# FAST-PLAN: FAilure Driven Training Data Synthesis for Ambiguity-Robust Embodied Task PLANning in Open Source LLMs

**Souradeep Mukhopadhyay** [1] **Vedang Vasant Avaghade** [1] **Ali Payani** [2] **Gaowen Liu** [2] **Jayanth Srinivasa** [2] **Chitta Baral** [1]

## Abstract

Open-source large language models (LLMs) are increasingly deployed for embodied task planning, yet they remain brittle under ambiguous, human-like instructions characterized by implicit pronouns, underspecified attributes, and noisy dialogue. Consequently, current state-of-the-art open models often suffer severe performance collapse in realistic, ambiguous scenarios. To address this limitation, we introduce **FAST-PLAN**, a scalable, failure-driven data synthesis framework that trains robust planners using their own mistakes. Operating within the AI2-THOR environment, FAST-PLAN programmatically injects linguistic ambiguity into deterministic symbolic tasks. It employs an automated pipeline to generate vague instructions via an LLM augmentor, validate groundings, evaluate execution, and leverage a critic LLM to classify planning failures and generate refined feedback. We fine-tuned Qwen and Llama family models on synthesized trajectories generated by FAST-PLAN. The resulting models outperform baselines on REI-Bench by +50% on success rate. Furthermore, they exhibit strong zero-shot generalization on out-of-distribution ambiguity benchmarks, improving macro-F1 scores by up to 6% on the CLARA benchmark, highlighting the effectiveness of FAST-PLAN.

## 1. Introduction

The deployment of Large Language Models (LLMs) has fundamentally transformed embodied task planning, enabling agents to parse natural language and generate sequential actions in complex environments. However, a critical bottleneck prevents these systems from achieving true real-world utility. When agents are placed in realistic household scenarios, they must process instructions the way humans naturally communicate: ambiguously, implicitly, and with incomplete context (Huang et al., 2022). A standard directive like "heat it and put it there" requires the agent to bridge severe semantic gaps by resolving pronouns and inferring missing destinations based on situational awareness. Unfortunately, standard training paradigms and benchmarks typically assume pristine, fully specified commands. When exposed to realistic linguistic noise, missing object attributes, or irrelevant dialogue, even highly capable open-source models suffer catastrophic performance drops. On ambiguity-focused evaluations such as REI-Bench (Jiang et al., 2025), baseline task success rates collapse to as low as 6 to 9 percent. This brittleness highlights a profound inability to ground vague human language into actionable environmental states (Ahn et al., 2022; Shridhar et al., 2019; Dubey et al., 2024).

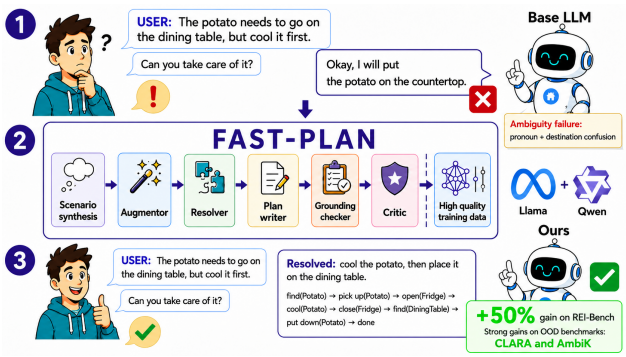


*Figure 1.* FAST-PLAN converts ambiguity-induced planning failures into synthetic training data generation and fine-tunes open-source LLMs with SFT, enabling robust resolution of vague household instructions.

The prevailing strategy to mitigate this brittleness relies on prompting massive, proprietary frontier models (such as GPT-4o (Achiam et al., 2023) or Claude). While these models possess the innate reasoning capabilities to decipher some linguistic ambiguity, they are fundamentally mismatched for scalable robotic deployment. Relying on cloud-based APIs introduces prohibitive latency, strict privacy vulnerabilities, and compounding inference costs that make

[1]Arizona State University [2]Cisco Research. Correspondence to: Souradeep Mukhopadhyay <smukho11@asu.edu>.

*Proceedings of the $43^{rd}$ International Conference on Machine Learning*, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

long-horizon, high-frequency control loops economically unviable. Open-source models present an ideal solution for on-device, zero-cost inference, yet their out-of-the-box instruction following falls short. Current attempts to boost these smaller models use inference-time interventions like Chain-of-Thought (Wei et al., 2022), in-context learning (Brown et al., 2020), and action-aware prompting (Gao et al., 2024). These techniques provide only marginal relief because they merely address the symptom of ambiguity during execution. They fail to resolve the root cause: the complete absence of targeted training signals that teach models how to systematically navigate underspecified embodied planning.

To bridge this gap, what is needed is a paradigm shift from superficial inference-time prompting to deep, failure-driven training supervision. Agents must be trained to recognize and recover from the specific instructional breakdowns they will inevitably encounter. We propose **FAST-PLAN**, a scalable, annotation-free data synthesis framework designed to generate this missing training signal using the AI2-THOR (Kolve et al., 2017) environment. FAST-PLAN operationalizes an automated pipeline that programmatically pairs deterministic symbolic tasks with LLM-generated vague instructions. It then executes these tasks, validates grounding correctness, and employs a critic LLM to classify execution failures across a fine-grained, eight-category error taxonomy. By capturing the model's own mistakes and refining them into corrective trajectories, FAST-PLAN creates highly targeted training data without relying on human annotators or static, perfect demonstrations. Our main contributions are summarized as follows:

**1. Failure-Driven Synthesis Pipeline:** We introduce **FAST-PLAN**, an automated synthetic data generation framework powered by an open-source LLM augmentor. It provides a principled method for generating ambiguous embodied scenarios and extracting targeted training signals from classified planning failures.

**2. Significant Performance Gains:** We demonstrate that supervised fine-tuning of compact open-source models (including the Llama-3.2 and Qwen3 families) on our synthesized data achieves 94 to 96% success rates on REI-Bench. This dramatically outperforms all prompting baselines, including Chain-of-Thought and in-context learning.

**3. Zero-Shot Generalization:** We show strong out-of-distribution generalization capabilities, yielding significant macro-F1 improvements on the CLARA and AmbiK benchmarks. These results validate failure-driven synthetic data as a practical, scalable path toward deploying robust embodied agents in human-centric environments.

## 2. Related Works

**Embodied AI Based on LLMs** Recent years have witnessed a surge in LLM-based embodied task planning, spanning graph-informed action generation, symbolic oversight, and multimodal benchmarking (Li et al., 2026; Yang et al., 2025; Byrd et al., 2025; Liu et al., 2025). Despite these advances, the dominant paradigm relies on frontier models (e.g., GPT-4o, Claude) via prompting. While effective, these approaches impose prohibitive API costs, latency constraints, and privacy risks, making them unsuitable for on-device robotic deployment. Our work diverges by focusing on training smaller, open-source LLMs through failure-driven synthetic supervision, directly addressing this practical deployment gap.

**Training of Embodied Agentic Systems** While prompting dominates, a growing line of work explores training-based approaches to improve agentic behavior, including Supervised Fine-Tuning (SFT), reinforcement learning, and knowledge distillation for small language models (Ghasemipour et al., 2025; Pham et al., 2025; Boyle et al., 2025; Liu et al., 2026). However, prior synthetic data generation focuses largely on long-horizon manipulation or path planning. To our knowledge, **FAST-PLAN** is the first to generate synthetic training data specifically targeting vague and ambiguous instruction-following failures, moving beyond static demonstrations to failure-driven supervision grounded in a principled eight-category error taxonomy.

**Benchmarks for Embodied AI with Vague Instructions** Benchmarking ambiguity in embodied AI has emerged as a critical frontier. Existing benchmarks evaluate instruction clarity and coreferential vagueness (CLARA (Park et al., 2023), REI-Bench (Jiang et al., 2025)), textual and visual intent disambiguation (AmbiK (Ivanova et al., 2025), VAGUE (Nam et al., 2024)), and open-vocabulary 3D ambiguity (Ambi3D (Ding et al., 2026)). Together, they reveal that even state-of-the-art models suffer severe performance drops under ambiguity arising from pronouns, missing referents, and underspecified attributes. Yet, while these works extensively diagnose the problem, no prior method provides targeted *training* data to resolve it. **FAST-PLAN** is designed to fill this precise gap.

## 3. Method

We propose **FAST-PLAN**, a six-stage, annotation-free pipeline designed to synthesize failure-driven training data for embodied task planning. The core motivation is to systematically expose small open-source LLMs to the precise failure modes they encounter in practice, rather than relying exclusively on pristine expert demonstrations. To achieve this, FAST-PLAN operates as a hybrid neuro-symbolic engine: it relies on deterministic rules to guarantee physical

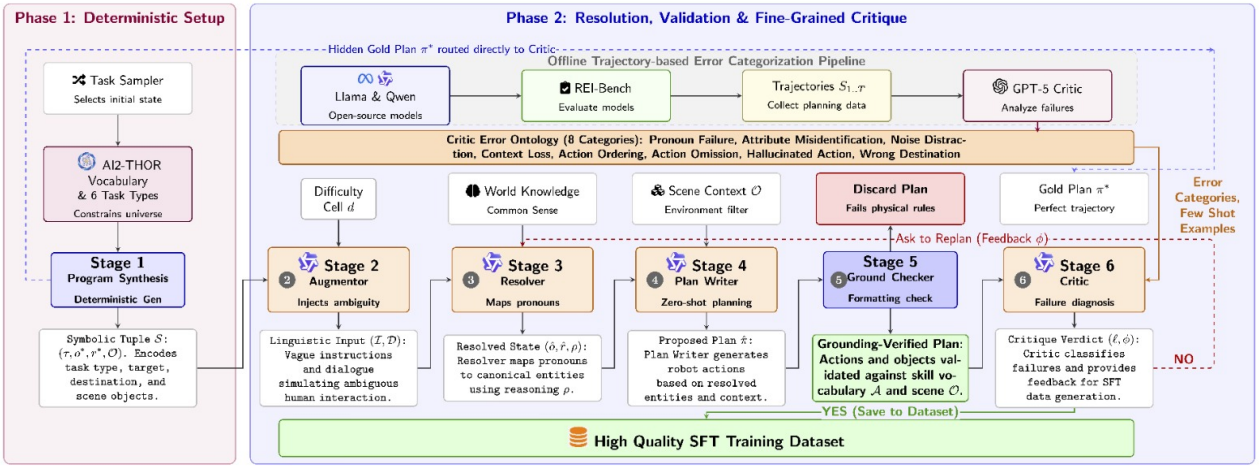


*Figure 2.* Overview of the FAST-PLAN data generation and validation pipeline. Phase 1 deterministically samples AI2-THOR task configurations and synthesizes symbolic task tuples, defining the target object, destination, scene context, and gold plan. Phase 2 converts these tuples into ambiguous linguistic instructions, resolves entities, generates candidate robot plans, and validates them through grounding checks and critic-based failure diagnosis. Invalid plans are rejected or sent for replanning with feedback, while grounding-verified plans are retained as high-quality SFT training data.

ground truth, while utilizing LLMs solely where semantic reasoning and linguistic variation are required. Figure 2 illustrates the complete pipeline.

### 3.1. Stage 1: Programmatic Scenario Synthesis

The pipeline begins by establishing a rigorous physical ground truth without any LLM involvement. We define a scenario as a deterministic tuple $\mathcal{S} = (\tau, o^*, r^*, \mathcal{O}, \pi^*)$. Here, $\tau$ represents a canonical task type (e.g., `heat_and_place`), $o^*$ and $r^*$ are the physically viable target object and destination receptacle, and $\mathcal{O}$ defines the populated 3D scene including distractors. The gold action sequence $\pi^*$ is instantiated deterministically from standard AI2-THOR templates. Because $\pi^*$ is constructed symbolically, we guarantee absolute ground-truth correctness by construction, entirely eliminating the risk of LLM hallucination in the target label.

### 3.2. Stage 2: The Augmentor

With the physical reality fixed, the Augmentor injects controlled linguistic chaos. Powered by an open-source frontier model (`Qwen3-235B`), it generates a natural language instruction $\mathcal{I}$ and multi-turn dialogue context $\mathcal{D}$. This generation is conditioned on a strict difficulty matrix, varying the explicitness of object references against the noisiness of the dialogue. For example, an *explicit* instruction clearly names the "potato," while an *implicit* instruction forces the agent to resolve pronouns like "it" or spatial markers like "there" based on the surrounding conversation. This stage produces the rich spectrum of human-like vagueness required for robust training. System Prompts are available in App. A.

### 3.3. Stage 3: The Resolver

The Resolver $\mathcal{R}$ acts as the bridge between human vagueness and physical reality. It processes $\mathcal{I}$ and $\mathcal{D}$ to perform explicit coreference resolution, outputting a structured JSON containing a natural language reasoning chain $\rho$, the resolved target $\hat{o}$, and the destination $\hat{r}$. This step grounds subsequent actions and exposes the model to the step-by-step disambiguation logic required for success. A resolution is deemed correct only if $\hat{o} = o^*$ and $\hat{r} = r^*$; any mismatch naturally generates a contrastive failure signal. System Prompts are available in App. A.

### 3.4. Stage 4 & 5: Plan Generation and Grounding Verification

Operating zero-shot, the Plan Writer $\mathcal{P}$ regenerates an executable action sequence $\hat{\pi}$ from the Resolver's output and the scene description $\mathcal{O}$. Crucially, it does not have access to the gold plan $\pi^*$. This forces a genuine generative test that naturally exposes sequential planning flaws. Immediately following generation, the deterministic Grounding Checker $\mathcal{G}$ intercepts $\hat{\pi}$ to enforce strict physical laws: all verbs must belong to the AI2-THOR skill vocabulary $\mathcal{A}$, all object arguments must exist in the scene $\mathcal{O}$, and the plan must logically terminate. Plans failing these hard constraints are discarded, ensuring the final dataset is free from physically impossible hallucinations. System Prompts are available in App. A.

### 3.5. Stage 6: The Critic and Data Auditing

The Critic $\mathcal{C}$ is the semantic engine of the pipeline. It evaluates the full interaction tuple $(\mathcal{I}, \mathcal{D}, \rho, \hat{\pi}, \pi^*)$ to perform fine-grained failure classification. Utilizing the eight stan-

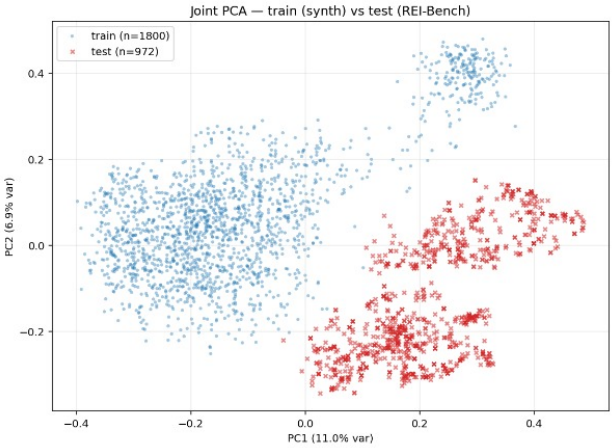


*Figure 3.* Joint PCA visualization of synthesized SFT training examples and REI-Bench test examples. Blue points denote synthetic training data, while red crosses denote REI-Bench test data, illustrating the distributional separation between the two sets in the shared embedding space.

dard REI-Bench error categories (such as *pronoun failure* or *action omission* metioned in App. D), the Critic outputs a structured verdict:

$$\mathcal{C}(\mathcal{I}, \mathcal{D}, \rho, \hat{\pi}, \pi^*) = \{ \texttt{label} : \ell \in \mathcal{E} \cup \{\texttt{correct}\}, \texttt{feedback} : \phi \} \tag{1}$$

where $\ell$ is the assigned label and $\phi$ provides natural language corrective feedback. Correct instances serve as positive demonstrations, while failure-labeled instances are used to refine the model's error-recovery capabilities, resulting in a highly targeted dataset balanced across all ambiguity conditions. System Prompts are available in App. A.

Finally, to ensure the integrity of our synthesized corpus, we conducted a rigorous data leakage audit. By encoding the training instances alongside the REI-Bench test instances using TEXT-EMBEDDING-3-LARGE, we visualize a joint PCA projection (Fig. 3). The synthetic and test examples occupy distinctly separate regions of the embedding space. This confirms our dataset provides genuine generalization of disambiguation behavior, rather than resulting in simple memorization of evaluation targets.

# 4. Experimental Result

## 4.1. Experimental Setups

**Models.** We evaluate FAST-PLAN on three open-source instruction-tuned LLM backbones: Llama-3.2-1B-Instruct, Llama-3.2-3B-Instruct, and Qwen3-4B-Instruct. For each backbone, we compare the original instruction-tuned model against its supervised fine-tuned variant trained on the FAST-PLAN synthesized corpus. To assess whether ambiguity robustness can be obtained purely at inference time, we also evaluate several prompting baselines, including normal prompting, action-aware prompting (AP), chain-of-thought prompting (CoT), in-context learning (ICL), and task-oriented contextual clarification (ToCC). This allows us to isolate the effect of failure-driven SFT from prompt-engineering improvements.

**Benchmarks.** We evaluate in-distribution performance on REI-Bench, which tests embodied task planning under vague and ambiguous human instructions. REI-Bench contains nine ambiguity settings formed by crossing three reference types—explicit, mixed, and implicit—with three context conditions—standard, noised, and short. To measure out-of-distribution generalization, we additionally evaluate on CLARA and AmbiK. CLARA tests whether robotic commands are clear, ambiguous, or infeasible, providing a broader ambiguity-classification setting. AmbiK evaluates ambiguous kitchen-task instructions across categories such as direct instructions, preferences, commonsense knowledge, and safety, allowing us to test whether the learned disambiguation behavior transfers beyond the REI-Bench scene grammar.

**Evaluation Metrics.** For REI-Bench and AmbiK, we report task success rate (SR), defined as the percentage of examples for which the model produces a valid plan matching the intended target object, destination, and required action sequence. We report SR separately for each ambiguity category and also summarize overall performance across categories. For CLARA, we report macro-F1, which gives equal weight to each class and is therefore appropriate for measuring robustness across both frequent and rare ambiguity labels. In addition to aggregate scores, we analyze category-wise performance to determine whether gains come from broad ambiguity robustness rather than improvements on only the easiest settings.

## 4.2. Main Results

### 4.2.1. IN-DISTRIBUTION REIBENCH

Table 1 summarises the gap between inference-time prompting and supervised fine-tuning on REI-Bench. Across every one of the nine ambiguity categories and every one of the three backbones, the SFT variant of the model is the strongest configuration, with bolded cells filling the entire bottom row of each block. The improvement over the un-modified base model is dramatic: Llama-3.2-1B-Inst. climbs from a 13.0–40.7% range to a 92.6–95.4% range, Llama-3.2-3B-Inst. from 28.7–49.1% to 94.4–96.3%, and Qwen3-4B-Instruct-2507 from 25.9–39.8% to 93.5–98.1%, lifting per-category success rates by 50–80 absolute points. Critically, this gain cannot be recovered by any of the four prompting strategies we tested: even the strongest

*Table 1.* REI-Bench success rate (%) on the 9 ambiguity categories. For each model we compare the base instruction-tuned variant under five inference-time settings (no prompt strategy, +AP, +CoT, +ICL, +ToCC) against the same model after supervised fine-tuning on REI-Bench (SFT). Ranking is per-model, per-category: **bold** marks the best of that model's six variants in each column; underline marks the second best.

| Model | Variant | Explicit | | | Mixed | | | Implicit | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Std | Noised | Short | Std | Noised | Short | Std | Noised | Short |
| Llama-3.2-1B-Inst. | Normal | 35.2 | 40.7 | 25.0 | 19.4 | 9.3 | 17.6 | 21.3 | 13.0 | 13.0 |
| | +AP (Gao et al., 2024) | 39.8 | 38.9 | 28.7 | 14.8 | 8.3 | 20.4 | 16.7 | 11.1 | 16.7 |
| | +CoT (Wei et al., 2022) | 29.6 | 29.6 | 22.2 | 11.1 | 5.6 | 12.0 | 13.0 | 7.4 | 13.9 |
| | +ICL (Brown et al., 2020) | 39.8 | 51.9 | 33.3 | 25.0 | 25.9 | 29.6 | 19.4 | 34.3 | 22.2 |
| | +ToCC (Jiang et al., 2025) | 34.3 | 32.4 | 20.4 | 32.4 | 25.0 | 22.2 | 33.3 | 28.7 | 20.4 |
| | SFT | **93.5** | **92.6** | **92.6** | **94.4** | **94.4** | **95.4** | **92.6** | **95.4** | **94.4** |
| Llama-3.2-3B-Inst. | Normal | 42.6 | 49.1 | 45.4 | 35.2 | 41.7 | 42.6 | 28.7 | 48.1 | 37.0 |
| | +AP (Gao et al., 2024) | 40.7 | 49.1 | 44.4 | 35.2 | 46.3 | 45.4 | 28.7 | 48.1 | 42.6 |
| | +CoT (Wei et al., 2022) | 39.8 | 41.7 | 36.1 | 35.2 | 34.3 | 25.0 | 37.0 | 35.2 | 38.0 |
| | +ICL (Brown et al., 2020) | 41.7 | 51.9 | 50.9 | 39.8 | 46.3 | 44.4 | 28.7 | 50.9 | 45.4 |
| | +ToCC (Jiang et al., 2025) | 44.4 | 43.5 | 47.2 | 42.6 | 46.3 | 34.3 | 35.2 | 50.0 | 35.2 |
| | SFT | **96.3** | **96.3** | **96.3** | **96.3** | **96.3** | **95.4** | **96.3** | **96.3** | **94.4** |
| Qwen3-4B-Instruct-2507 | Normal | 36.1 | 39.8 | 26.9 | 31.5 | 35.2 | 25.9 | 35.2 | 35.2 | 28.7 |
| | +AP (Gao et al., 2024) | 36.1 | 34.3 | 26.9 | 30.6 | 30.6 | 25.0 | 34.3 | 34.3 | 29.6 |
| | +CoT (Wei et al., 2022) | 26.9 | 20.4 | 23.1 | 25.9 | 34.3 | 31.5 | 30.6 | 33.3 | 30.6 |
| | +ICL (Brown et al., 2020) | 67.6 | 63.9 | 62.0 | 48.1 | 56.5 | 60.2 | 68.5 | 59.3 | 63.0 |
| | +ToCC (Jiang et al., 2025) | 36.1 | 38.0 | 31.5 | 25.9 | 30.6 | 26.9 | 33.3 | 32.4 | 29.6 |
| | SFT | **98.1** | **98.1** | **98.1** | **97.2** | **93.5** | **94.4** | **97.2** | **96.3** | **94.4** |

non-SFT entry in the table—Qwen3-4B-Inst. with in-context learning, which reaches 67.6–68.5% on the explicit and `implicit_standard` columns—still trails its own SFT counterpart by roughly 30 points, and on the harder `mixed` and `implicit` conditions ICL drops into the 48–63% band where the same SFT model is at 93–97%. AP, CoT and ToCC are weaker still, often degrading the base score (e.g. CoT cuts Llama-1B-Inst.'s `explicit_standard` score from 35.2 to 29.6 and Qwen3-4B-Inst.'s `explicit_noised` from 39.8 to 20.4), confirming that the failure mode targeted by REI-Bench—resolving referential ambiguity in embodied instructions—is not merely a prompt-formatting problem but requires the model to internalise the disambiguation behaviour through fine-tuning. The prompting underlines in the table also reveal a stable secondary pattern: ICL is the dominant runner-up in 21 of 27 model×category cells, especially for the larger Qwen backbone, suggesting that exemplar-based grounding is the only inference-time intervention that provides consistent, if still insufficient, improvement over the base model.

#### 4.2.2. Out of Order Distribution

On CLARA's 5,345-instruction macro-F1 evaluation, fine-tuning lifts Llama–3.2–3B-Inst. from 18.29% to 24.43% (Fig. 4, +6.14 absolute). Because macro-F1 weights every class equally, this gain reflects improved coverage of CLARA's minority categories rather than a shift toward the dominant label, suggesting that the disambiguation behavior learned on REI-Bench transfers to a broader taxonomy of clarification decisions. On AmbiK (Fig. 4), the same Llama–3.2–3B-Inst. SFT checkpoint improves over its instruction-tuned base on the categories where the benchmark expects a confident, well-grounded plan: *unambiguous direct* (89.4 → 92.1, +2.7 SR), *preferences* (34.5 → 35.0, +0.5), and notably *safety* (34.6 → 40.4, +5.8), with the safety gain being the largest single-category effect we observe on AmbiK. These cross-benchmark improvements—obtained without exposing the model to a single CLARA or AmbiK example during training—indicate that REI-Bench supervision encodes a form of disambiguation competence that generalizes beyond its own scene grammar, particularly for safety-critical and rare-class behaviors. Qwen Results on CLARA are mentioned in App. B.

### 4.3. Ablation Studies

**Effect of per category training data** To probe how the SFT gains depend on supervision volume, we re-train Llama-3.2-3B-Inst. while varying the number of training samples per REI-Bench category, $n \in \{50, 100, 150, 200\}$, keeping every other hyper-parameter unchanged (Fig. 5). Llama-3.2-3B-Inst. exhibits a sharp phase transition: 50, 100, and 150 samples per category produce comparable mean SRs of 81.80%, 75.71%, and 74.49%, respectively, with non-monotonic per-category behaviour. For example, `implicit_standard` bottoms out at 64.8% under $n$=150. Increasing the budget to $n$=200 raises the mean SR to 95.99% and flattens every category into the 94–96%

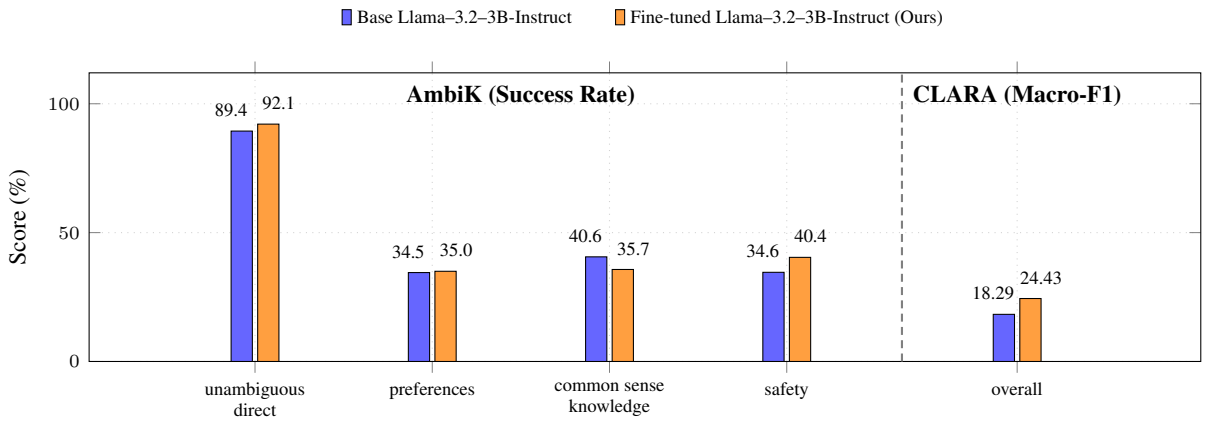


*Figure 4.* Out-of-domain benchmark results for Llama–3.2–3B–Inst. before and after SFT. AmbiK reports success rate across ambiguity categories, while CLARA reports macro-averaged F1 on the overall benchmark. Both benchmarks are shown on the same percentage scale, and the dashed line separates the two evaluation settings.

band, gaining nearly 14 absolute points over $n$=150. We interpret this jump as evidence that REI-Bench's implicit and mixed ambiguity categories contain a long-tailed distribution of disambiguation patterns, requiring a critical mass of approximately 200 samples per category for the smaller Llama-3B-Inst. backbone to consolidate the desired behaviour. Practically, this motivates using $n$=200 samples per category as the default budget for Llama-class backbones. Qwen Ablation study is found in App.C.

**Number of ambiguity class in Training data** We next isolate how many of REI-Bench's nine ambiguity classes ({explicit, mixed, implicit}$\times${standard, noised, short}) the model needs to observe during fine-tuning. Holding the per-class budget at 200 samples, we sweep the number of distinct classes in the training mixture: 1 class (200 samples), 4 classes (800 samples), and the full 9 classes (1,800 samples), while keeping all other hyper-parameters fixed (Fig. 6). Llama-3.2-3B-Inst. requires coverage of the full taxonomy: mean SR climbs $62.73 \rightarrow 78.31 \rightarrow 95.99\%$ across the sweep. The $4 \rightarrow 9$ class jump ($+17.68$) is almost as large as the $1 \rightarrow 4$ class jump ($+15.58$), indicating that partial class coverage is insufficient for this backbone. The 4-class run still shows strong non-uniformity across cells, with explicit conditions at 79.6–84.3% and mixed/implicit conditions at 75.0–77.8%. Combined with the per-class budget ablation, this suggests that REI-Bench-quality SFT for Llama-3B-Inst. requires approximately 200 samples per class across all nine ambiguity classes. Qwen Ablation study is found in App.C.

**Diversity of the Synthesized SFT Training Data** Figure 7 shows that the synthesized SFT training set spans a broad embedding space rather than collapsing into a narrow cluster. In the left panel, examples from the explicit, implicit, and mixed referring-expression types are interleaved across multiple PCA regions, indicating that the dataset covers diverse ambiguity forms rather than separating into isolated template-specific groups. In the right panel, the noised, short, and standard context variants also occupy overlapping but widely distributed regions, showing that linguistic surface form and context style introduce substantial variation while preserving shared task structure. Together, these PCA projections suggest that the 1,800 training instances provide diverse semantic and contextual coverage for fine-tuning.

### 4.4. Error Analysis of Baselines

We analyse the baseline REI-Bench results and identify **8 error categories** that account for all observed instruction-following failures across the 2 Llama models. They are: *(1) Action Omission, (2) Action Ordering, (3) Hallucinated Action, (4) Attributive Misidentification, (5) Pronoun Failure, (6) Noise Distraction, (7) Context Loss,* and *(8) Wrong Destination*. Details of them are in App. D. As shown in Figure 8, the error distribution differs noticeably between Llama 3.2 3B Inst and Llama 3.2 1B Inst. For Llama 3.2 3B Inst, errors are dominated by *Action Omission* at 47.20%, followed by *Action Ordering* at 37.41%. This suggests that the larger model most often fails by leaving out required actions or placing actions in the wrong sequence. Smaller categories include *Attributive Misidentification* at 6.12%, *Hallucinated Action* at 5.24%, *Pronoun Failure* at 2.62%, and *Wrong Destination* at 1.40%. In contrast, Llama 3.2 1B Inst shows a more distributed error pattern. Its largest category is *Action Ordering* at 36.61%, followed by *Hallucinated Action* at 22.44%, *Action Omission* at 20.21%, and *Attributive Misidentification* at 17.32%. This indicates that the smaller model not only struggles with ordering and omission, but also produces more hallucinated actions and attribution errors than the 3B model. Minor error categories include *Pronoun Failure* at 2.76%, *Wrong Destination* at

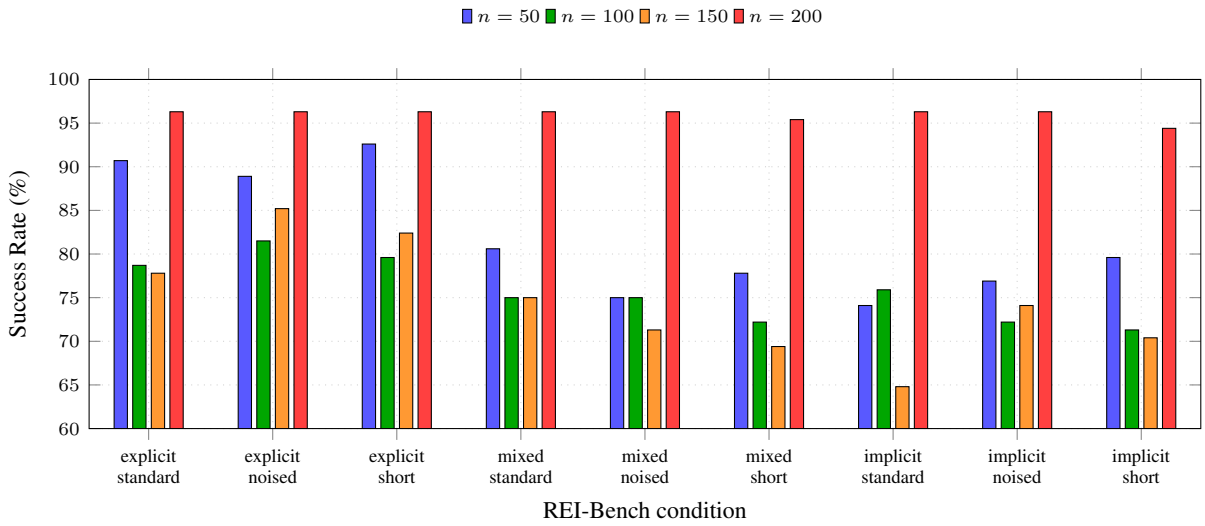


*Figure 5.* Llama–3.2–3B-Instruct SFT scaling on REI-Bench across all nine ambiguity conditions.

0.39%, *Noise Distraction* at 0.13%, and *Context Loss* at 0.13%.

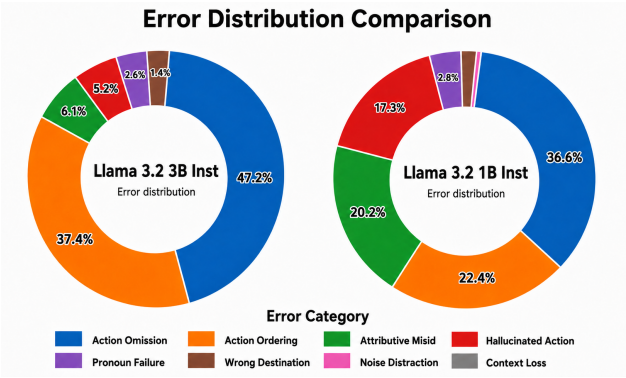


*Figure 8.* Error category distribution for Llama 3.2 3B Inst and Llama 3.2 1B Inst.

## 5. Conclusion

We presented **FAST-PLAN**, a scalable, annotation-free, failure-driven framework that transforms open-source LLM planning mistakes into high-quality supervised fine-tuning signals. It achieves this by programmatically injecting linguistic ambiguity into AI2-THOR scenarios and classifying failures across eight fine-grained error categories. Fine-tuning Llama-3.2-1B-Inst., Llama-3.2-3B-Inst., and Qwen3-4B-Inst. on synthesized trajectories yields 94 to 96% success rates on REI-Bench, achieving a 50 percentage point improvement over advanced prompting baselines. Crucially, these models exhibit strong zero-shot transfer to CLARA (+6.14 macro-F1) and AmbiK (+5.8 on safety-critical categories), confirming that the improvements reflect genuine disambiguation competence rather than memorization. These results establish that LLM fragility under human-like ambiguity is fundamentally a training data problem, and FAST-PLAN offers a practical solution. Looking forward, exciting avenues include incorporating visual observations for perception-grounded failure classification, exploring online failure harvesting for continual self-improvement, and extending the scenario synthesizer to richer compositional tasks beyond the current six AI2-THOR categories.

## Impact Statement

This work advances the capacity of small, open-source language models to follow ambiguous, human-like instructions in embodied planning settings — a capability with direct relevance to assistive robotics, smart home automation, and human-robot interaction for diverse user populations including elderly individuals and those with limited expressive ability. By eliminating dependence on frontier API-based models, FAST-PLAN enables privacy-preserving, on-device deployment at zero inference cost, lowering barriers to accessible robotic assistance. The annotation-free nature of our pipeline further democratizes high-quality training data generation for resource-constrained research groups. We do not foresee significant negative societal impacts from this work; the failure-driven taxonomy and grounding checker introduced here actively reduce the risk of unsafe or incorrect robot actions by making planning failures explicit and correctable. As with any system that enables greater autonomy in physical agents, responsible deployment in safety-critical environments should involve appropriate human oversight.

## References

Achiam, O. J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., de Almeida, D. M., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji,

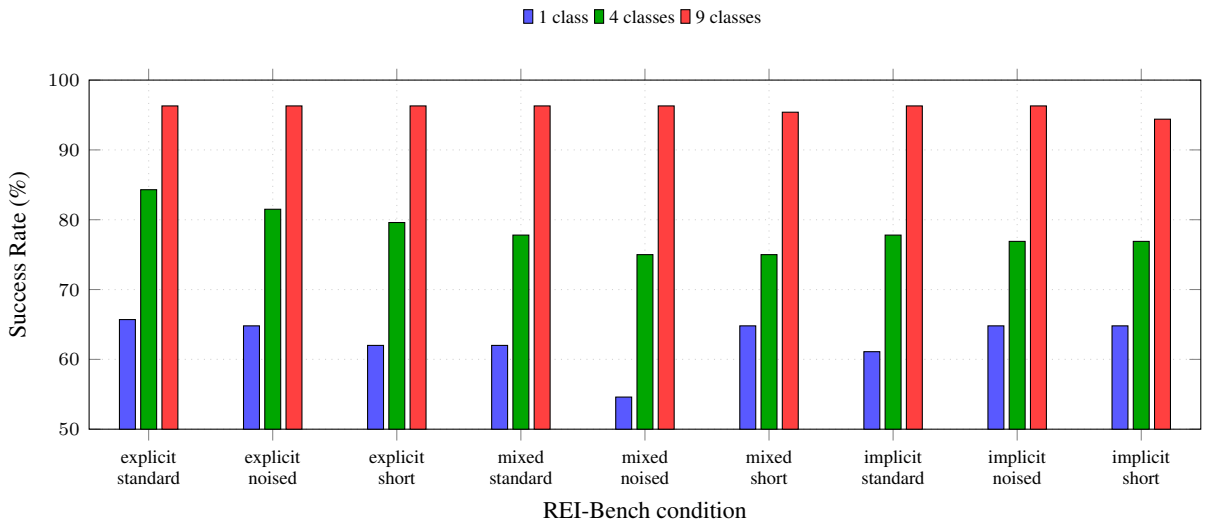


*Figure 6.* Llama–3.2–3B-Inst. class-coverage ablation on REI-Bench across all nine ambiguity conditions. Each selected class is trained with 200 examples; the average summary subplot is omitted.

*Figure 7.* Training Data Diversity

S., Balcom, V., Baltescu, P., ing Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., laine Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., abella Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Kaiser, L., Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, H., Kiros, J. R., Knight, M., Kokotajlo, D., Kondraciuk, L., Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C., Lim, R., Lin, M., Lin, S. L., teusz Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D. P., Mu, T., Murati, M., Murk, O., M'ely, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Long, O., O'Keefe, C., Pachocki, J. W., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Pokorny, M., Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J. W., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M. D., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N. M., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N. A., Thompson, M., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C. L., Wang, J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao,

K., Xu, T., Yoo, S., Yu, K., ing Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report. 2023. URL https://api.semanticscholar.org/CorpusID:257532815.

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. M. J., Jeffrey, K., Jesmonth, S., Joshi, N. J., Julian, R. C., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D. M., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., and Yan, M. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:247939706.

Boyle, L., Baumann, N., Sivasothilingam, P., Magno, M., and Benini, L. Robotxr1: Enabling embodied robotic intelligence on large language models through closed-loop reinforcement learning. *ArXiv*, abs/2505.03238, 2025. URL https://api.semanticscholar.org/CorpusID:278339497.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., teusz Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. URL https://api.semanticscholar.org/CorpusID:218971783.

Byrd, G., Rivera, C., Kemp, B., Booker, M., Schmidt, A., de Melo, C., Seenivasan, L., and Unberath, M. Constrained natural language action planning for resilient embodied systems. *ArXiv*, abs/2510.06357, 2025. URL https://api.semanticscholar.org/CorpusID:281892228.

Ding, J., Tang, H., Jin, H., Gao, W., and Li, G. 3d instruction ambiguity detection. 2026. URL https://api.semanticscholar.org/CorpusID:284597138.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A. S., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Rozière, B., Biron, B. M., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D. J., Pintz, D., Livshits, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E. A., Lobanova, E. I., Dinan, E., Smith, E. M., Radenovic, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I. M., Misra, I., Evtimov, I., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J.-Q., Alwala, K. V., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K. R., El-Arini, K., Iyer, K., Malik, K., ley Chiu, K., Bhalla, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., hesh Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M. H. M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., lay Bashlykov, N., Bogoychev, N., Chatterji, N. S., Duchenne, O., cCelebi, O., Alrassy, P., Zhang, P., Li, P., Vasić, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Girdhar, R., Patel, R., Sauvestre, R., nie Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., hana Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S. C., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., ney Meers, W., Martinet, X., Wang, X., Tan, X. E., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A. K., Grattafiori, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Vaughan, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Franco, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, P.-Y. B., Loyd, B., de Paola, B., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Wang, C., Kim, C.,

Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Civin, D., Beaty, D., Kreymer, D., Li, S.-W., Wyatt, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Ozgenel, F., Caggioni, F., Guzmán, F. P., Kanayet, F. J., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Thattai, G., Herman, G., Sizov, G., Zhang, G., Lakshminarayanan, G., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Molybog, I., Tufanov, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., KamHou, U., Saxena, K., Prasad, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Huang, K., Chawla, K., Lakhotia, K., Huang, K., Chen, L., Garg, L., Lavender, A., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Tsimpoukelli, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., ish Bansal, M., Santhanam, N., Parks, N., White, N., ata Bawa, N., Singhal, N., Egebo, N., Usunier, N., Laptev, N. P., Dong, N., Zhang, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., dro Rittner, P., Bontrager, P., Roux, P., Dollár, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Li, R., Hogan, R., Battey, R., Wang, R., han Maheswari, R., Howes, R., Rinott, R., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S. Y., Pan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Feng, S., Lin, S., Zha, S., Shankar, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S. K., Cho, S.-B., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Kohler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V. O., Montanez, V., Mohan, V., Kumar, V., Mangla, V., Ionescu, V., Poenaru, V. A., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wang, X., Wu, X., Wang, X., Xia, X., Wu, X., Gao, X., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Wang, Y., Hao, Y., Qian, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., and Zhao, Z. The llama 3 herd of models. 2024. URL https://api.semanticscholar.org/CorpusID:271571434.

Gao, J., Gan, L., Li, Y., Ye, Y., and Wang, D. Dissecting dissonance: Benchmarking large multimodal models against self-contradictory instructions. *ArXiv*, abs/2408.01091, 2024. URL https://api.semanticscholar.org/CorpusID:271693415.

Ghasemipour, S. K. S., Wahid, A., Tompson, J., Sanketi, P. R., and Mordatch, I. Self-improving embodied foundation models. *ArXiv*, abs/2509.15155, 2025. URL https://api.semanticscholar.org/CorpusID:281393945.

Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *ArXiv*, abs/2201.07207, 2022. URL https://api.semanticscholar.org/CorpusID:246035276.

Ivanova, A., Bakaeva, E., Volovikova, Z., Kovalev, A. K., and Panov, A. I. Ambik: Dataset of ambiguous tasks in kitchen environment. *ArXiv*, abs/2506.04089, 2025. URL https://api.semanticscholar.org/CorpusID:279155321.

Jiang, C., Zhou, C., and Yang, J. Rei-bench: Can embodied agents understand vague human instructions in task planning? *ArXiv*, abs/2505.10872, 2025. URL https://api.semanticscholar.org/CorpusID:278714946.

Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., Kembhavi, A., Gupta, A. K., and Farhadi, A. Ai2-thor: An interactive 3d environment for visual ai. *ArXiv*, abs/1712.05474, 2017. URL https://api.semanticscholar.org/CorpusID:28328610.

Li, X., Yan, N., and Mortazavi, M. S. Embodied task planning via graph-informed action generation with large lanaguage model. *ArXiv*, abs/2601.21841, 2026. URL https://api.semanticscholar.org/CorpusID:285139940.

Liu, J., Nie, B., Li, B., Chen, Y., Wang, Y., He, S., and Li, H. Robogpt-r1: Enhancing robot planning with reinforcement learning. *ArXiv*, abs/2510.14828, 2025. URL https://api.semanticscholar.org/CorpusID:282138891.

Liu, Y., Wang, A., and Ye, N. V-cage: Context-aware generation and verification for scalable long-horizon embodied tasks. *ArXiv*, abs/2601.15164,

2026. URL https://api.semanticscholar.org/CorpusID:284917499.

Nam, H., Ahn, J., Ka, K., Chung, J., and Yu, Y. Vague: Visual contexts clarify ambiguous expressions. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1537–1547, 2024. URL https://api.semanticscholar.org/CorpusID:276928069.

Park, J., Lim, S., Lee, J., Park, S., Chang, M., Yu, Y., and Choi, S. Clara: Classifying and disambiguating user commands for reliable interactive robotic agents. *IEEE Robotics and Automation Letters*, 9:1059–1066, 2023. URL https://api.semanticscholar.org/CorpusID:259202592.

Pham, Q. P. M., Nguyen, K. T. N., Doan, N. H., Pham, C. A., Inui, K., and Song, D. Smallplan: Leverage small language models for sequential path planning with simulation-powered, llm-guided distillation. *ArXiv*, abs/2505.00831, 2025. URL https://api.semanticscholar.org/CorpusID:278310557.

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., and Fox, D. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10737–10746, 2019. URL https://api.semanticscholar.org/CorpusID:208617407.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E. H., Xia, F., Le, Q., and Zhou, D. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022. URL https://api.semanticscholar.org/CorpusID:246411621.

Yang, R., Chen, H., Zhang, J., Zhao, M., Qian, C., Wang, K., Wang, Q., Koripella, T. V., Movahedi, M., Li, M., Ji, H., Zhang, H., and Zhang, T. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *ArXiv*, abs/2502.09560, 2025. URL https://api.semanticscholar.org/CorpusID:276317279.

# A. System Prompts used in Data genration

**Augmenter – generates a (dialogue, instruction) pair for a target (re_type, context_type, task) cell**

**SYSTEM**

```
You write fresh AI2THOR scenarios for a household robot from a
SCENARIO SPECIFICATION. The specification fixes the task type, target object,
and final receptacle (if any). Your job is to invent a natural human-robot
dialogue and a final instruction that match a desired (re_type, context_type)
cell.

re_type controls the surface form of the final instruction:
  explicit  -- name the target object directly.
               Example: "Put the apple on the dining table."
  mixed     -- describe the target attributively (no pronoun).
               Example: "Put the round red fruit on the dining table."
  implicit  -- use a pronoun or definite reference whose antecedent is in
               the dialogue.
               Example: "Put it on the dining table."

context_type controls the dialogue:
  standard -- 3-5 turn dialogue that grounds the referent naturally
  noised   -- 3-5 turns with 1-2 distractor mentions of OTHER objects from
              scene_objects (the wrong ones)
  short    -- 1-2 turns, minimal context

Hard constraints:
  - The instruction MUST imply the SAME target_object and (when applicable)
    the SAME receptacle as the spec. Do NOT pick a different target.
  - The instruction MUST imply the FULL task semantics for task_type.
    A bare "put X on Y" is only valid for pick_and_place.
  - Never invent objects outside scene_objects. Distractors must be drawn
    from scene_objects.
  - For implicit/mixed, an antecedent or attributive cue for the target
    MUST appear in the dialogue.
  - Dialogue uses "Human:" / "Robot:" turn prefixes, separated by newlines.
  - Output STRICT JSON only -- no markdown fences, no commentary.

Task-type-specific instruction requirements:
  pick_and_place  : "put / place / move <target> on/in <receptacle>"
  stack_and_place : must imply BOTH stacking AND moving
  clean_and_place : must include a cleaning verb --
                    "wash / clean / rinse <target>, then put it on <receptacle>"
  heat_and_place  : must include a heating verb --
                    "heat / warm / microwave <target>, then put it on <receptacle>"
  cool_and_place  : must include a cooling verb --
                    "cool / chill / refrigerate <target>, then put it on <receptacle>"
  examine_in_light: must imply examining under a lamp --
                    "examine / inspect / look at <target> under the <lamp>"
                    -- NO receptacle

Output schema:
  {"instruction": "<string>", "context_memory": "<string>"}
```

**USER**

```
Scenario specification:
  task_type:        {task_type}
  target_object:    {target_object}
  receptacle:       {receptacle}
  secondary_object: {secondary}
  scene_objects:    {scene_objects}

Desired cell:
  re_type:      {re_type}
  context_type: {context_type}
```

**Resolver – emits a chain-of-thought that pins the referent of the final instruction**

**SYSTEM**

```
You are a referring-expression resolver for a household robot. Given a possibly
vague instruction and prior dialogue, produce a SHORT chain-of-thought (2-5
sentences) that:

  1. Identifies the referring expression in the instruction (pronoun, definite
     description, attributive phrase, or none if explicit).
  2. Names the canonical object class it refers to (must match one of
     scene_objects).
  3. Cites the dialogue evidence that pins the reference (quote a short phrase
     OR say "instruction is explicit, no resolution needed").
  4. If the instruction implies a destination/receptacle, name it too.

Be concise. No bullet points, no headers -- flowing prose.
Output STRICT JSON only:

  {"reasoning": "<2-5 sentence CoT>", "target": "<CanonicalObjectName>",
   "destination": "<CanonicalReceptacleName or empty string>"}
```

**USER**

```
Scene objects: {scene_objects}

Dialogue:
{context_memory}

Instruction: {instruction}
```

**Plan-Writer – compiles the resolved (target, destination, task) into an executable plan**

**SYSTEM**

```
You write executable plans for a household robot. You are given the resolved
target/destination plus the original instruction and the task type. Output a
plan as a list of action strings, one per step.

Allowed verbs (and only these):
  find, pick_up, put_down, open, close, toggle_on, toggle_off,
  slice, clean, heat, cool, examine, done.

The final action MUST be exactly: done

Action format:
  "find <object>"
  "pick_up <object>"
  "put_down <object> <receptacle>"
  "open <object>"  /  "close <object>"
  "toggle_on <object>"  /  "toggle_off <object>"
  "slice <object>"  /  "clean <object>"  /  "heat <object>"
  "cool <object>"   /  "examine <object>"
  "done"

Object/receptacle names must be lowercase and come from the provided
scene_objects list.

Reference plan templates by task_type:
  pick_and_place  : find {obj}; pick_up {obj}; find {recep};
                    put_down {obj} {recep}; done
  stack_and_place : find {a}; pick_up {a}; find {b}; put_down {a} {b};
                    find {recep}; pick_up {b}; put_down {b} {recep}; done
  clean_and_place : find {obj}; pick_up {obj}; find sink; clean {obj};
                    find {recep}; put_down {obj} {recep}; done
  heat_and_place  : find {obj}; pick_up {obj}; find microwave; open microwave;
                    heat {obj}; close microwave; find {recep};
                    put_down {obj} {recep}; done
  cool_and_place  : find {obj}; pick_up {obj}; find fridge; open fridge;
                    cool {obj}; close fridge; find {recep};
                    put_down {obj} {recep}; done
  examine_in_light: find {obj}; pick_up {obj}; find {lamp};
                    toggle_on {lamp}; examine {obj}; done


Output STRICT JSON only:

  {"plan": ["find apple", "pick_up apple", "...", "done"]}
```

**USER**

```
task_type:     {task_type}
target:        {target}
destination:   {destination}
scene_objects: {scene_objects}
instruction:   {instruction}
```

**Critic – accepts/rejects a (reasoning, plan) pair against the REI-Bench error taxonomy**

**SYSTEM**

```
You are a strict judge of robot plans for the Planner's plan. Given an
instruction, dialogue context, scene objects, the ground truth plan, and the
proposed plan, decide whether the plan correctly executes the instruction.
Use the following error taxonomy:

  pronoun_failure     -- pronoun in instruction was resolved to the wrong object. <few shot examples>
  attributive_misid   -- attributive description (e.g. "the round fruit")
                         resolved wrong.  <few shot examples>
  noise_distraction   -- plan acted on a distractor mentioned in noised context. <few shot examples>
  context_loss        -- plan ignored required info from prior dialogue. <few shot examples>
  action_ordering     -- actions are correct but in the wrong sequence. <few shot examples>
  action_omission     -- a required action is missing. <few shot examples>
  hallucinated_action -- plan contains an action/object not warranted by
                         instruction or scene. <few shot examples>
  wrong_destination   -- final placement uses wrong receptacle. <few shot examples>

If the plan is correct on ALL axes, return error="none".

Output STRICT JSON only:

  {"verdict": "accept" | "reject",
   "error": "<one of the above or 'none'>",
   "diagnostic": "<one short sentence>"}
```

**USER**

```
Scene objects: {scene_objects}

Dialogue:
{context_memory}

Instruction: {instruction}

GT plan:
{from environment of AI2THOR}

Proposed plan:
{plan}
```

## B. Qwen Result on CLARA

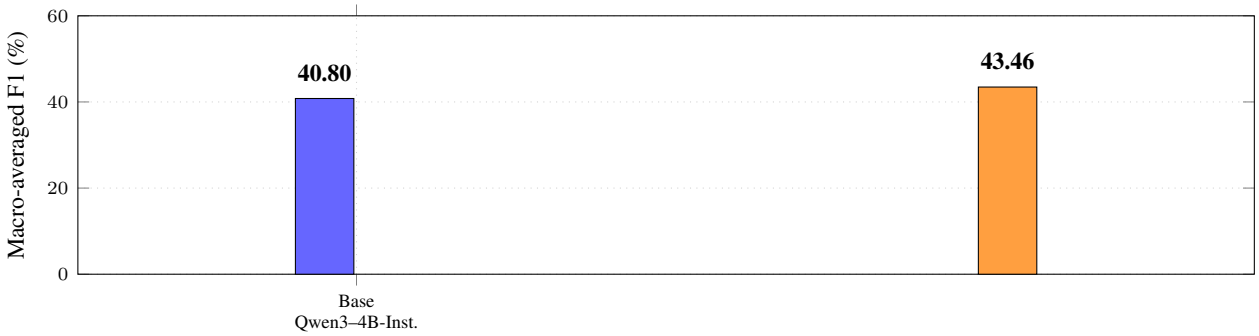


*Figure 9.* CLARA out-of-domain evaluation for Qwen3–4B-Inst.. Fine-tuning improves macro-averaged F1 from $40.80\%$ to $43.46\%$, corresponding to a $+2.66$ absolute gain.

On CLARA, fine-tuning also improves the stronger Qwen3–4B-Inst. backbone, raising macro-averaged F1 from $40.80\%$ to $43.46\%$ (Fig. 9, $+2.66$ absolute). This indicates that the REI-Bench supervision transfers beyond the Llama backbone and provides consistent gains even for a higher-performing base model.

## C. Qwen Ablations

For completeness, we report the corresponding Qwen3-4B-Instruct-2507 supervision-scaling ablation in the (Fig. 10). Unlike Llama-3.2-3B-Inst., the larger Qwen3-4B-Inst. backbone is already near saturation at $n$=50 samples per category, achieving $95.66\%$ mean SR across the nine REI-Bench cells. Quadrupling the budget to $n$=200 yields only a further $0.71$ absolute point gain, reaching $96.37\%$ mean SR. The explicit conditions remain fixed at $98.1\%$ for every value of $n$, and the per-cell curves largely overlap in the 93–98% band. This suggests that Qwen3-4B-Inst. has sufficient prior knowledge to interpolate from as few as $50$ samples per category; any budget in $[50, 200]$ recovers more than $99\%$ of the maximum attainable in-distribution performance for Qwen-class backbones.

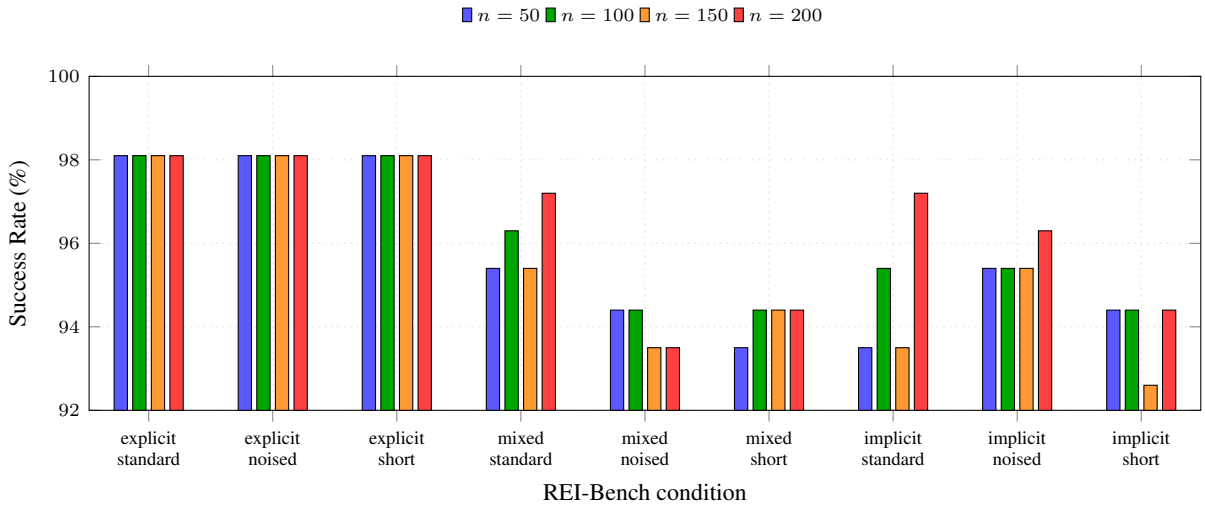


*Figure 10.* Qwen3–4B-Instruct-2507 SFT scaling on REI-Bench across all nine ambiguity conditions.

We also report the corresponding Qwen3-4B-Instruct-2507 class-coverage ablation in the (Fig. 11). In contrast to Llama-3.2-3B-Inst., Qwen3-4B-Inst. captures almost the entire gain in the first jump: $76.96 \rightarrow 95.67\%$ mean SR when moving from $1$ to $4$ classes ($+18.71$), followed by only a $+0.70$ gain from $4$ to $9$ classes. The $1$-class variant even generalizes partially across the taxonomy, reaching $86.1\%$ on `explicit_short` despite never seeing an explicit training example. These results suggest that the larger Qwen backbone needs as few as four ambiguity classes at $200$ samples per class to recover nearly all in-distribution SFT gains, after which returns diminish sharply.

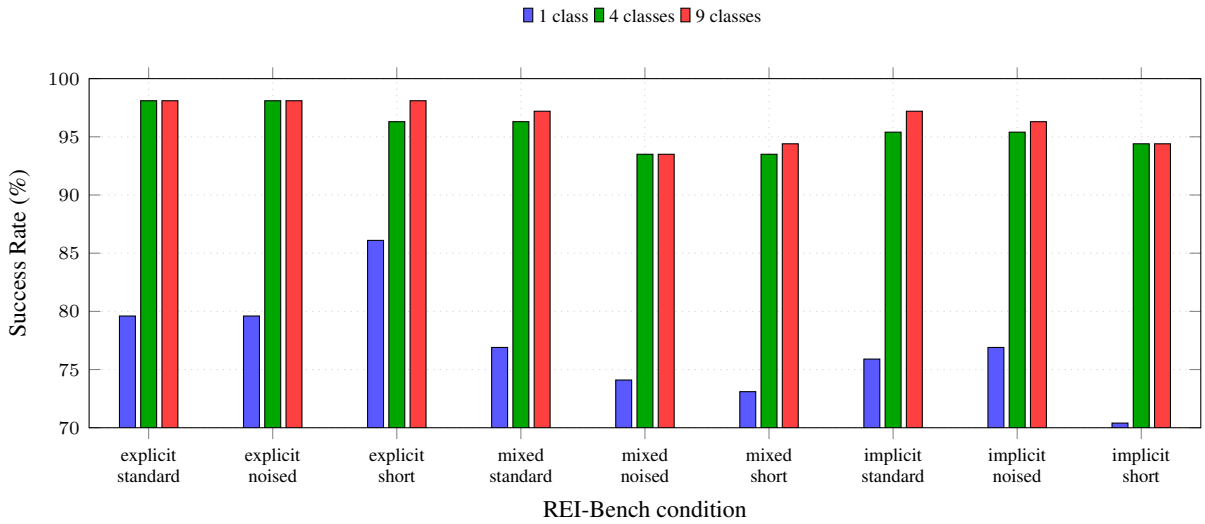


*Figure 11.* Finetuned Qwen3–4B-Instruct-2507 class-coverage ablation on REI-Bench across all nine ambiguity conditions. Each selected class is trained with 200 examples.

## D. Error Categorization Details

We have found 8 categories of errors from running REI-Bench on Llama, Qwen models. The details are as follows:

**1 Action Omission** One or more **critical steps** required by the ground-truth plan are entirely absent from the agent's output. The agent understands the object and destination but treats intermediate manipulation steps (e.g., heating, cleaning) as implicit background assumptions rather than explicit actions to emit. The plan is structurally short, often completing directly from `pick_up` to `put_down`, skipping mandatory sub-routines.

> **Instruction:** "*Heat and serve the potato.*"
>
> **Gold Plan:** `find(Potato)` → `pick_up(Potato)` → `find(Microwave)` → `open(Microwave)` → `heat(Potato, Microwave)` → `close(Microwave)` → `find(DiningTable)` → `put_down(Potato, DiningTable)` → `done`
>
> **Agent Output:** `find(Potato)` → `pick_up(Potato)` → `find(DiningTable)` → `put_down(Potato, DiningTable)` → `done`
>
> *Agent picks and places the potato but omits the entire `heat` sub-routine.*

**2 Action Ordering**

All required actions are **present** in the plan but are executed in the **wrong logical sequence**. The agent fails to model causal dependencies between actions (e.g., an object must be found before it can be picked up).Often driven by surface mention-order in the instruction: the agent plans actions in the order entities are named, not the order they must be performed.

> **Instruction:** "*Locate and pick up the apple.*"
>
> **Gold Plan:** `find(Apple)` → `pick_up(Apple)` → `done`
>
> **Agent Output:** `pick_up(Apple)` → `find(Apple)` → `done`
>
> *Agent attempts `pick_up` before `find`, violating the prerequisite dependency.*

**3 Hallucinated Action**

The agent generates action verbs that are **not in the REI-Bench skill vocabulary**, or invokes objects that do not exist in the current scene. This arises when the model defaults to world-knowledge priors from pre-training rather than constraining generation to the closed, permitted action set. This error is more frequent at smaller model scales, where the skill vocabulary has not been sufficiently memorised during fine-tuning.

> **Instruction:** "*Use available skills to prepare the food.*"
>
> **Valid skills include:** `heat`, `pick_up`, `find`, `put_down`, `open`, `close`
>
> **Agent Output:** `find(Apple)` → `pick_up(Apple)` → **`cook(Apple)`** → `done`
>
> *`cook` is not a valid REI-Bench skill; the agent fabricates it from culinary world knowledge instead of using `heat`.*

**4 Attributive Misidentification**

The agent selects the **wrong object** due to incorrect interpretation of a descriptor such as **color, size, or state** (e.g., clean vs. dirty). The descriptor is parsed but not used as a hard filter; instead, the model resolves to the most scene-frequent or most recently encoded object sharing the same head noun. The resulting plan is structurally valid but operates on the wrong physical entity throughout.

> **Instruction:** "*Pick up the red fruit.*"
>
> **Scene objects:** RedApple, GreenApple, Bowl, CounterTop

**Gold Plan:** `find(RedApple)` → `pick_up(RedApple)` → `done`

**Agent Output:** `find(`**`GreenApple`**`)` → `pick_up(`**`GreenApple`**`)` → `done`

*Agent picks the green apple despite the explicit colour attribute in the instruction.*

### 5 Pronoun Failure

The agent fails to **resolve an anaphoric reference** ("it", "them", "there") to the correct entity established earlier in the instruction or dialogue context. Results in the entire plan being executed on or directed toward the wrong object or location. Common when the referent is introduced in a prior dialogue turn, far from the action-generating step in the token sequence.

**Instruction:** "*Pick up the apple. Heat it.*"

**Gold Plan:** `find(Apple)` → `pick_up(Apple)` → `find(Microwave)` → `open(Microwave)` → `heat(`**`Apple`**`, Microwave)` → `close(Microwave)` → `done`

**Agent Output:** `find(Apple)` → `pick_up(Apple)` → `find(Microwave)` → `open(Microwave)` → `heat(`**`Potato`**`, Microwave)` → `close(Microwave)` → `done`

*Agent failed to associate "it" with the apple and resolved the pronoun to a distractor object in the scene.*

### 6 Noise Distraction

Ambiguous object names or **distractor objects** whose names overlap with the target cause the agent to select the wrong entity. The agent matches on the lexical head noun (e.g., "book") but does not resolve partial-name distractors (e.g., `BookShelf`) or irrelevant co-present objects. Typically observable when the scene contains objects whose names are substrings or superstrings of the target name.

**Instruction:** "*Find the book.*"

**Scene objects:** Book, BookShelf, Newspaper, Desk

**Gold Plan:** `find(`**`Book`**`)` → `done`

**Agent Output:** `find(`**`BookShelf`**`)` → `done`

*Agent targets `BookShelf` because "book" appears as a prefix in its name, failing to distinguish the exact target from a containing object.*

### 7 Context Loss

The agent **fails to maintain task context** across multiple steps in a long-horizon plan, leading to hallucinated targets or arbitrary object substitution mid-execution. The agent begins the task correctly but forgets the original target partway through, switching to a random or scene-salient object. Particularly prevalent under implicit instructions where the target is established in a prior dialogue turn rather than the immediate instruction.

**Context memory:** "*User: I left the tomato on the counter. It needs to be chilled.*"

**Instruction:** "*Can you take care of it?*"

**Gold Plan:** `find(`**`Tomato`**`)` → `pick_up(`**`Tomato`**`)` → `find(Fridge)` → `open(Fridge)` → `cool(`**`Tomato`**`, Fridge)` → ···

**Agent Output:** `find(`**`Apple`**`)` → `pick_up(`**`Apple`**`)` → `find(Fridge)` → `cool(`**`Apple`**`, Fridge)` → ···

*Agent ignores the dialogue-established target and substitutes `Apple` — the first coolable object encountered in the scene list.*

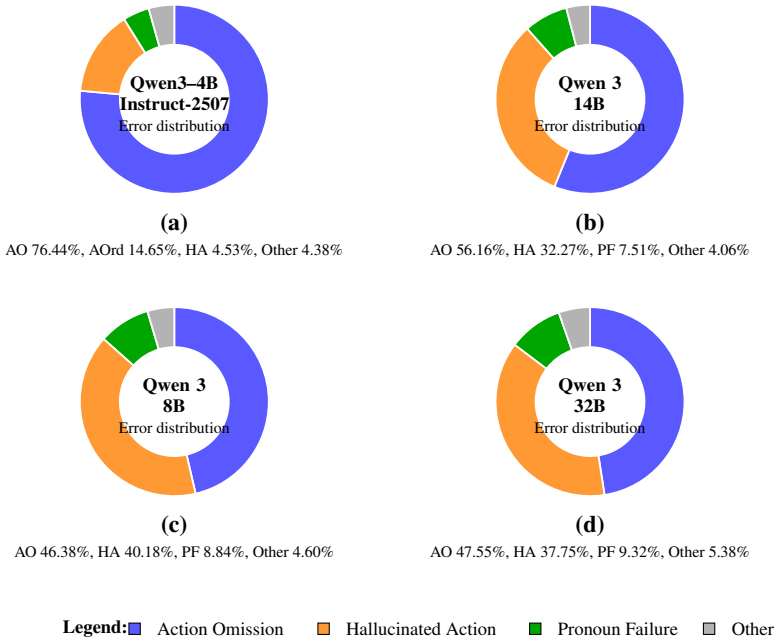


*Figure 12.* Critic-derived error distributions across Qwen-family models. For readability, minor categories are grouped into *Other*. AO denotes action omission, AOrd denotes action ordering, HA denotes hallucinated action, and PF denotes pronoun failure.

### 8 Wrong Destination

The target object is **correctly identified and manipulated** throughout the plan, but is placed at the **wrong final receptacle**. The agent resolves the destination using stereotypical spatial priors from pre-training (e.g., food goes on the counter) rather than the explicit or contextually specified location. Plans are otherwise complete and well-ordered; only the `put_down` argument is wrong.

**Instruction:** "*Put it in the fridge.*"

**Gold Plan:** $\cdots \rightarrow$ `find(`**`Fridge`**`)` $\rightarrow$ `open(Fridge)` $\rightarrow$ `put_down(Apple, `**`Fridge`**`)` $\rightarrow$ `close(Fridge)` $\rightarrow$ `done`

**Agent Output:** $\cdots \rightarrow$ `find(`**`CounterTop`**`)` $\rightarrow$ `put_down(Apple, `**`CounterTop`**`)` $\rightarrow$ `done`

*Agent places the object on the counter despite the instruction explicitly specifying the fridge as the destination.*

Figure 12 shows the critic-derived error distribution across Qwen-family models. Action omission is the dominant failure mode across all variants, accounting for 76.44% of errors for Qwen3-4B-Instruct-2507 and remaining the largest category for Qwen 3 14B (56.16%), Qwen 3 8B (46.38%), and Qwen 3 32B (47.55%). As model size increases, the share of hallucinated actions becomes more prominent, reaching 37.75% for Qwen 3 32B, while pronoun-related failures also remain non-trivial across the stronger models.

## E. Cost and Token Analysis

Figure 13 reveals a clear cost-quality tradeoff across the four prompting baselines that further motivates the SFT approach. ICL incurs the highest prompt token consumption for both models ($\sim$839–840k tokens), owing to the inclusion of in-context exemplars, yet achieves the best inference throughput at 2.3 examples/s for Llama-3.2-1B-Inst. — a consequence of its relatively short completions (35k tokens). CoT and ToCC present the opposite profile: both generate substantially longer completions (282k and 151k tokens for 1B; 110k and 39k for 3B respectively) and suffer the worst evaluation times, with CoT requiring up to 3,840s and ToCC up to 2,646s across the two models, reducing throughput to as low as 0.2–0.4 examples/s. AP is the most lightweight baseline in terms of completion length and wall-clock time, yet as shown in Figure 13, it provides the least reliable performance gain. Crucially, none of these baselines escape the fundamental cost-scaling problem of agentic deployment: even the cheapest configuration (ICL on 3B at 1.5 examples/s) accumulates

prohibitive token budgets over hundreds of planning steps per episode. By contrast, once trained, our SFT models require no additional prompt augmentation, operate at native inference speed with minimal completion tokens, and eliminate per-query API overhead entirely — making FAST-PLAN not only more accurate but substantially more economical for real-world embodied deployment.

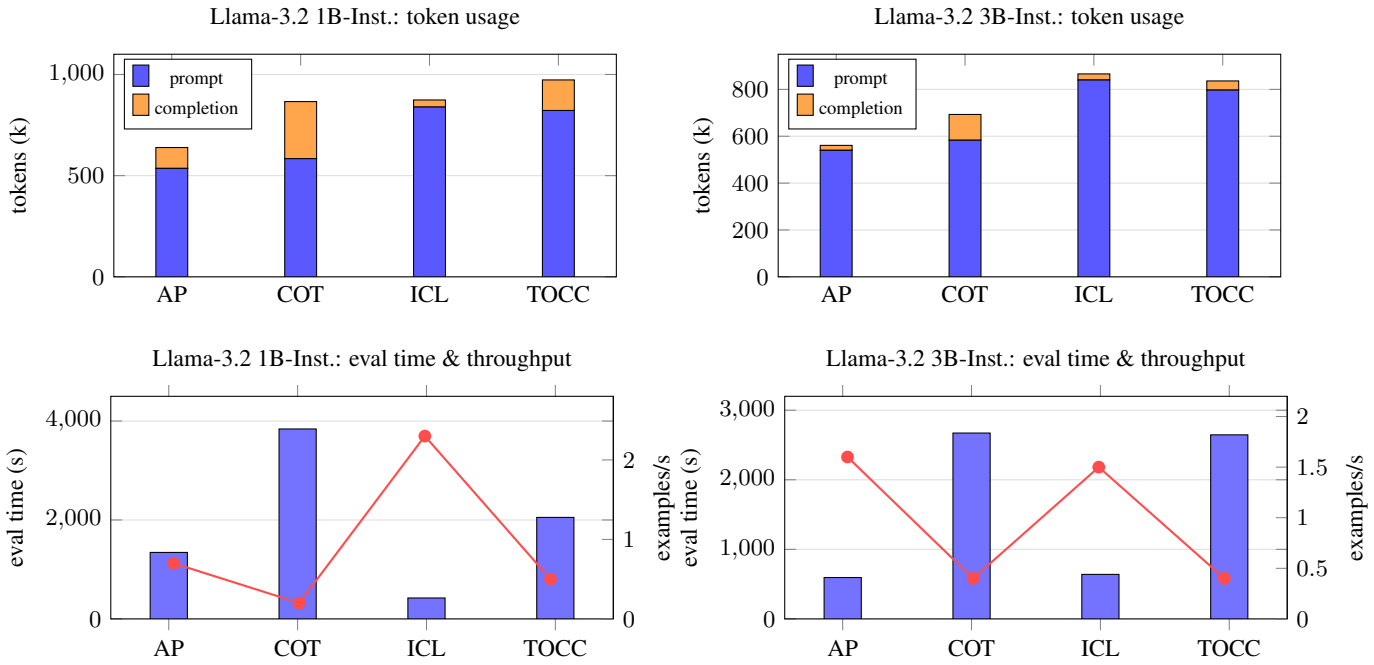


*Figure 13.* Cost ablation of prompting baselines for Llama-3.2-1B-Inst. and Llama-3.2 3B-Inst. The top row shows prompt and completion token usage, while the bottom row shows evaluation time and throughput for AP, COT, ICL, and TOCC.