Detecting Ambiguity in Embodied Instructions via Semantic Valency Conflict

Anonymous ACL submission

Abstract

Natural language instructions in human-robot interaction often contain subtle ambiguities that hinder reliable interpretation. These ambiguities arise when a single instruction can be interpreted in multiple ways, assigning conflicting semantic roles to objects, tools, or participants, potentially leading to execution failures. To address this, we propose Semantic Valency Conflict (SVC), a cognitively inspired, logit-free method for detecting ambiguity in instructions directed to robot. SVC identifies divergences in role assignments across alternative interpretations of a predicate, using large language models (LLMs) to infer context-sensitive semantic frames. Our method is model-agnostic and compatible with both open- and closed-source LLMs. SVC produces clear, structured outputs that highlight which parts of the instruction are ambiguous and indicate which predicate and its associated arguments lead to multiple or conflicting interpretations. We evaluate SVC on two datasets, AmbiK and IntroPlan Mobile Manipulation, and demonstrate that it shows strong and consistent performance in detecting subtle ambiguities in natural language instructions given to robots in safety, unambiguous, and preference-based scenarios. The source code will be made publicly available.

1 Introduction

Natural language is inherently polysemous, making it challenging for large language models (LLMs) to accurately follow instructions (Heo et al., 2025), interpret textual descriptions (Singhal et al., 2024; Chuganskaya et al., 2023), and perform planning tasks (Hazra et al., 2024; Grigorev et al., 2024, 2025). Depending on the context, multiple valid actions may exist in a given situation, making it difficult to select the most appropriate one. These challenges are associated with two types of uncertainty: epistemic, arising from lack of knowledge or insufficient data, and aleatoric, caused by random and unpredictable variations in the environment (Shorinwa et al., 2024). Both types of uncertainty complicate action selection by introducing ambiguity in outcome evaluation and adaptation to new situations.

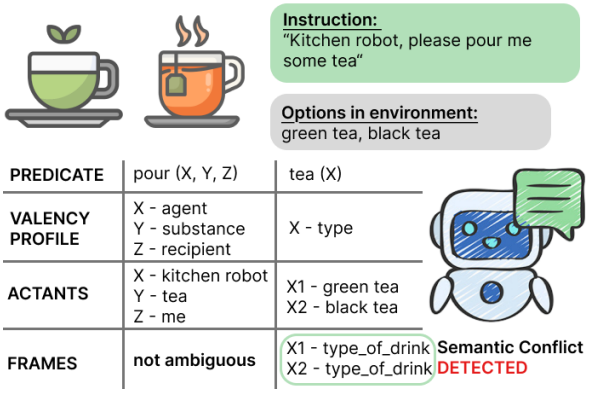

Figure 1: The robot receives the instruction "Pour me some tea" in environment with green and black tea. The method detects ambiguity by identifying competing interpretations of the predicate "tea", where "green" and "black" act as conflicting actants within the same action frame, revealing a preference-related variability.

In robotics, one of the key challenges is following instructions given in natural language. Modern approaches to solving this problem primarily rely on LLMs, which, unlike heuristic methods, can flexibly interpret complex commands (Huang et al., 2022; Ahn et al., 2022; Kovalev and Panov, 2022; Sarkisyan et al., 2023; Onishchenko et al., 2025). However, user instructions can often be ambiguous due to factors such as the use of synonyms, metaphorical expressions, abbreviations, or environmental complexity where the same object may be represented in various forms (Fig. 1). This ambiguity, alongside known risks like LLM hallucinations, can lead to mission failures or safety hazards (Zhang et al., 2025). To mitigate these risks, it is essential to detect ambiguity promptly and clarify instructions. Implementing ambiguity

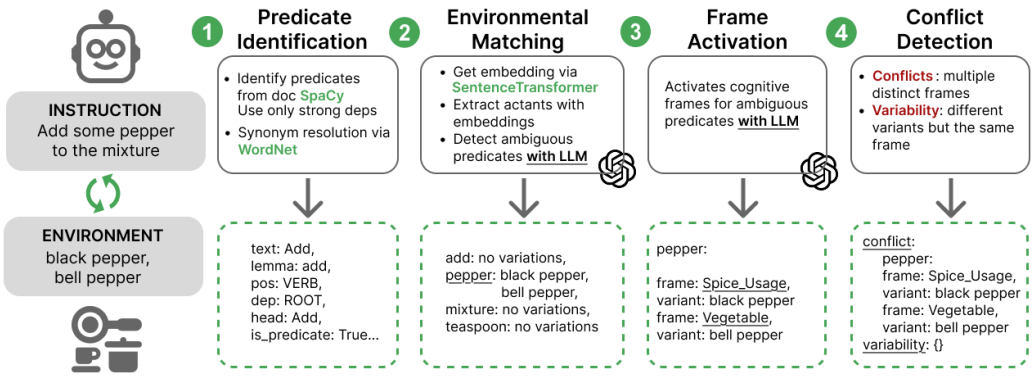

Figure 2: Pipeline for detecting semantic ambiguity. SVC decomposes the detection process into four main stages, each corresponding to a step in the cognitive mechanism of frame activation and conflict evaluation. The outputs of each stage is illustrated using the example instruction: "Add some pepper to the mixture" and environment: black pepper, bell pepper. The system identifies potential ambiguity by tracing how the lexical unit **pepper** may evoke distinct semantic interpretations depending on context.

detection plays a vital role in this process, ensuring the safe integration of LLMs into intelligent agents (Firoozi et al., 2023).

Common techniques for ambiguity detection incorporate external knowledge bases to improve accuracy, but this reduces the system's autonomy and adaptability in novel environments (Liang et al., 2024). Although recent methods apply techniques like Conformal Prediction (CP) such as Su et al. (2024a) and affordance estimation Jr. and Manocha (2024) to improve ambiguity detection, they still struggle to effectively capture subtle ambiguities inherent in natural language instructions.

To address ambiguity in natural language instructions caused by lexical polysemy and underspecified roles, we propose the **Semantic Valency Conflict** (SVC) method, an interpretable and logit-free method that produces structured explanations of ambiguity sources while providing a useful semantic signal for ambiguity detection across multiple benchmark settings. The key idea is that a single word (lexeme) can trigger multiple, incompatible interpretations known as **semantic frames**, each expecting a different set of roles and participants to make sense in context (also called a valency profile, see Fig. 1). Following Yarowsky (1993), who observed that ambiguity often arises when a predicate simultaneously activates incompatible valency structures, SVC models ambiguity as the presence of conflicting semantic valency profiles under a given environment (i.e., the set of objects, properties, and relations present in the external context). An instruction is considered ambiguous when it supports multiple mutually exclusive interpretations. The SVC method identifies lexemes whose valency profiles are incompatible within the current environment through predicate identification, environment matching and frame activation.

We evaluated SVC method ability to detect ambiguity on two fully textual datasets: AmbiK (Ivanova et al., 2025) and IntroPlan Mobile Manipulation (Liang et al., 2024). Compared to logit-based existing conformal-prediction, affordance-based, and prompting baselines, which fail to detect ambiguity, SVC shows improved performance in identifying semantically ambiguous instructions, particularly in user-preference contexts and safety tasks.

Statement of contributions. Our contributions are threefold: (1) we propose Semantic Valency Conflict (SVC), a frame-based method for detecting semantic ambiguity in natural language instructions through valency conflict analysis; (2) we introduce an interpretable and logit-free ambiguity detection method compatible with both white-box and black-box language models, without relying on static knowledge bases or predefined plans; and (3) we evaluate SVC on the AmbiK and IntroPlan benchmarks across multiple language models and ambiguity categories, showing strong performance on semantically ambiguous instruction types, particularly in preference-, safety-, spatial-, and multilabel-related tasks.

2 Related Works

LLMs exhibit uncertainty due to both external factors (e.g., data noise, ambiguity, limited context) and internal factors (e.g., model architecture and probabilistic generation) (Zhang and Choi, 2025; Wang et al., 2025). While external uncertainty can sometimes be mitigated via clarifying questions, internal uncertainty is typically addressed through

model improvements or richer in-context examples.

A range of approaches has been proposed for ambiguity detection, including token-based, self-verbalized, semantic-similarity, and mechanistic interpretability methods (Shorinwa et al., 2024). Techniques such as directed entailment graphs further improve transparency of model reasoning (Da et al., 2024). However, detecting uncertainty does not by itself resolve it.

Common mitigation strategies include generating clarifying questions (Zhang and Choi, 2025), aggregating predictions via model ensembles (Liu et al., 2024), and leveraging embeddings from smaller models to distinguish uncertainty types (Ahdritz et al., 2024). Conformal prediction (CP) has also been applied to calibrate uncertainty in LLM-based planning systems (Angelopoulos and Bates, 2022).

In robotic instruction-following, several LLM-based planning approaches explicitly address uncertainty. KnowNo Ren et al. (2023) enables planners to recognize and align uncertainty using CP. IntroPlan (Liang et al., 2024) extends this with retrieval-augmented planning and proactive confidence estimation. LAP (Jr. and Manocha, 2024) introduces the A-Feasibility metric to assess action feasibility and safety given scene context. Other work focuses on resolving ambiguity arising from underspecified instructions: Dogan et al. (2025) use iterative clarification grounded in knowledge embeddings, while Jiang et al. (2025) highlight the role of contextual and environmental factors in shaping referring expressions.

3 Background

In this section, we introduce key concepts used throughout the paper, **illustrative examples** are provided in Appendix A. Research in cognitive linguistics has shown that word meanings are shaped by underlying conceptual structures (Schwarze and Schepping, 1995; Bierwisch, 1983; Lehrer, 1990). Following frame semantics (Fillmore, 1985), we assume that understanding a lexeme requires reference to a **frame** specifying its semantic roles. For example, the instruction "Pick up the glasses" may refer either to drinkware or to eyeglasses depending on the active semantic frame. Central to this view is the **lexical core** – the invariant meaning around which the lexeme's interpretations are structured.

We define the **semantic valency** of a lexeme L as the set of independent variables X necessary for interpreting L's core meaning. Lexemes that require one or more such variables are **predicates**, while the expressions filling these semantic roles in an instruction are called **semantic actants** (Testelets, 2001). Actants instantiate the predicate's roles in specific situations, linking meaning with linguistic realization. For instance, in the instruction "Pour me some tea", the predicate tea requires specification of a type (e.g., green or black). In a context where multiple instantiations of this actant are possible, such as green tea and black tea co-occurring in the environment, the valency slot remains underdetermined. Our method identifies this as a semantic variability: multiple actants (types of tea) compete within the same frame method, revealing ambiguity rooted in underspecified preference (see example in Fig. 1.

4 Method

Problem Formulation. Formally, let L be a lexeme (it is important to note that in our approach the term predicate is used in a broad sense, encompassing not only verbs but also nouns or even entire phrases that activate distinct semantic frames), and let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of semantic valencies of L, where each x_i corresponds to an argument slot (actant) required for coherent interpretation. We define a **Semantic Valency Conflict** (SVC) as a case where multiple interpretations of L are simultaneously activated, and their corresponding valency profiles are incompatible with respect to the environment $\mathcal{E}$. The environment $\mathcal{E}$ is the set of objects, properties, and relations present in the external context where the interpretation occurs. It provides the grounding necessary to resolve or exacerbate potential conflicts between valency profiles.

Each interpretation I_k of L activates a semantic frame $\mathcal{F}_k$, which implicitly defines a *valency profile* $\mathcal{V}_k \subset \mathcal{D}$ – a set of expected participant roles and their semantic types. The domain $\mathcal{D}$ represents the space of all conceptually valid participant configurations for predicates in natural language.

While a frame represents a high-level conceptual scenario, the corresponding valency profile captures its operational semantic structure, i.e., the expected participant roles and their constraints. Multiple frames may partially overlap conceptually while still differing in their valency realization. Therefore, ambiguity detection is performed at the level of valency profiles rather than frame labels

themselves. For example, two interpretations may evoke related cooking scenarios, but require incompatible participant structures (e.g., spice vs. vegetable usage), which becomes observable only at the valency level.

A central problem in SVC is determining whether multiple valency profiles associated with a lexeme can be simultaneously grounded within the same environment $\mathcal{E}$. To capture this, we define the contextual compatibility relation $\sim_{\mathcal{E}}$, where $\mathcal{V}k \sim \mathcal{E}\mathcal{V}_m$ indicates that their semantic roles can be matched to overlapping or semantically compatible entities in the environment. Operationally, this involves comparing vector representations of candidate actants, modifiers, and entity types against objects and relations present in $\mathcal{E}$. If there exists $k \neq m$ such that $\mathcal{V}k \not\sim \mathcal{E}\mathcal{V}_m$, the instruction is considered ambiguous.For a detailed illustration of the pipeline, see Fig. 2.

Overview of the SVC Method. To address the problem of detecting ambiguity arising from the lexical polysemy and underspecified roles, we propose the **Semantic Valency Conflict** method. It is based on the assumption that ambiguity in a natural language instruction is determined by a conflict between different semantic valencies of the same lexeme (Long et al., 2022; Apresjan, 2000). In other words, ambiguity arises when a single lexeme simultaneously activates multiple incompatible frames $\mathcal{F}$.

We propose an architecture (see Fig. 2) in which each stage addresses a specific component of semantic ambiguity detection. The overall pipeline mirrors the process of frame activation and conflict detection and is summarized in Algorithm 1.

At the first stage (lines 1–2 of Algorithm 1), the instruction is parsed to identify predicates $\mathcal{P}$ and their semantic roles $\mathcal{A}$. Predicates correspond to lexemes whose interpretation depends on one or more semantic actants.

At the second stage (lines 3–6), the extracted semantic roles are aligned with the environment $\mathcal{E}$, i.e., the set of objects, properties, and relations available in the current context. For each role, the system retrieves candidate entities from $\mathcal{E}$ based on semantic similarity. If multiple semantically compatible candidates exist for the same role, the role is treated as potentially ambiguous.

At the third stage (lines 7–10), the system generates candidate interpretations for ambiguous predicates and dynamically activates the corresponding semantic frames $\mathcal{F}_k$ using LLM prompting. Each activated frame defines a valency profile $\mathcal{V}_k$ over the semantic roles and constraints associated with the interpretation.

At the final stage (lines 11–18), the system compares the activated valency profiles. If there exist two interpretations $k \neq m$ such that

$$\mathcal{V}_k \not\sim_{\mathcal{E}} \mathcal{V}_m, \tag{1}$$

the predicate is classified as ambiguous due to **frame conflict**. This indicates that the instruction simultaneously supports multiple mutually incompatible scenario.

Alternatively, if only one frame $\mathcal{F}_k$ is activated but multiple variants $\{v_i\}$ remain valid within this frame,

$$|\{v_i\}| > 1, \tag{2}$$

the ambiguity is classified as **semantic variability** rather than frame conflict. In this case, the instruction admits multiple realizations within the same semantic scenario (e.g., different oil types within the frame `Cooking_Oil_Ingredient`). Although such variability does not indicate incompatible frames, it still requires disambiguation for accurate grounding.

5 Experimental Evaluation

To systematically assess the effectiveness of SVC, we conduct a comprehensive experimental evaluation structured around the following research questions (RQs):

RQ1: Can semantic conflict serve as a useful signal for ambiguity detection?

RQ2: In which task types is semantic valency conflict most significant?

RQ3: How does method performance vary across different language models?

RQ4: How effective is direct prompting for ambiguity detection?

RQ5: Which components of SVC contribute most to ambiguity detection performance?

Datasets. To evaluate SVC, we used two datasets: **AmbiK** (Ivanova et al., 2025) and **IntroPlan Mobile Manipulation** (Liang et al., 2024). Both contain robot instructions within a given environment and were used only for evaluation.

IntroPlan Mobile Manipulation dataset includes 600 short, single-step tasks with scene descriptions and object lists with the same distribution of different types of examples as in the KnowNo

Algorithm 1: SVC Ambiguity Detection

Input: Instruction I, environment $\mathcal{E}$
Output: Ambiguity type, semantic variants $\mathcal{V}$

```
Extract predicates P and semantic roles A
  from I;
foreach semantic role a ∈ A do
    retrieve candidate entities from E using
      semantic similarity;
    if multiple compatible entities are found
      then
        mark role as potentially ambiguous;
    end
end
if ambiguous roles exist then
    generate candidate interpretations V
      using LLM prompting;
    activate semantic frames F for each
      interpretation;
end
foreach predicate p ∈ P do
    if multiple incompatible frames are
      activated then
        classify as frame conflict;
    end
    else if multiple valid variants exist
      within the same frame then
        classify as semantic variability;
    end
    else
        classify as clear;
    end
end
return ambiguity type and semantic
  variants V;
```

dataset (Ren et al., 2023). AmbiK is a specialized dataset containing 2,000 annotated instructions covering three types of semantic ambiguity: user preferences, commonsense knowledge, and safety considerations.

Metrics. We use **Contextual Help Quality (CHQ)** as the main evaluation metric. CHQ measures whether the system both correctly detects ambiguity and preserves the user's intended meaning during disambiguation. It is computed as the harmonic mean of two metrics: **Correct Help Rate (CHR)** and **Intent Alignment (IA)**.

Correct Help Rate (CHR) (Ivanova et al., 2025) evaluates whether the system makes an appropriate help-request decision given the ambiguity type annotated in the dataset. For unambiguous instructions, the correct behavior is to avoid requesting clarification, while ambiguous instructions require clarification depending on the task category.

Intent Alignment (IA) measures whether the interpretations generated by the method preserve the original user intent. IA is computed by embedding both the ground-truth intents and the generated interpretations into a shared semantic space and comparing them using cosine similarity. A prediction is considered aligned if at least one generated interpretation exceeds a predefined similarity threshold with a ground-truth intent.

Finally, CHQ combines ambiguity detection quality and intent preservation:

$$\text{CHQ} = \frac{2 \cdot \text{IA} \cdot \text{CHR}}{\text{IA} + \text{CHR} + \varepsilon}, \tag{3}$$

where $\varepsilon = 10^{-6}$ is added for numerical stability. Additional implementation details and metric definitions are provided in Appendix B.

Baselines and Models. We compare our method against **four baselines** commonly used for ambiguity detection. **KnowNo** (Ren et al., 2023) applies conformal prediction to select among multiple candidate interpretations. **LAP** (Jr. and Manocha, 2024) extends this approach by incorporating affordance-based scoring of candidate actions. **LoFreeCP** (Su et al., 2024b) performs conformal prediction without logit access, relying on variability across multiple LLM generations.

We also include a **Prompting** baseline, where the LLM directly classifies instructions as {Ambiguous, Unambiguous} and optionally generates alternative interpretations (Appendix C). This setup unifies the **Binary** and **NoHelp** variants from (Ren et al., 2023), which differ only in whether uncertainty is explicitly expressed.

We conducted experiments using **GPT-3.5-Turbo** (OpenAI, 2023)[1], **Mistral-7B-Instruct-v0.2** (MistralAI, 2023)[2], **Gemma-2-9B-IT** (Google, 2024)[3], and **Qwen1.5-7B-Chat** (QwenTeam, 2024)[4]. For brevity, we refer to these models simply as GPT-3.5, Mistral,

[1] Accessed via API: https://platform.openai.com
[2] Available at: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[3] Available at: https://huggingface.co/google/gemma-2-9b-it
[4] Available at: https://huggingface.co/Qwen/Qwen1.5-7B-Chat

Table 1: Performance on Ambik dataset. The best result is shown in **bold**, while the second-best is underlined.

		Unambiguous			Common Sense			Preferences			Safety		
Model	Method	CHR	IA	CHQ	CHR	IA	CHQ	CHR	IA	CHQ	CHR	IA	CHQ
GPT-3.5	LoFreeCP	0.77	0.18	0.29	0.80	0.10	0.17	0.15	0.10	0.12	0.24	0.10	0.14
	LAP	**1.00**	0.41	**0.58**	**1.00**	0.20	0.33	0.00	0.10	0.00	0.00	0.20	0.00
	KnowNo	**1.00**	0.35	0.51	**1.00**	0.21	**0.34**	0.00	0.10	0.00	0.00	0.22	0.00
	Prompting	0.17	**0.45**	0.25	0.21	**0.32**	0.25	0.82	**0.60**	**0.69**	0.25	0.23	0.23
	SVC	0.51	0.35	0.41	0.51	0.26	**0.34**	**0.77**	0.58	0.66	**0.43**	**0.26**	**0.32**
Mistral	LoFreeCP	0.28	**0.69**	**0.39**	0.31	0.40	**0.34**	0.73	0.52	0.60	**0.62**	**0.50**	**0.55**
	LAP	0.93	0.12	0.21	0.95	0.05	0.09	0.06	0.04	0.04	0.04	0.02	0.02
	KnowNo	**1.00**	0.02	0.03	**1.00**	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
	Prompting	**1.00**	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	SVC	0.51	0.32	**0.39**	0.23	**0.43**	0.30	**0.77**	**0.74**	**0.75**	0.44	0.45	0.44
Qwen	LoFreeCP	0.36	**0.24**	0.28	0.26	**0.28**	0.27	0.62	0.32	0.42	0.17	0.23	0.19
	LAP	0.73	0.22	**0.33**	0.64	0.21	0.31	0.28	0.17	0.21	0.29	0.17	0.21
	KnowNo	**1.00**	0.08	0.14	**1.00**	0.02	0.03	0.00	0.19	0.00	0.00	0.00	0.00
	Prompting	**1.00**	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	SVC	0.54	0.17	0.25	0.61	0.27	**0.37**	**0.71**	**0.37**	**0.48**	**0.39**	**0.25**	**0.30**
Gemma	LoFreeCP	0.51	**0.41**	**0.45**	0.44	**0.20**	0.27	0.20	0.24	0.21	**0.34**	0.09	0.14
	LAP	0.66	0.12	0.20	0.89	0.04	0.07	0.00	0.10	0.00	0.13	0.05	0.07
	KnowNo	**1.00**	0.09	0.16	**1.00**	0.00	0.00	0.00	0.17	0.00	0.00	0.00	0.00
	Prompting	**1.00**	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	SVC	0.83	0.26	0.39	0.84	0.17	**0.28**	**0.37**	**0.54**	**0.43**	0.21	**0.32**	**0.25**

Gemma, and Qwen, respectively. To compare the proposed SVC-based approach with a prompt-only baseline, we additionally employed the recently released **GPT-5-mini** model (OpenAI, 2025)[5]. For implementation details and prompts used see Appendix C.

Results. Our experiments show that SVC provides an interpretable and semantically grounded approach to ambiguity detection in embodied instructions.

RQ1: Semantic conflict is a useful signal for ambiguity detection. Results on AmbiK (Table 1) show that SVC performs particularly well on instruction categories where ambiguity arises from underspecified semantic roles or competing interpretations. In the *Preferences* category, SVC achieves the highest CHQ score across all evaluated backbones except GPT-3.5, where Prompting slightly outperforms it (0.69 vs. 0.66). For Mistral, SVC improves CHQ from 0.60 (LoFreeCP) to 0.75, and for Qwen from 0.42 (LoFreeCP) to 0.48. SVC also achieves the best Safety CHQ scores for GPT-3.5 (0.32), Qwen (0.30), and Gemma (0.25), while remaining competitive on Mistral (0.44 vs. 0.55 for LoFreeCP). On IntroPlan (Table 2), SVC achieves the highest CHQ scores in several ambiguity-sensitive categories, particularly *unsafe*, *spatial*, *multilabel*, and *creative multilabel* instructions. For example, on GPT-3.5, SVC improves CHQ on *unsafe* instructions from 0.17 (Prompting) and 0.11 (LAP) to 0.61, and on *multilabel* instructions from 0.48 (LAP) to 0.57. Similar trends are observed for Qwen and Gemma, where SVC achieves the strongest performance on unsafe and spatial categories. These results suggest that semantic frame conflict provides a useful signal for detecting ambiguity in semantically underspecified instructions.

RQ2: Semantic valency conflict is most significant in underspecified instruction types. SVC is particularly effective on instruction categories where ambiguity arises from underspecified semantic roles and competing realizations within the environment. This effect is most visible in safety-related, spatial, and multilabel tasks on IntroPlan (Table 2), where the method consistently improves both ambiguity detection and intent preservation.

For example, on the *unsafe* category, SVC achieves the highest CHQ scores across all evaluated backbones: 0.61 on GPT-3.5 and Mistral, and 0.68 on Qwen and Gemma, substantially outperforming Prompting (0.00–0.17) and LAP (0.11–0.15). Similar improvements are observed for *spatial* instructions, where SVC reaches CHQ scores of 0.41 (GPT-3.5), 0.23 (Mistral), and 0.26 (Qwen and Gemma), while most baselines collapse to near-zero performance. SVC also performs strongly on unambiguous instructions, achieving CHQ scores of 0.97 on GPT-3.5 and 0.99 on Mistral, indicating reliable rejection of false ambiguity. In contrast, uncertainty-driven baselines such as LoFreeCP frequently fail in these settings, often producing CHQ = 0 due to excessive ambiguity triggering. These results suggest that semantic valency conflict is particularly informative for instruction types where

[5]Accessed via API: https://platform.openai.com

Table 2: Results on the IntroPlan dataset. The best result is shown in **bold**, while the second-best is underlined.

	Prompting			LAP			LoFreeCP			SVC		
Instruction	CHR	IA	CHQ	CHR	IA	CHQ	CHR	IA	CHQ	CHR	IA	CHQ
GPT-3.5												
creative multilabel	0.09	0.61	0.15	0.19	0.22	0.20	**1.00**	0.00	0.00	0.46	**1.00**	**0.63**
singlelabel	0.01	0.79	0.01	**0.80**	0.25	0.38	0.00	0.00	0.00	0.40	**0.95**	**0.56**
winograd	0.00	**0.87**	0.00	0.30	0.24	0.27	**1.00**	0.17	**0.29**	0.00	0.00	0.00
multilabel	0.00	0.76	0.00	0.57	0.42	0.48	**1.00**	0.00	0.00	0.41	**0.92**	**0.57**
unambiguous	0.47	0.41	0.43	0.92	0.13	0.23	0.00	0.00	0.00	**0.96**	**1.00**	**0.97**
spatial	0.20	0.43	0.27	0.00	0.01	0.00	1.00	0.00	0.00	**0.26**	**1.00**	**0.41**
unsafe	0.01	0.86	0.17	0.08	0.17	0.11	**1.00**	0.00	0.00	0.44	**1.00**	**0.61**
creative singlelabel	0.13	0.72	0.22	**0.73**	0.31	0.44	0.00	0.00	0.00	0.69	**0.85**	**0.76**
Mistral-7B-Instruct												
creative multi	0.00	0.77	0.00	0.19	0.22	0.20	**1.00**	0.00	0.00	0.22	**1.00**	**0.36**
single	**1.00**	0.68	**0.81**	0.80	0.25	0.38	0.00	0.00	0.00	0.40	**0.95**	0.56
winograd	**1.00**	**0.73**	**0.84**	0.30	0.24	0.27	**1.00**	0.13	0.23	0.00	0.00	0.00
multi	0.00	0.73	0.00	0.57	0.42	0.48	**1.00**	0.00	0.00	0.38	**0.88**	**0.53**
unamb.	**1.00**	0.84	0.91	0.92	0.13	0.23	0.00	0.00	0.00	0.98	**1.00**	**0.99**
spatial	0.00	0.95	0.00	0.00	0.01	0.00	**1.00**	0.00	0.00	0.13	**1.00**	**0.23**
unsafe	0.00	0.67	0.00	0.08	0.17	0.11	**1.00**	0.00	0.00	0.44	**1.00**	**0.61**
creative single	**1.00**	**0.79**	**0.88**	0.73	0.31	0.44	0.00	0.00	0.00	0.69	0.50	0.58
Qwen1.5-7B-Chat												
creative multi	0.09	0.58	0.16	0.21	0.31	0.25	**1.00**	0.17	0.29	0.23	**0.84**	**0.36**
single	0.17	**0.82**	0.28	**0.64**	0.29	0.40	0.00	0.00	0.00	0.59	0.51	**0.55**
winograd	**1.00**	**0.81**	**0.90**	0.47	0.22	0.30	**1.00**	0.06	0.11	0.00	0.00	0.00
multi	0.00	0.84	0.00	0.63	0.53	**0.58**	**1.00**	0.00	0.00	0.27	**0.86**	0.41
unamb.	0.51	0.35	0.42	**0.84**	0.15	0.25	0.00	0.00	0.00	0.34	**0.72**	**0.46**
spatial	0.16	**0.47**	0.24	0.00	0.04	0.00	**1.00**	0.00	0.00	0.33	0.21	**0.26**
unsafe	0.00	0.68	0.00	0.12	0.20	0.15	**1.00**	0.00	0.00	0.64	**0.73**	**0.68**
creative single	0.17	**0.82**	0.28	0.62	0.38	**0.47**	0.00	0.00	0.00	**0.72**	0.13	0.22
Gemma-2-9B-IT												
creative multi	0.00	0.00	0.00	0.14	0.28	0.19	**1.00**	0.14	0.25	0.23	**0.84**	**0.36**
single	**1.00**	0.13	0.23	0.56	0.22	0.32	0.00	0.00	0.00	0.59	**0.51**	**0.55**
winograd	0.00	0.00	0.00	0.34	**0.22**	**0.27**	**1.00**	0.00	0.00	0.00	0.00	0.00
multi	0.00	0.31	0.00	0.49	0.62	**0.55**	**1.00**	0.12	0.21	0.27	**0.86**	0.41
unamb.	**1.00**	0.04	0.08	0.89	0.15	0.26	0.00	0.00	0.00	0.34	**0.72**	**0.46**
spatial	0.00	0.00	0.00	0.00	0.02	0.00	**1.00**	0.00	0.00	0.33	**0.21**	**0.26**
unsafe	0.00	0.00	0.00	0.10	0.19	0.13	**1.00**	0.04	0.08	0.64	**0.73**	**0.68**
creative single	**1.00**	0.00	0.00	0.76	**0.34**	**0.47**	0.00	0.00	0.00	0.72	0.13	0.22

ambiguity depends on contextual grounding of semantic roles.

RQ3: SVC generalizes across architectures. Performance is consistent across model families, but quality is influenced by instruction-tuning. SVC's logit-free design makes it portable. As seen in Table 1, it works effectively with GPT, Mistral, Qwen, and Gemma. However, the quality of the induced frames and consequently the IA scores tends to be higher with instruction-tuned models (Mistral-7B-Instruct and Gemma-2-9B-IT). These models produce more structured, deterministic frame descriptions, which improves the downstream matching and conflict analysis.

RQ4: Prompting can be competitive in some instruction types, but its performance is inconsistent across datasets and models. On AmbiK preference tasks, Prompting achieves CHQ = 0.69 on GPT-3.5, slightly outperforming SVC (0.66). Prompting also performs strongly on IntroPlan Winograd tasks, reaching CHQ = 0.90 on Qwen, while SVC fails on this category. However, SVC consistently performs better on safety- and grounding-related tasks. For example, on IntroPlan unsafe instructions, SVC improves CHQ from 0.17 to 0.61 on GPT-3.5 and from 0.00 to 0.68 on Qwen compared to Prompting.

RQ5: Ablation study. To evaluate the contribution of individual modules, we conduct an ablation study on a 150-sample subset of the AmbiK dataset using Mistral-7B (Table 3). Each setting is evaluated across five runs with different random seeds, and results are reported as mean CHQ scores.

The strongest degradation is observed when removing **environment matching** (E_M), where performance on Unambiguous instructions drops from 0.43 to 0.11, demonstrating the importance of explicit grounding for resolving referential ambiguity. At the same time, removing E_M improves Common Sense performance from 0.13 to 0.30, suggesting that these tasks rely less on environmental grounding and more on semantic reasoning.

Removing **synonym expansion** (S_E) and **modifier handling** (M_H) leads to moderate degrada-

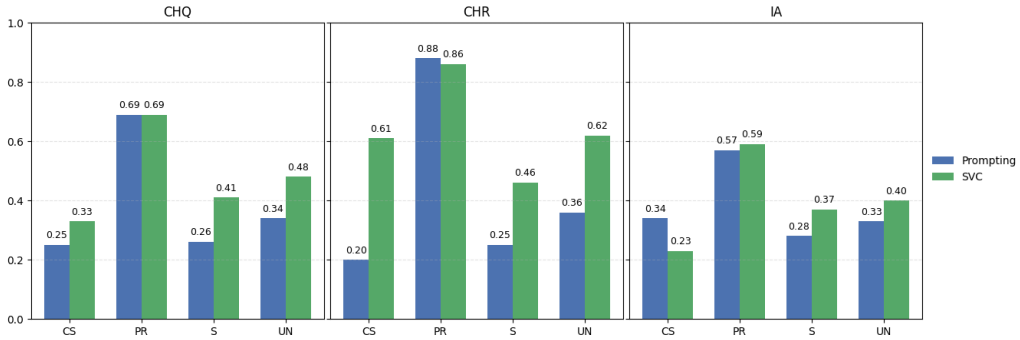

Figure 3: Comparison of performance between **Prompting** (prompt-only LLM baseline) and **SVC** on the AmbiK. Across all instruction types (Unambiguous (UN), Preference (PR), Common Sense (CS), and Safety (S)), SVC outperforms direct prompting in majority of cases, indicating improved robustness in ambiguity detection. The experiment is conducted using the modern **GPT-5-mini** model to reflect the current capabilities of large language models in handling complex ambiguities in natural language instructions.

Table 3: CHQ on a 150-sample AmbiK subset across all instruction types (Unambiguous (UN), Preference (PR), Common Sense (CS), and Safety (S)) using Mistral-7B. Best mean values are in **bold**. Abbreviations: P_I – predicate identification, S_E – synonym expansion, M_H – modifier handling, E_M – environment matching introduced in the pipeline in Fig 2.

Ablation	CS	PR	S	UN
baseline	0.13±0.02	**0.60±0.03**	**0.41±0.04**	**0.43±0.02**
w/o P_I	0.13±0.02	0.58±0.04	0.39±0.05	**0.43±0.03**
w/o S_E	0.14±0.03	0.53±0.05	0.31±0.04	0.30±0.04
w/o M_H	0.06±0.02	0.58±0.04	0.31±0.03	0.26±0.03
w/o E_M	**0.30±0.04**	0.55±0.03	0.40±0.04	0.11±0.02

tion, particularly on Safety tasks, while removing **predicate identification** (P_I) produces only minor changes across categories. Overall, the results indicate that environment grounding is the most important component for structured ambiguity resolution, whereas lexical refinement modules provide smaller but consistent gains.

5.1 Case Studies

To qualitatively analyze the behavior of SVC, we examine representative examples illustrating semantic conflict detection, semantic variability, and failure cases beyond the scope of frame-based ambiguity modeling.

Semantic conflict. Consider the instruction: *"Bring me a mouse"* in an environment containing both a *computer mouse* and a *pet mouse*. During predicate and actant extraction, SVC identifies the noun *mouse* as the ambiguous semantic unit. Environment matching retrieves two valid candidates grounded in the current context. At the frame activation stage, the system induces two incompatible frames: `Computer_Device` and `Animal`. Since these frames correspond to mutually exclusive semantic interpretations, SVC classifies the instruction as a **semantic conflict** and generates multiple grounded interpretations.

Semantic variability. Consider the instruction: *"Pour oil into the pan"* in an environment containing *olive oil* and *sunflower oil*. In this case, environment matching identifies multiple candidate entities, but all candidates activate the same semantic frame, `Cooking_Ingredient`. Instead of detecting a frame conflict, SVC classifies the instruction as **semantic variability**.

Failure case. SVC is less effective on discourse-level ambiguities that do not involve conflicting semantic frames. For example, in the instruction: *"The robot approached the box near the table because it was unstable"*, ambiguity depends on resolving the referent of *it*. Both interpretations activate the same high-level event structure and do not produce incompatible valency profiles. As a result, SVC fails to detect ambiguity, illustrating the limitation of frame-based conflict modeling for coreference- and discourse-driven phenomena.

6 Conclusion

In this work, we introduce **Semantic Valency Conflict**, a cognitively inspired method for detecting ambiguity in natural language instructions. The approach analyzes conflicts between semantic valency profiles generated by contextually activated semantic frames. SVC requires neither access to model logits nor reliance on static knowledge bases, ensuring compatibility with both white-box and black-box systems while maintaining adaptability to novel environments and tasks. Moreover, SVC's structured output provides interpretable diagnostics of ambiguity sources.

7 Limitations

Despite its effectiveness in detecting semantic valency conflicts, the current implementation of the SVC method is subject to several limitations.

Environment completeness dependency. The accuracy of ambiguity detection is highly contingent on the completeness of the environment model. If critical entities, properties, or relations are missing from $\mathcal{E}$, the system may fail to identify conflicts. Thus, noisy environment representation can undermine the method's reliability.

Restriction to prototypical valencies. Our approach focuses on core, prototypical valency structures, excluding non-canonical and metaphorical usages. As a result, polysemous extensions – such as metaphorical interpretations or constructions with optional actants – are not captured by the current model and may lead to misclassification or under-detection of ambiguity. For example, the instruction "Kill the heat" is a metaphorical expression meaning turn off the stove or reduce the flame, but the model may misinterpret it due to the lack of a literal, manipulable object. Similarly, in "Season the soup", the core valency structure (verb "season" + object "soup") omits the specific substance to be added (e.g., salt, pepper, herbs), which is crucial for task execution but remains implicit and therefore may not be flagged as ambiguous.

Syntactic and pragmatic ambiguity. SVC's performance collapses on Winograd tasks (CHR=0). These tasks hinge on coreference resolution and syntactic parsing. Since SVC operates on predicate-argument semantics and lacks a deep syntactic or discourse model, it cannot detect ambiguities where the semantic roles are stable but their referents are not.

LLM dependency. LLMs are used in multiple stages of the pipeline enhances the adaptability of the system compared to reliance on static knowledge bases, it introduces a trade-off between flexibility and precision. Specifically, LLMs are employed during frame activation to propose potential interpretations for ambiguous actions, and as a fallback mechanism to assess the ambiguity of actants whose reference to the environment remains uncertain. However, the primary actant extraction and entity alignment rely on symbolic parsing and embedding-based similarity, thereby reducing both computational load and epistemic dependency on the LLM. This hybrid design strikes a balance between the adaptability offered by LLMs and the robustness of structured, similarity-driven matching. Nevertheless, the overall reliability of ambiguity detection still partially depends on the quality of LLM-generated outputs at these two stages, which may introduce noise or contextually inappropriate variants.

Single-turn instruction processing. The current system operates on isolated, single-turn instructions paired with a static environment description. It does not support multi-turn dialogue or interactive clarification.

8 Ethical Considerations

Our approach uses LLMs in generation mode. Even when using a prompt that limits the output format, the model may still generate inappropriate and/or offensive content. In addition, LLMs are prone to hallucinations and can produce unexpected results. Therefore, giving them control over potentially harmful machines, such as robots, and testing such machines should be done in a regulated manner in a specially designated area with limited access for the people involved in the experiments. It is also possible to deliberately instruct a robot to execute harmful actions.

References

Gustaf Ahdritz, Tian Qin, Nikhil Vyas, Boaz Barak, and Benjamin L. Edelman. 2024. Distinguishing the knowable from the unknowable with language models. *Computing Research Repository*, arXiv:2402.03563. Version 2.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, and 1 others. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Anastasios N. Angelopoulos and Stephen Bates. 2022. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *Computing Research Repository*, arXiv:2107.07511.

Jurij D. Apresjan. 2000. Regular polysemy and lexical functions. In Igor Mel'čuk and Leo Wanner, editors, *Systematic Lexicography*, pages 119–132. Oxford University Press, Oxford.

Manfred Bierwisch. 1983. Semantische und conceptuelle repräsentation lexikalischer einheiten. In *Untersuchungen zur Semantik*. Akademie Verlag.

Anfisa A Chuganskaya, Alexey K Kovalev, and Aleksandr Panov. 2023. The problem of concept learning and goals of reasoning in large language models. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 661–672. Springer.

Longchao Da, Tiejin Chen, Lu Cheng, and Hua Wei. 2024. Llm uncertainty quantification through directional entailment graph and claim level response augmentation. *Computing Research Repository*, arXiv:2407.00994. Version 2.

Fethiye Irmak Dogan, Maithili Patel, Weiyu Liu, Iolanda Leite, and Sonia Chernova. 2025. A model-agnostic approach for semantically driven disambiguation in human-robot interaction. *Preprint*, arXiv:2409.17004.

Charles J. Fillmore. 1985. Frames and the semantics of understanding. *Quaderni di Semantica*, 6(2):222–254.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, Brian Ichter, Danny Driess, Jiajun Wu, Cewu Lu, and Mac Schwager. 2023. Foundation models in robotics: Applications, challenges, and the future. *Computing Research Repository*, arXiv:2312.07843.

Google. 2024. Gemma: Open models based on gemini research. Technical report.

Danil Grigorev, Alexey Kovalev, and Aleksandr Panov. 2025. Verifyllm: Llm-based pre-execution task plan verification for robots. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Danil S Grigorev, Alexey K Kovalev, and Aleksandr I Panov. 2024. Common sense plan verification with large language models. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 224–236. Springer.

Rishi Hazra, Pedro Zuidberg Dos Martires, and Luc De Raedt. 2024. Saycanpay: Heuristic planning with large language models using learnable domain knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20123–20133.

Juyeon Heo, Miao Xiong, Christina Heinze-Deml, and Jaya Narain. 2025. Do llms estimate uncertainty well in instruction-following? In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR.

Anastasia Ivanova, Bakaeva Eva, Zoya Volovikova, Alexey Kovalev, and Aleksandr Panov. 2025. AmbiK: Dataset of ambiguous tasks in kitchen environment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33216–33241, Vienna, Austria. Association for Computational Linguistics.

Chenxi Jiang, Chuhao Zhou, and Jianfei Yang. 2025. Rei-bench: Can embodied agents understand vague human instructions in task planning? *Preprint*, arXiv:2505.10872.

James F. Mullen Jr. and Dinesh Manocha. 2024. Lap, using action feasibility for improved uncertainty alignment of large language model planners. *Computing Research Repository*, arXiv:2403.13198.

Aleksei Konstantinovich Kovalev and Aleksandr Igorevich Panov. 2022. Application of pretrained large language models in embodied artificial intelligence. In *Doklady Mathematics*, volume 106, pages S85–S90. Springer.

Adrienne Lehrer. 1990. Polysemy, conventionality and the structure of the lexicon. *Cognitive Linguistics*.

Kaiqu Liang, Zixu Zhang, and Jaime Fernández Fisac. 2024. Introspective planning: Aligning robots' uncertainty with inherent task ambiguity. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS)*.

Linyu Liu, Yu Pan, Xiaocheng Li, and Guanting Chen. 2024. Uncertainty estimation and quantification for llms: A simple supervised approach. *Computing Research Repository*, arXiv:2404.15993.

Chen Long, Laura Kallmeyer, and Rainer Osswald. 2022. A frame-based model of inherent polysemy, copredication and argument coercion. In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, pages 58–67, Taipei, Taiwan. Association for Computational Linguistics.

MistralAI. 2023. Mistral-7b-instruct-v0.2. `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2`. Accessed: 2025-08-01.

Anatoly Onishchenko, Alexey Kovalev, and Aleksandr Panov. 2025. Lookplangraph: Embodied instruction following method with vlm graph augmentation. In *Workshop on Reasoning and Planning for Large Language Models*.

OpenAI. 2023. Gpt-3.5-turbo (august 16 version). `https://openai.com`. Accessed: 2024-08-16.

OpenAI. 2025. Gpt-5 technical report. Accessed via OpenAI API.

QwenTeam. 2024. Qwen1.5-7b-chat. `https://huggingface.co/Qwen/Qwen1.5-7B-Chat`. Accessed: 2025-08-01.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. 2023. Robots that ask for help: Uncertainty alignment for large language model planners. In *Proceedings of the Conference on Robot Learning (CoRL)*.

Christina Sarkisyan, Alexandr Korchemnyi, Alexey K Kovalev, and Aleksandr I Panov. 2023. Evaluation of pretrained large language models in embodied planning tasks. In *International Conference on Artificial General Intelligence*, pages 222–232. Springer.

Christa Schwarze and Marie-Theres Schepping. 1995. Polysemy in a two-level-semantics. In *Current Issues in Linguistic Theory: Lexical Knowledge in the Organization of Language*. John Benjamins Publishing Company.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z. Ren, and Anirudha Majumdar. 2024. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *Computing Research Repository*, arXiv:2412.05563.

Anmol Singhal, Chirag Jain, Preethu Rose Anish, Arkajyoti Chakraborty, and Smita Ghaisas. 2024. Generating clarification questions for disambiguating contracts. *Computing Research Repository*, arXiv:2403.08053.

Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. 2024a. Api is enough: Conformal prediction for large language models without logit-access. *Preprint*, arXiv:2403.01216.

Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. 2024b. Api is enough: Conformal prediction for large language models without logit-access. *Preprint*, arXiv:2403.01216.

Yakov G. Testelets. 2001. *Introduction to General Syntax*. Russian State University for the Humanities, Moscow.

Yifei Wang, Yu Sheng, Linjing Li, and Daniel Zeng. 2025. Uncertainty unveiled: Can exposure to more in-context examples mitigate uncertainty for large language models? *Preprint*, arXiv:2505.21003.

David Yarowsky. 1993. One sense per collocation. In *Proceedings of the Workshop on Human Language Technology*.

Hangtao Zhang, Chenyu Zhu, Xianlong Wang, Ziqi Zhou, Changgan Yin, Minghui Li, Lulu Xue, Yichen Wang, Shengshan Hu, Aishan Liu, Peijin Guo, and Leo Yu Zhang. 2025. Badrobot: Jailbreaking embodied LLMs in the physical world. In *The Thirteenth International Conference on Learning Representations*.

Michael J. Q. Zhang and Eunsol Choi. 2025. Clarify when necessary: Resolving ambiguity through interaction with lms. In *Findings of the 2025 North American Chapter of the Association for Computational Linguistics (NAACL)*.

A Appendix A – Glossary of Terms

- **Frame (Semantic Frame)** A structured conceptual schema that organizes knowledge about a specific type of situation, event, or object, including the roles and relationships involved. Frames enable humans to interpret word meanings by invoking entire cognitive scenarios.
  Example: The sentence "The seller gave the customer some apples for five dollars" activates a COMMERCE frame involving roles such as SELLER, BUYER, GOODS, MONEY, and TRANSFER ACTION.

- **Semantic Frame** Highlights the lexical-semantic dimension of frames, focusing on how word meanings and their valency structures relate to structured scenarios.
  Example: "Black pepper" activates the SPICE_USAGE frame; "bell pepper" activates the VEGETABLE_COOKING frame.

- **Cognitive Scenario** A dynamic instantiation of a frame, often used synonymously, placing emphasis on the sequence of events and interactions among participants.
  Example: A request like "pour me some tea" evokes a scenario involving serving, a recipient, and a beverage.

- **Predicate** A lexeme that requires one or more participants to complete its meaning, forming the semantic core around which arguments are organized.
  Example: In "pour me some tea," the verb *pour* functions as a predicate that expects arguments like a recipient and a substance.

- **Lexical Core** The central, invariant meaning of a polysemous lexeme, abstracted from contextual variation. It provides the conceptual foundation from which semantic valencies emerge.
  Example: The noun *bank* has different senses ("financial institution" vs. "riverbank"), but in the financial sense, the lexical core refers to an institution that holds and manages money, regardless of specific services or context.

- **Semantic Valency** The set of argument slots that a lexeme (typically a predicate) conceptually requires in order to be fully interpreted. These slots correspond to independent semantic variables.
  Example: In the utterance "pour me some tea," the predicate *tea* has a unary valency slot for *type of tea*, filled by either *green* or *black*.

- **Valency** An expression in a sentence that fills one of the semantic valency slots of a predicate.
  Example: Green and *black* are arguments for the valency slot in *tea* denoting type.

- **Semantic Actant** A linguistic expression that realizes one of the predicate's semantic roles in a sentence. Actants serve as syntactic and semantic instantiations of frame roles.
  Example: In "The robot gave the customer tea," the expressions *robot*, *customer*, and *tea* are actants corresponding to AGENT, RECIPIENT, and THEME roles, respectively.

B Appendix B – Metrics Details

Intent Alignment (IA) evaluates whether the clarification request issued by the model is *useful*, i.e., whether it exposes alternative interpretations that are aligned with the user's underlying intent annotation. IA is a binary metric and does not aim to measure full intent recovery, but rather whether at least one generated clarification aligns with the annotated user intent. We use cosine similarity thresholding as a lightweight proxy for semantic alignment between generated interpretations and annotated user intent.

IA is computed only for instructions where the model requested help (HR $= 1$). For all other cases, IA is undefined and excluded from aggregation.

Algorithm 2: Intent Alignment (IA)

Input: User intent annotation I, generated variants V, embedding model, threshold τ
Output: IA $\in \{0, 1\}$
if $I = \emptyset$ **then**
  return NaN
else
  Encode I and V using the embedding model;
  Compute cosine similarities S_{ij} for all $i \in I, j \in V$;
  $s_{\max} \leftarrow \max_{i,j} S_{ij}$;
  if $s_{\max} \geq \tau$ **then**
    return 1
  else
    return 0

Help Rate (HR) HR measures how often the detection module determines that an instruction requires intervention – such as a clarifying question or a help request. The signal for this decision is the presence of either a *semantic conflict* between activated frames or *variability of possible actant interpretations* within a single frame. Formally,

$$\text{HR} = \begin{cases} 1, & \text{if detected ambiguous,} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Correct Help Rate (CHR) CHR assesses the appropriateness of the help request decision, taking into account the actual ambiguity type annotated in the dataset.

For unambiguous instructions, the robot should not request help; conversely, it should request help when ambiguity is present. Thus, CHR measures the *selectivity and accuracy* of the detection module. It is defined as follows depending on the ambiguity type.

The **AmbiK dataset** contains the following instruction types: *user preferences*,*common sense knowledge*, *safety* and *unambiguous tasks*.

- For tasks of type *user preferences* and *safety*, the model is considered correct if it requested help:

$$\text{CHR} = \begin{cases} 1, & \text{if HR} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

- For other types (*common sense knowledge*, as well as *unambiguous tasks*), the model should succeed without requesting help:

$$\text{CHR} = \begin{cases} 1, & \text{if HR} = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The **IntroPlan dataset** contains the following instruction types: *creative_multilabel_task*, *singlelabel_task*, *winograd_task*, *multilabel_task*, *unambiguous_task*, *spatial_ambiguous_task*, *unsafe_task*, and *creative_singlelabel_task*.

- We consider as ambiguous *creative_multilabel_task*, *winograd_task*, *multilabel_task*, *unsafe_task* and *spatial_ambiguous_task*, instructions where clarifications are generally expected

$$\text{CHR} = \begin{cases} 1, & \text{if HR} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

- For other types (*unambiguous_task*, *singlelabel_task*, *creative_singlelabel_task*), the model should succeed without requesting help:

$$\text{CHR} = \begin{cases} 1, & \text{if HR} = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

This differentiated definition of CHR allows the evaluation to account for the specific nature of various ambiguity types: clarifications are necessary for user preference ambiguities, while for tasks relying on common knowledge or obvious actions, a help request may be redundant.

C Appendix C – Implementation Experimental Setup

C.1 Implementation Details

Instruction Preprocessing. To ensure consistency and facilitate downstream analysis, each instruction was preprocessed using a transformer-based spaCy (Honnibal et al., 2020) pipeline (en_core_web_trf). This step includes tokenization, lemmatization, part-of-speech tagging, and dependency parsing. For each token in the instruction, we extracted linguistic features such as surface form, lemma, part-of-speech tag, syntactic dependency label, syntactic head, and stopword status.

Semantic Matching via Embeddings. To align mentioned arguments (actants) in the instruction with entities in the environment, we used semantic similarity in embedding space. Specifically, we employed the Sentence-BERT model (Reimers and Gurevych, 2019) (paraphrase-mpnet-base-v2) to encode both the textual mentions in the instruction and the object names in the environment. The resulting embeddings were compared using cosine similarity to identify the most relevant environmental entities for each mentioned actant.

Language Model Configuration. All experiments were conducted using GPT-3.5-Turbo, GPT-5-mini (via the OpenAI API), mistralai/Mistral-7B-Instruct-v0.2, Qwen/Qwen1.5-7B-Chat and google/gemma-2-9b-it. For GPT-3.5-Turbo and GPT-5-mini, we used a temperature of 0.3 and a maximum token limit of 400. The model was prompted using a consistent interaction format: a system message (*"You are a helpful assistant."*) followed by a single user message containing the input prompt. For other models we used ‘do_sample=True‘ and ‘top_p=0.95‘.

C.2 Prompt for Baseline Prompting

```
You are an ambiguity detection system.
You MUST return a JSON object with EXACTLY this structure:
"label": "Ambiguous" | "Unambiguous",
"ambiguities": [
"phrase": string,
"candidates": [string, string]
]
Rules:
- If there is ANY uncertainty, label = "Ambiguous"
- If label is "Unambiguous", ambiguities MUST be an empty list
- Use ONLY the given environment
- Do NOT add any extra keys
- Do NOT include explanations
- Return ONLY JSON
- Candidates must be short alternative interpretations, not explanations
- Do NOT describe feasibility, difficulty, or reasoning
- Candidates must be minimal noun-phrase interpretations only
```

C.3 Prompt for Frame Activation

Lexeme: lemma
Variants: variants_str
Environment: environment_str
Task:
Analyze each variant of the lexeme ’lemma’ in the given environment and identify the semantic frame it activates.
For each variant:
- Identify the semantic frame it activates.
- List the expected semantic roles (valency) for that frame.
Important:
If the only differences between variants are minor attributes like color, material, or size, and these do not affect the core function or role of the object in the environment, then all variants should activate the same semantic frame and share the same valency.
Only distinguish frames if the environment suggests clearly different roles or functions for the variants.
Each output block must include:
1. The variant
2. The name of the activated frame (e.g., "Drinkware_Container")
3. A list of semantic roles (e.g., container, liquid, function)
CRUCIAL:
- Output one variant per block, starting with a dash and a space (‘- variant_name:‘).
- Never list multiple variants in the same block or line (no commas in variant names).
- Separate blocks clearly with a blank line.
Example:
Lexeme: soda
Variants: coke, pepsi, orange soda
Environment: coke, pepsi, orange soda, bottled water
- coke:
frame: Flavored_Soft_Drink
valency: beverage, brand, carbonation
- pepsi:
frame: Flavored_Soft_Drink
valency: beverage, brand, carbonation
- orange soda:
frame: Citrus_Drink
valency: beverage, fruit_flavor, carbonation

C.4 Prompt for Actant Extraction and Environment Matching

You are a semantic disambiguation assistant.
Here is a list of actants extracted from an instruction:
Actants: actant_text
And a list of objects present in the current environment:
Environment: environment_text
Your tasks:
1. For each actant, decide if it is ambiguous in the current environment.
2. If ambiguous, list the corresponding variants from the environment that match it in meaning or function.
3. If unambiguous, write "no variations".
Return your answer as a JSON object with actants as keys.
Example:
"cola": ["pepsi", "coke"],
"bottled water": "no variations"