-grained relational and semantic details, which is a core focus of our HSR model. Each clip comprises an average of 7.6 ± 4.4 actions. We report results on the 133 primary action steps. Note that the original CrossTask dataset has 4,700 videos and 83 tasks, but common practice in prior works focuses on a subset for more rigorous evaluation of core planning capabilities.

A.2. COIN

The COIN (Comprehensive Instructional Video Analysis) dataset [28] is a large-scale collection of instructional videos sourced from YouTube. It is organized hierarchically into 12 distinct domains related to daily life activities, such as "vehicles" and "gadgets." The dataset contains $11,827$ videos covering 180 different tasks. Each video is segmented into a series of action steps with corresponding temporal boundaries and textual descriptions.COIN features an average of 3.9 ± 2.4 actions per video. Its large scale and diverse range of tasks make it a challenging benchmark for evaluating the generalizability of procedural planning models. In our experiments, we use the official splits and report results on its 778 unique action steps.

A.3. NIV

The NIV (Narrated Instructional Videos) dataset [1] is a more recent benchmark designed to leverage narrated instructions as a source of weak supervision. While smaller in scale compared to COIN and CrossTask, it features a dense collection of videos for a focused set of tasks. The dataset contains videos for 5 distinct tasks with a total of 48 unique action steps. With an average of 8.8 ± 2.8 actions per video, NIV has a higher density of actions, making the fine-grained nature of its action vocabulary and the reliance on aligning narration with visual content a challenging benchmark for testing the precision of state representations.

B. Prompt Design

This section provides detailed information on the prompt design used for generating Semantic State Graphs (SSGs) and contextual QA pairs. As illustrated in Table 6, we designed a specific prompt to generate our Hybrid State Representation (HSR) for action prediction.

The prompt is structured to produce three distinct triplets for both the 'before' and 'after' semantic state graphs. These triplets explicitly capture the relations between objects relevant to a given action. Additionally, the prompt generates a single contextual QA pair to resolve potential ambiguities.

To mitigate the risk of hallucinations, which can occasionally occur during LLM-based generation, we incorporated a manual verification and refinement process. After the initial generation, we carefully reviewed all SSGs and QA pairs to ensure they were directly relevant to the corresponding action. Any generated content that was inaccurate or irrelevant was manually corrected to ensure the quality and fidelity of our final representation.

C. Validity and Robustness of Semantic State Graph

Validity Assurance. To ensure the high fidelity of our Semantic State Graphs (SSGs), we employ a rigorous two-stage generation pipeline. First, we utilize GPT-4 with the structured prompts detailed in Table 6 to generate candidate triplets. While Large Language Models (LLMs) provide strong commonsense reasoning, they can occasionally produce hallucinations or task-irrelevant relations. To mitigate this, as mentioned in Section B, we incorporate a post-generation verification step where the generated triplets are

Prompt Design for semantic state graphs (SSGs) and QA
You are the professional for generating state trplets and related QAs. Following the given example, ###Example### Task: "Add Oil to Your Car" Action: "remove cap" before semantic state graph: ⟨semantic state graph⟩ after semantic state graph: ⟨semantic state graph⟩ Contextual QA: Q: "What is the state of the oil cap after the action?", A: "The oil cap is removed." Please generate each three before and three after triplet for the given task and action. Then, generate one contextual QA pair that is related to the objects with the action. Task: "Make a Latte" Action: "add coffee" Video frames: ⟨video frames⟩

Table 6. The Prompt used for the Semantic State Graphs (SSGs) and contextual QAs.

reviewed against the visual context to ensure they explicitly capture the objects and relations relevant to the action.

Robustness against Noise. Despite these verification measures, perfect accuracy in automated graph generation is challenging to guarantee in all scenarios. However, our Hybrid State Representation (HSR) framework is explicitly designed to be robust against such potential structural noise. The core mechanism for this robustness lies in our Heterogeneous State Encoder and Visual-State Alignment (VSA) objective.The Fusion module integrates the structural signals from SSGs (g) with the semantic signals from contextual QA pairs (q) and visual features (v). Crucially, the VSA mechanism acts as a semantic filter during training. If a specific triplet in the SSG contradicts the visual evidence in the video frames, the contrastive alignment loss ($\mathcal{L}_{VSA}$) encourages the model to downweight the misaligned graph embeddings in favor of the correctly grounded visual and QA embeddings. This "hybrid" nature ensures that the planner does not rely solely on the graph structure but leverages the consensus between visual, structural, and textual modalities, thereby maintaining high planning performance even in the presence of minor graph inaccuracies.

D. Detailed Formulation of Visual-State Alignment

In this section, we provide the detailed formulation of the Visual-State Alignment (VSA) objective introduced in Eq. (1). To ground our hybrid state representations in the visual context, we employ a Q-Former architecture. Specifically, we use the visual features of the start and goal frames, denoted as $v_{ctx} = [v_0, v_T]$, as the image input. For the textual/state input, we use the sequence of hybrid state embeddings $s_{1:T}$, which fuses the Semantic State Graph (SSG) and contextual QA embeddings. Using learnable queries $q_{1:T}$, the Q-Former outputs the visually-grounded step embeddings $x^q_{1:T}$:

$$x^q_{1:T} = \text{QFormer}\left(q_{1:T};\ v_{ctx};\ s_{1:T}\right) \tag{3}$$

Following the alignment strategy in BLIP-2, we optimize the Visual-State Contrastive (VSC) loss and the Visual-State Matching (VSM) loss to maximize the mutual information between the visual context and the hybrid state representations.

Visual-State Contrastive (VSC) Loss. Within a batch of size B, we treat the aligned visual-state pairs from the same video as positive, and unmatched pairs as negative. The VSC loss contrasts the embedding similarity of a positive pair against the negative ones using a softmax-normalized objective:

$$\mathcal{L}_{VSC} = -\sum_{j=1}^{B} \log \frac{\exp\left(sim\left(v_{ctx}[j], x^q[j]\right)/\tau\right)}{\sum_{k=1}^{B} \exp\left(sim\left(v_{ctx}[j], x^q[k]\right)/\tau\right)} \tag{4}$$

where $sim(\cdot,\cdot)$ denotes the cosine similarity function, τ is a learnable temperature parameter, and $v_{ctx}[j]$ and $x^q[j]$ represent the visual context features and the output embeddings from the Q-Former for the j-th sample, respectively.

Visual-State Matching (VSM) Loss. The VSM objective tasks the model with a binary classification problem to predict whether a given pair of visual context and hybrid state embedding is matched (positive) or unmatched (negative). This is supervised using the Binary Cross-Entropy (BCE) loss:

$$\mathcal{L}_{VSM} = \sum_{j=1}^{B}\sum_{k=1}^{B} \mathcal{L}_{BCE}\left(y_{match}(v_{ctx}[j], x^q[k]); 1(j=k)\right) \tag{5}$$

where $y_{match}(\cdot)$ is the matching score predicted by a binary classifier head attached to the Q-Former output, and $1(j=k)$ is the indicator function which is 1 if the pair belongs to the same sample and 0 otherwise. The total VSA loss is the weighted sum of these two terms as defined in Eq. (1).

E. Extended Qualitative Analysis

In Figure 3 of the main paper, we compared our HSR model with the PlanLLM baseline. A notable observation in the "Make Kerala Fish Curry" task (Figure 3(a)) is that the baseline fails to predict the correct action step (predicting "Add curry leaves" instead of "Add chili powder"), even though

its generated text description is factually correct ("Add chili powder to the fish curry").

This discrepancy highlights a fundamental limitation of unstructured text representations. While the sentence capture the general semantic content, it lacks the explicit structural constraints necessary for precise visual grounding. In unstructured text, the model's attention mechanism may suffer from "semantic drift," focusing on high-level context words (e.g., "fish curry") rather than the specific object interaction required for the immediate step. Consequently, even with a correct description, the visual alignment may remain ambiguous.

In contrast, HSR explicitly models this state through the Semantic State Graph (SSG) triplet: `(pan, contain, chili powder)`. This structured representation acts as a hard constraint that forces the Visual-State Alignment (VSA) module to bind the visual features to the specific interaction with the "pan," rather than the general context of the dish. This demonstrates that the performance gain of HSR stems not merely from better alignment training, but from the **structural guidance** that disambiguates visual grounding, preventing the model from overlooking fine-grained state changes.