Resolving Ambiguity in Embodied Instructions via Semantic Valency Conflict

Anonymous 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 robot-directed instructions. 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 Introspective Planning, and demonstrate that it outperforms existing baselines in detecting subtle ambiguities in natural language instructions given to robots.

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), and perform planning tasks (Hazra, Martires, and Raedt 2024). 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.

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. 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

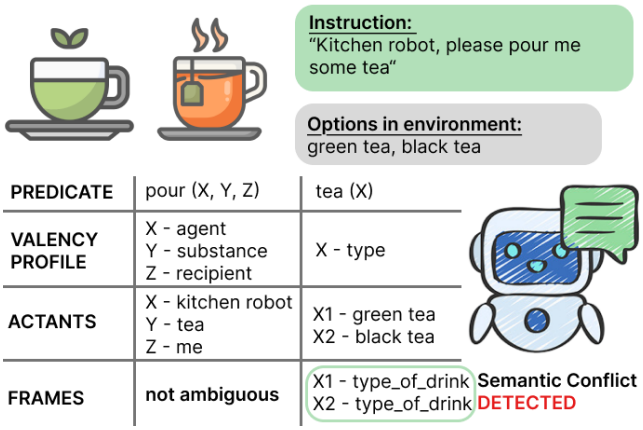

Figure 1: The robot receives the instruction "Pour me some tea" in a setting with green and black tea. The method detects ambiguity by identifying competing interpretations of the predicate "tea", where "green" and "black" act as conflicting actants within the same action frame, revealing a preference-related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

(Figure 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 detection plays a vital role in this process, ensuring the safe integration of LLMs into intelligent agents (Firoozi et al. 2023). Accurate interpretation and execution of instructions are therefore crucial for building reliable and safe robotic systems.

Common techniques for ambiguity resolution include generating multiple candidate interpretations ranked by contextual relevance and interactive clarification (Ren et al. 2023). Some approaches also incorporate external knowledge bases to improve accuracy, but this reduces the system's autonomy and adaptability in novel environments (Liang, Zhang, and Fisac 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 and resolve subtle ambiguities inherent in natural language instructions.

To address ambiguity in natural language instructions caused by lexical polysemy and underspecified argument structures, we propose the **Semantic Valency Conflict** (SVC) method. The key idea is that a single word (or 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 Figure 1). By modeling ambiguity as a conflict between these profiles under a given environment (set of objects, properties, and relations present in the external context), SVC determines whether an instruction supports multiple mutually exclusive interpretations.

The SVC method identifies lexical units whose frame-induced valency profiles are incompatible within the current environment. This is achieved through predicate identification, dynamic frame induction using LLMs, and contextual role alignment.

Unlike prior approaches, SVC detects fine-grained semantic conflicts without logit access or static knowledge bases, enabling interpretability and adaptability in both white- and black-box settings. Here, fine-grained ambiguity refers to subtle semantic inconsistencies arising at a detailed level of meaning, such as nuanced conflicts between semantic roles, argument structures, or contextual interpretations, rather than coarse or surface-level ambiguities. This granularity allows SVC to identify and clarify specific sources of misunderstanding that traditional methods often overlook, thereby enhancing the precision and reliability of ambiguity detection.

We evaluated SVC method ability to detect ambiguity on two fully textual datasets: IntroPlan Mobile Manipulation (Liang, Zhang, and Fisac 2024) and AmbiK (Ivanova, Kovalev, and Panov 2024). Compared to logit-based and heuristic baselines, which either fail to detect ambiguity or ignore uncertain cases entirely, SVC demonstrates superior performance in identifying semantically ambiguous instructions, particularly in user-preference contexts.

Statement of contributions. In this work, we propose Semantic Valency Conflict, a new method for ambiguity detection grounded in frame-based valency analysis. We develop logit-free architecture that is compatible with both white-box and black-box language models. Unlike prior work, our method with LLM dynamically induces cognitive frames without relying on static knowledge bases or plans, making it adaptable. Finally, SVC provides structured outputs that explain the nature of ambiguity.

Related Works

LLMs exhibit uncertainty due to external and internal factors. External factors, like data noise, ambiguity, and lack of contextual information can sometimes be mitigated with clarifying questions (Zhang and Choi 2025). Internal factors, such as model architecture, training limitations, and probabilistic text generation can be reduced through better training and model refinement or by increasing the number and diversity of in-context examples (Wang et al. 2025).

Various methods are used for resolving ambiguity, including token-based, self-verbalized, semantic-similarity, and mechanistic interpretability methods (Shorinwa et al. 2024). Model decision analysis using methods such as directed entailment graphs improves the transparency of LLM reasoning (Da et al. 2024). However, detecting uncertainty is only the first step. The key challenge is mitigating or resolving it.

One approach to reducing ambiguity in LLMs is the generation of clarifying questions, where the model requests additional context to improve confidence in its response (Zhang and Choi 2025). Another strategy involves model ensembles, which aggregate outputs from multiple independently trained LLMs with different parameters and architectures, reducing overall uncertainty (Liu et al. 2024). Additionally, embeddings from small language models can be used to distinguish between different types of uncertainty (Ahdritz et al. 2024). Methods based on CP have also been proposed to calibrate uncertainty in LLM-based planning systems (Angelopoulos and Bates 2022).

As a result, ambiguity detection is emerging as a distinct line of research, complementary to general-purpose uncertainty quantification. It aims to identify cases where a prompt admits multiple interpretations. NLI-based methods that analyze semantic divergence across generated outputs (Kuhn, Gal, and Farquhar 2023), and techniques like uncertainty decomposition through input reformulation (Hou et al. 2024), offer promising tools for this task. In particular, such methods allow one to differentiate between ambiguity-induced variation and noise or error.

To address ambiguity resolving in robotic instruction-following, various approaches have been introduced in LLM-based planning. KnowNo Ren et al. (2023) is a framework that enables planners to assess and align uncertainty, helping them recognize when they lack confidence and need external input. This ensures statistically reliable task completion through CP. Building on this, Introspective Planning (IntroPlan) (Liang, Zhang, and Fisac 2024) integrates retrieval-augmented planning with CP, allowing models to proactively assess their confidence before taking action. By doing so, it reduces the number of user queries for task clarification, while maintaining statistical safety guarantees. Another complementary approach is LAP (Jr. and Manocha 2024), which includes the A-Feasibility metric. This metric combines scene context and model prompting to evaluate whether an action is both feasible and safe in environment. Ambiguities in large, shared spaces often arise from underspecified instructions that depend on implicit semantic features (e.g. cleanliness, fullness). To address this, Dogan et al. (2025) propose a model-agnostic approach leveraging iterative clarifications grounded in knowledge embeddings to infer missing attributes and improve object localization. A related line of work by Jiang, Zhou, and Yang (2025) emphasizes that such vagueness frequently originates from referring expressions whose meaning is shaped by dialogue context and environmental factors.

In our work, we address a specific and underexplored source of ambiguity: semantic ambiguity in natural language instructions caused by lexical polysemy and underspecified argument roles. Rather than relying on statistical signals or response-level heuristics, we treat ambiguity as a structural conflict between competing semantic frames activated by the same lexeme. Specifically, we focus on ambiguous predi-

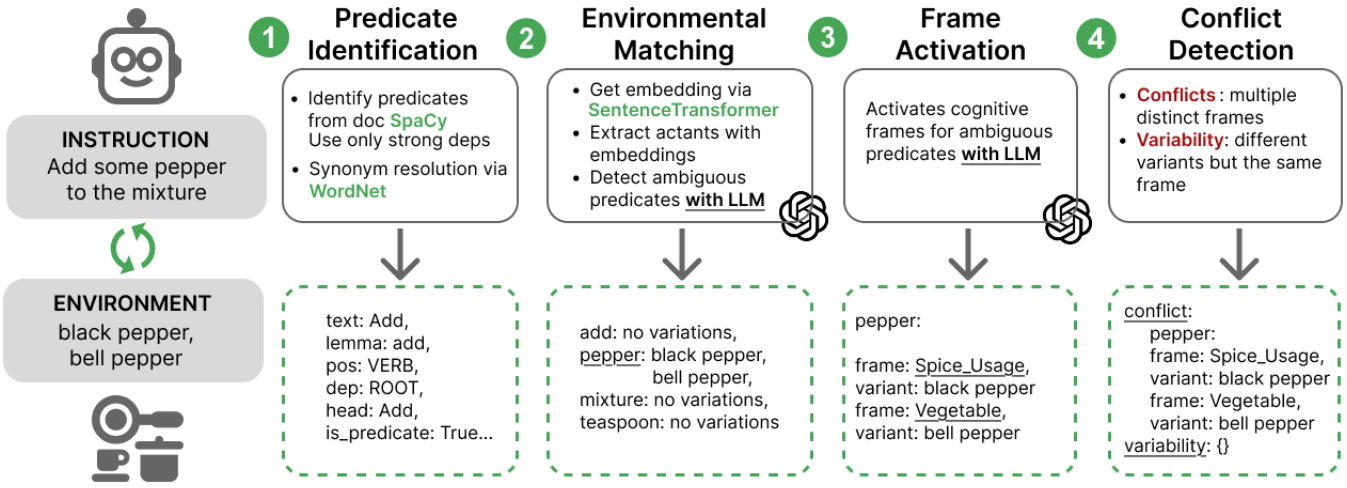

Figure 2: This figure illustrates the pipeline for detecting semantic ambiguity of the SVC method. The architecture decomposes the detection process into four main stages, each corresponding to a step in the cognitive mechanism of frame activation and conflict evaluation. The outputs of each stage of pipeline 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.

cates whose alternative frames impose incompatible valency requirements. By analyzing the completeness and coherence of argument slot realizations across candidate frames, we identify structural conflicts that serve as indicators of semantic ambiguity.

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 conceptual **frame** specifying its semantic roles. 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 conceptual variables X necessary for interpreting L's core meaning. Lexemes that require one or more such variables are **predicates**, and the expressions that realize them in context are **semantic actants** (Testelets 2001). Actants instantiate the predicate's roles in specific situations, linking conceptual 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 conflict: multiple actants (types of tea) compete within the same frame (`Type_of_drink`), revealing ambiguity rooted in underspecified preference (see example in Fig. 1. Our ambiguity detection module follows Yarowsky (1993), who observed that ambiguity often arises when a predicate activates incompatible valency structures simultaneously.

Method

Problem Formulation. Formally, let L be a lexeme (e.g. a verb, noun or pronoun), 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 contextual 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. For example, let L be the noun soda, which can evoke variants such as coke, pepsi, and orange soda. While coke and pepsi activate the frame `Flavored_Soft_Drink` with valencies like beverage, brand, and carbonation, orange soda evokes the frame `Citrus_Drink`, with a distinct valency profile including fruit_flavor.

Each interpretation I_k of L activates a cognitive 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. Formally, the domain $\mathcal{D}$ represents the space of all conceptually valid participant configurations for predicates in natural language. For example, the noun phrase "the mug" may activate the following valency profile under the frame `Drinkware_Container` : container, liquid, function.

To determine whether two valency profiles $\mathcal{V}_k$ and $\mathcal{V}_m$ are semantically compatible in a given environment $\mathcal{E}$, we use the relation $\sim_{\mathcal{E}}$. This relation denotes approximate contex-

Method	CHR				HR				IA			
	UN	PR	CS	S	UN	PR	CS	S	UN	PR	CS	S
GPT												
KnowNo	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.35	0.10	0.22	0.22
LAP	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.41	0.10	0.20	0.23
LoFreeCP	0.77	0.15	0.80	0.24	0.23	0.15	0.20	0.24	0.18	0.10	0.10	0.18
NoHelp	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Binary	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	**0.91**	**0.60**	**0.50**	**0.50**
SVC	0.51	0.77	0.51	0.43	0.49	0.77	0.48	0.43	0.35	0.58	0.26	0.32
Mistral												
KnowNo	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.02	0.03	0.00	0.00
LAP	0.93	0.06	0.95	0.04	0.07	0.06	0.05	0.04	0.12	0.04	0.05	0.02
LoFreeCP	0.28	0.73	0.31	0.62	0.72	0.73	0.69	0.62	0.69	0.52	0.40	0.50
NoHelp	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Binary	1.00	0.01	1.00	0.00	0.00	0.01	0.00	0.00	**0.81**	0.47	0.41	**0.61**
SVC	0.51	0.77	0.55	0.44	0.49	0.45	0.77	0.44	0.31	**0.74**	**0.43**	0.45
Qwen												
LAP	0.73	0.72	0.64	0.29	0.27	0.28	0.36	0.29	0.22	0.17	0.21	0.17
NoHelp	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Binary	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	**0.62**	0.18	**0.32**	**0.25**
SVC	0.54	0.71	0.61	0.39	0.46	0.71	0.39	0.39	0.17	**0.37**	0.27	**0.25**

Table 1: Performance of our method (SVC) in comparison to baselines on the AmbiK dataset. Comparison across models (GPT, Mistral, Qwen) and methods. Metrics include Correct Help Rate (CHR), Help Rate (HR), and Intent Alignment (IA), each broken down by instruction type: Unambiguous (UN), Preference (PR), Common Sense (CS), and Safety (S).

tual equivalence: $\mathcal{V}_k \sim_{\mathcal{E}} \mathcal{V}_m$ if their roles can be grounded to overlapping or compatible entities in $\mathcal{E}$, based on semantic similarity. In practice, this is operationalized using vector representations (embeddings) of candidate actants, their modifiers, and their types, matched against the entities and relations present in the environment.

If there exists $k \neq m$ such that $\mathcal{V}_k \not\sim_{\mathcal{E}} \mathcal{V}_m$, then L is ambiguous:

$$\exists\, k \neq m\colon \mathcal{V}_k, \mathcal{V}_m \text{ activated} \wedge \mathcal{V}_k \not\sim_E \mathcal{V}_m \Rightarrow \text{Ambiguity}(L).$$

For a detailed illustration of the valency conflict detection process, see Figure 2.

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, adjectives, or even entire phrases that activate distinct cognitive frames.

Overview of the SVC Method. To address the problem of detecting ambiguity arising from the lexical polysemy and underspecified argument structures, 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, Kallmeyer, and Osswald 2022; Apresjan 2000). In other words, ambiguity arises when a single lexeme simultaneously activates multiple incompatible frames (cognitive scenarios) $\mathcal{F}$.

To implement the SVC method, we propose an architecture (see Figure 2) where each stage addresses a specific task, from preprocessing to ambiguity detection. The overall architecture mirrors the cognitive process of frame activation and conflict resolution, and is implemented in four sequential stages. More details of each stage of SVC architecture are provided in Appendix E.

The actants of each predicate are extracted and aligned with environmental constraints $\mathcal{E}$ (i.e., the set of relevant objects, properties, and relations in the instruction's context). The goal is to determine whether all participant roles required by the predicate are contextually supported. For each possible interpretation I_k of a predicate, the corresponding cognitive frame $\mathcal{F}_k$ is activated. Frame induction is performed dynamically using LLMs, allowing the system to infer plausible frames and their valency structures without relying on static knowledge bases (Liang, Zhang, and Fisac 2024; Chaplot and Salakhutdinov 2018). Each frame implicitly defines a valency profile $\mathcal{V}_k$ over the domain $\mathcal{D}$.

At the final stage, the system evaluates the set of cognitive frames $\{\mathcal{F}_k\}$ activated for each predicate, each associated with a valency profile $\mathcal{V}_k$. The method compares valency profiles $\mathcal{V}_k$ for all competing interpretations. If there exist two interpretations $k \neq m$ such that their valency profiles are semantically incompatible with respect to the environment, formally expressed as

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

the predicate is flagged as ambiguous due to **frame conflict**. This condition indicates that the predicate simultaneously activates multiple, mutually exclusive event schemas, reflecting true semantic ambiguity—for example, the predicate *pepper* activating frames $\mathcal{F}_1 = \texttt{Spice\_Usage}$ and $\mathcal{F}_2 = \texttt{Vegetable\_Cooking}$.

Instruction Type	Binary			NoHelp			LoFreeCP			LAP			SVC		
	HR	CHR	IA	HR	CHR	IA	HR	CHR	IA	HR	CHR	IA	HR	CHR	IA
creative_multilabel	0.00	0.00	0.77	0.00	0.00	0.87	1.00	1.00	0.00	0.19	0.19	0.22	0.22	0.22	**1.00**
single_label	0.00	1.00	0.68	0.00	1.00	0.93	1.00	0.00	0.00	0.20	0.80	0.25	0.60	0.40	**0.95**
winograd	0.00	0.00	0.73	0.00	0.00	**0.84**	1.00	1.00	0.13	0.30	0.30	0.24	0.00	0.00	0.00
multilabel	0.00	0.00	0.73	0.00	0.00	**0.96**	1.00	1.00	0.00	0.57	0.57	0.42	0.38	0.38	0.88
unambiguous	0.00	1.00	0.84	0.00	1.00	0.92	1.00	0.00	0.00	0.08	0.92	0.13	0.02	0.98	**1.00**
spatial_amb	0.00	0.00	0.95	0.00	0.00	0.96	1.00	1.00	0.00	0.00	0.00	0.01	0.13	0.13	**1.00**
unsafe	0.00	0.00	0.67	0.00	0.00	0.59	1.00	1.00	0.00	0.08	0.08	0.17	0.44	0.44	**1.00**
creative_singlelabel	0.00	1.00	0.79	0.00	1.00	**0.93**	1.00	0.00	0.00	0.27	0.73	0.31	0.31	0.69	0.50

Table 2: The performance of our method (SVC) in comparison to baselines on the IntroPlan Mobile Manipulation dataset. Experiments were conducted using the **Mistral**. Help Rate (HR), Correct Help Rate (CHR), and Intent Alignment (IA) by task and method.

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

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

the system interprets this as a **choice within the scenario** rather than a semantic conflict. Such variability represents different referents or attributes within the same event schema (e.g., variants *olive oil* and *sunflower oil* under the frame `Cooking_Oil_Ingredient`). Although this does not signify ambiguity at the frame level, it highlights distinctions that must be resolved for precise grounding or execution. Thus, the ambiguity detection mechanism differentiates between the cases providing a robust foundation for subsequent disambiguation steps.

Experimental Evaluation

Datasets. To evaluate SVC, we used two datasets: **IntroPlan Mobile Manipulation** (Liang, Zhang, and Fisac 2024) and **AmbiK** (Ivanova et al. 2025). Both contain robot instructions within a given environment but differ significantly in ambiguity coverage and annotation detail. These datasets were used only for evaluation; no training or tuning was performed on them. The input to the model consists of a brief scene description paired with a task instruction.

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 dataset (Ren et al. 2023). Although it contains ambiguous examples, it lacks precise ambiguity definitions and has an incomplete typology that overlaps with general linguistic phenomena. IntroPlan mainly assesses model sensitivity to varied phrasing but is insufficient for detailed ambiguity resolution evaluation.

AmbiK is a specialized dataset containing 2,000 annotated instructions covering three types of semantic ambiguity: user preferences, commonsense knowledge, and safety considerations. Each task has clarifying questions, answers, and execution plans for both ambiguous and unambiguous versions, and success markers for disambiguation.

Metrics. To evaluate the quality of ambiguity detection, we employ two binary metrics from Ivanova et al. (2025): **Help Rate (HR)** and **Correct Help Rate (CHR)**. Both metrics rely on the internal decision of the detection module regarding the presence of ambiguity, which is based on analyzing frame conflicts and actant variability.

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 the variability of possible actant interpretations within a single frame. **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, depending on the type of instruction.

To assess the alignment between system behavior and user intent, we additionally report **Intent Alignment (IA)**, which captures whether the method's interpretation or suggested disambiguation correctly preserves the user's intended goal. IA is computed by embedding both predicted and ground-truth intents into a semantic vector space and measuring their cosine similarity.

Finally, we introduce **Contextual Help Quality (CHQ)** to measure the usefulness of contextual clarifications in supporting user intent. CHQ evaluates the combined effect of accurate ambiguity detection and intent preservation, and is defined as the harmonic mean of IA and CHR:

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

where ε is a small constant ($\varepsilon = 10^{-6}$) added for numerical stability. For more information on the metrics, see Appendix C.

Models and Baselines. We conducted experiments using **GPT-3.5-Turbo** (OpenAI 2023) [1], **Mistral-7B-Instruct-v0.2** (MistralAI 2023) [2], and **Qwen1.5-7B-Chat** (Qwen-Team 2024) [3]. For brevity and clarity, we refer to these models simply as GPT, Mistral and Qwen, respectively. For more

[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/Qwen/Qwen1.5-7B-Chat

Instruction	KnowNo	LAP	LoFree	NoHelp	Binary	SVC
Ambik (CHQ↑)						
UN	0.039	0.228	0.319	0.000	**0.936**	0.484
PR	0.000	0.048	0.455	0.000	0.020	**0.755**
CS	0.000	0.095	0.329	0.000	**0.742**	0.397
S	0.000	0.065	0.386	0.000	0.000	**0.435**
IntroPlan (CHQ↑)						
c_multi	0.000	0.204	0.000	0.000	0.115	**0.374**
single_l	0.000	0.381	0.000	**0.964**	0.810	0.563
winograd	0.000	0.267	0.000	0.000	**0.135**	0.000
multi_l	0.000	0.484	0.000	0.000	0.135	**0.531**
unambig	0.000	0.228	0.000	0.958	0.500	**0.990**
spat_amb	0.000	0.000	0.000	0.000	0.025	**0.230**
unsafe	0.000	0.109	0.000	0.000	0.165	**0.611**
c_single	0.000	0.435	0.000	**0.964**	0.500	0.580

Table 3: **Contextual Help Quality (CHQ↑)** of SVC and baseline systems on the Ambik and IntroPlan datasets. CHQ reflects the usefulness of contextual suggestions in supporting user goals. Instruction types in IntroPlan Mobile: c_multi (creative multilabel), single_l (single-label), winograd, multi_l (multilabel), unambig, spat_amb (spatial ambiguity), unsafe, c_single (creative single-label). Instruction types in Ambik: Unambiguous (UN), Preference (PR), Common Sense (CS), and Safety (S). Evaluation on Mistral. Best values are highlighted in **bold**.

implementation details and prompts used in the LLM stages of our pipeline see Appendix D.

We compare our method against **five baselines** previously proposed in the context of instruction following and embodied agents. **KnowNo** (Ren et al. 2023) – one of the first methods to apply CP for ambiguity detection in kitchen task scenarios with LLMs. The LLM is prompted to generate multiple answer options and then select the most appropriate one. **LAP** (Jr. and Manocha 2024) – an extension of KnowNo that integrates affordance estimation. For each candidate response, it multiplies the softmax confidence by two affordance scores: Context-Based Affordance and Prompt-Based Affordance, which queries the LLM to assess whether the action is feasible and safe in the given context. **LoFreeCP** (Su et al. 2024b) – a method that does not rely on logit access and instead uses uncertainty-based conformal prediction over multiple generations. **Binary** (Ren et al. 2023) – a prompting-based baseline where the LLM is asked to generate the most likely answer and explicitly label it as "Certain" or "Uncertain" in a few-shot setting. Help is requested only when the model self-reports uncertainty. **NoHelp** (Ren et al. 2023) – a naive baseline where the LLM is prompted to output a single answer, assuming complete certainty. The agent never asks for clarification.

Results. We evaluated the proposed SVC on AmbiK (Table 1) and IntroPlan Mobile Manipulation (Table 2).

Our evaluation reveals clear advantages of the proposed SVC method over existing baselines across both the AmbiK and IntroPlan datasets. In contrast to heuristic or logit-based approaches, which often fail to capture subtle forms of ambiguity (it can be seen by cases where HR = 0 in all instruction types and CHR = 1, or HR = 1 in all instruction types and CHR = 0), SVC demonstrates robust and precise behavior in identifying situations requiring clarification, particularly those driven by user preferences and contextual underspecification as in the example shown in Fig. 1.

Analysis of the IA metric revealed that naive baselines often achieve high coverage of the ground-truth user intent. However, this alone does not ensure contextual appropriateness. Notably, CHQ scores for naive baselines are significantly lower compared to our proposed method, SVC, which effectively balances intent coverage with contextual relevance.

The most relevant baseline for comparison is LoFreeCP, a method that does not rely on logit access and instead uses uncertainty-based conformal prediction over multiple generations. It combines coarse-grained signals (e.g., answer frequency) with fine-grained metrics (e.g., semantic similarity, normalized entropy), enabling a multi-layered assessment of model confidence. However, even LoFreeCP struggled to capture fine-grained ambiguity, particularly on the AmbiK dataset, where many instructions rely on implicit user preferences, common sence knowledge or context-sensitive interpretations. These limitations underline the need for more semantically aware detection methods.

On the AmbiK dataset (Table 1), SVC consistently outperforms LoFreeCP in the Preference category, achieving a CHR of 0.77 compared to 0.15 (GPT) and 0.73 (Mistral). This demonstrates SVC's capacity to detect ambiguity arising from subjective preferences – one of the most challenging forms of underspecification. Furthermore, SVC achieves higher IA scores (up to 0.74 on Mistral), indicating that its disambiguation strategies often align well with the user's intended goal. While LoFreeCP tends to yield a higher HR, its performance suffers from over-triggering, particularly in common-sense and unambiguous tasks. This leads to low CHR and inflated intervention behavior, which undermines its practical usefulness. According to CHQ, SVC surpasses all other methods across most instruction types, especially in Preference (0.755 vs. 0.455 for LoFreeCP) and Unambiguous (0.484 vs. 0.319), underscoring its ability to provide both relevant and user-aligned support. In Common Sense and Safety categories, SVC also performs competitively.

On the IntroPlan dataset (Table 2), SVC continues to demonstrate strong performance. It achieves the highest CHQ in the creative_multilabel (0.374), multilabel (0.531), unsafe (0.611), and unambiguous (0.990) tasks, indicating that its clarification behavior is both selective and beneficial. These results show that SVC can generalize to a wide range of instruction types.

The Winograd tasks remain a challenge with both CHR and HR equal to zero. This difficulty arises because these tasks primarily hinge on syntactic ambiguity, rather than relying on surface-level semantic roles. Since our valency-based detection method focuses on semantic role analysis, it is inherently limited in addressing such syntactic and pragmatic complexities. Appendix B contains a more detailed account of the limitations associated with our method.

SVC demonstrates consistent performance across all three evaluated models. The highest IA and CHR scores are achieved with Mistral-7B-Instruct-v0.2, which is designed for instruction-following and tends to produce more deterministic, interpretable completions. In contrast, Qwen1.5-7B-Chat, being a chat-oriented model primarily optimized for multi-turn dialogue, exhibits greater variability in its behavior. This makes it less reliable in tasks that require strict adherence to prompt structure and deterministic reasoning. As a result, while Qwen achieves moderate IA in certain tasks, its overall performance is less consistent.

Overall, SVC offers several key **advantages**: **1) Logit-free design**: SVC does not require access to logits, making it suitable for both white-box and black-box deployment scenarios; **2) Fine-grained ambiguity detection**: It effectively identifies subtle forms of ambiguity, particularly in the preferences category, where user intent must be inferred from context. For example, in the instruction *"pour me some tea"*, ambiguity arises when both green and black tea are available in the environment. Without explicit user input, the system should detect the need for clarification. Such cases require semantic sensitivity to object variability, which is beyond the reach of heuristic or logit-based approaches; **3) Minimal input requirements**: SVC operates solely on the instruction text and a textual description of the agent's environment, without requiring additional inputs such as plans, action histories, or external knowledge. This makes it lightweight and easily integrable into existing pipelines; **4) Interpretable output**: Instead of a binary ambiguity flag, SVC provides structured descriptions of possible semantic splits, indicating which arguments may lead to multiple interpretations and under what contextual conditions. For example, given the instruction "Bring me a cola" and the environment ["Pepsi", "bottled unsweetened tea", "Coke"], the system identifies ambiguity in the argument *cola*, mapping it to possible referents Pepsi and Coke. This output is particularly useful for downstream disambiguation modules powered by language models.

Ablation Studies. To elucidate the contribution of individual components within our disambiguation pipeline, we conducted systematic ablation studies aimed at assessing model sensitivity and relative importance of each module.

Our investigation revealed that variations in the generation temperature parameter exerted negligible influence on overall system performance. Empirical evaluation demonstrated that the instruct-style Mistral 7B model consistently surpassed the performance of the architecturally distinct yet parameter-equivalent Qwen 7B chat model. Furthermore, the integration of synonym processing yielded measurable improvements in key metrics, notwithstanding the incomplete lexical coverage of WordNet. In contrast, the explicit handling of modifiers, referential expressions, and spatial relations exhibited limited incremental benefit, suggesting that robust grounding of lexical cores suffices to achieve effective ambiguity resolution.

Critically, ablation of the LLM component within the frame activation module resulted in a substantial decline in performance, thereby confirming the essential role of LLM-based reasoning in capturing and resolving complex semantic frame conflicts. Conversely, exclusion of the LLM from the predicate ambiguity detection phase was found to be acceptable; the rule-based detection mechanism, supplemented by LLM-driven enrichment for particularly challenging cases, maintained robustness without necessitating mandatory LLM involvement.

Conclusion

In this work, we introduce **Semantic Valency Conflict** (SVC), a cognitively inspired, logit-free method for detecting ambiguity in natural language instructions. The approach analyzes conflicts between semantic valency profiles generated by contextually activated cognitive frames. In contrast to existing methods, 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. Comprehensive evaluations on two benchmark datasets demonstrate SVC's superior performance in identifying subtle, context-dependent ambiguities, particularly in user-preference-sensitive scenarios. The method outperforms existing logit-based and heuristic approaches, which either fail to detect such ambiguities or disregard them entirely, thereby compromising interpretation quality and system safety. Moreover, SVC's structured output provides interpretable diagnostics of ambiguity sources, facilitating effective downstream clarification and resolution mechanisms.

References

Ahdritz, G.; Qin, T.; Vyas, N.; Barak, B.; and Edelman, B. L. 2024. Distinguishing the Knowable from the Unknowable with Language Models. *Computing Research Repository*, arXiv:2402.03563. Version 2.

Angelopoulos, A. N.; and Bates, S. 2022. A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification. *Computing Research Repository*, arXiv:2107.07511.

Apresjan, J. D. 2000. Regular Polysemy and Lexical Functions. In Mel'čuk, I.; and Wanner, L., eds., *Systematic Lexicography*, 119–132. Oxford: Oxford University Press.

Bierwisch, M. 1983. Semantische und conceptuelle Repräsentation lexikalischer Einheiten. In *Untersuchungen zur Semantik*. Akademie Verlag.

Chaplot, D. S.; and Salakhutdinov, R. 2018. Knowledge-based Word Sense Disambiguation using Topic Models. *Computing Research Repository*, arXiv:1801.01900.

Da, L.; Chen, T.; Cheng, L.; and Wei, H. 2024. LLM Uncertainty Quantification through Directional Entailment Graph and Claim Level Response Augmentation. *Computing Research Repository*, arXiv:2407.00994. Version 2.

Dogan, F. I.; Patel, M.; Liu, W.; Leite, I.; and Chernova, S. 2025. A Model-Agnostic Approach for Semantically Driven Disambiguation in Human-Robot Interaction. arXiv:2409.17004.

Fillmore, C. J. 1985. Frames and the Semantics of Understanding. *Quaderni di Semantica*, 6(2): 222–254.

Firoozi, R.; Tucker, J.; Tian, S.; Majumdar, A.; Sun, J.; Liu, W.; Zhu, Y.; Song, S.; Kapoor, A.; Hausman, K.; Ichter, B.; Driess, D.; Wu, J.; Lu, C.; and Schwager, M. 2023. Foundation Models in Robotics: Applications, Challenges, and the Future. *Computing Research Repository*, arXiv:2312.07843.

Hazra, R.; Martires, P. Z. D.; and Raedt, L. D. 2024. SayCanPay: Heuristic Planning with Large Language Models using Learnable Domain Knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 20123–20133.

Heo, J.; Xiong, M.; Heinze-Deml, C.; and Narain, J. 2025. Do LLMs Estimate Uncertainty Well in Instruction-Following? In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hou, B.; Liu, Y.; Qian, K.; Andreas, J.; Chang, S.; and Zhang, Y. 2024. Decomposing Uncertainty for Large Language Models through Input Clarification Ensembling. In *Proceedings of the 41st International Conference on Machine Learning*. Vienna, Austria: PMLR.

Ivanova, A.; Eva, B.; Volovikova, Z.; Kovalev, A.; and Panov, A. 2025. AmbiK: Dataset of Ambiguous Tasks in Kitchen Environment. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 33216–33241. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Ivanova, A.; Kovalev, A. K.; and Panov, A. I. 2024. AmbiK: Dataset of Ambiguous Tasks in Kitchen Environment. In *The Fifth Annual Embodied AI Workshop at CVPR 2024*.

Jiang, C.; Zhou, C.; and Yang, J. 2025. REI-Bench: Can Embodied Agents Understand Vague Human Instructions in Task Planning? arXiv:2505.10872.

Jr., J. F. M.; and Manocha, D. 2024. LAP, Using Action Feasibility for Improved Uncertainty Alignment of Large Language Model Planners. *Computing Research Repository*, arXiv:2403.13198.

Kuhn, L.; Gal, Y.; and Farquhar, S. 2023. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Lehrer, A. 1990. Polysemy, Conventionality and the Structure of the Lexicon. *Cognitive Linguistics*.

Liang, K.; Zhang, Z.; and Fisac, J. F. 2024. Introspective Planning: Aligning Robots' Uncertainty with Inherent Task Ambiguity. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS)*.

Liu, L.; Pan, Y.; Li, X.; and Chen, G. 2024. Uncertainty Estimation and Quantification for LLMs: A Simple Supervised Approach. *Computing Research Repository*, arXiv:2404.15993.

Long, C.; Kallmeyer, L.; and Osswald, R. 2022. A Frame-Based Model of Inherent Polysemy, Copredication and Argument Coercion. In Zock, M.; Chersoni, E.; Hsu, Y.-Y.; and Santus, E., eds., *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, 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.

OpenAI. 2023. GPT-3.5-turbo (August 16 version). https://openai.com. Accessed: 2024-08-16.

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

Ren, A. Z.; Dixit, A.; Bodrova, A.; Singh, S.; Tu, S.; Brown, N.; Xu, P.; Takayama, L.; Xia, F.; Varley, J.; Xu, Z.; Sadigh, D.; Zeng, A.; and Majumdar, A. 2023. Robots That Ask For Help: Uncertainty Alignment for Large Language Model Planners. In *Proceedings of the Conference on Robot Learning (CoRL)*.

Schwarze, C.; and Schepping, M.-T. 1995. Polysemy in a Two-Level-Semantics. In *Current Issues in Linguistic Theory: Lexical Knowledge in the Organization of Language*. John Benjamins Publishing Company.

Shorinwa, O.; Mei, Z.; Lidard, J.; Ren, A. Z.; and Majumdar, A. 2024. A Survey on Uncertainty Quantification of Large Language Models: Taxonomy, Open Research Challenges, and Future Directions. *Computing Research Repository*, arXiv:2412.05563.

Singhal, A.; Jain, C.; Anish, P. R.; Chakraborty, A.; and Ghaisas, S. 2024. Generating Clarification Questions for Disambiguating Contracts. *Computing Research Repository*, arXiv:2403.08053.

Su, J.; Luo, J.; Wang, H.; and Cheng, L. 2024a. API Is Enough: Conformal Prediction for Large Language Models Without Logit-Access. arXiv:2403.01216.

Su, J.; Luo, J.; Wang, H.; and Cheng, L. 2024b. API Is Enough: Conformal Prediction for Large Language Models Without Logit-Access. arXiv:2403.01216.

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

Wang, Y.; Sheng, Y.; Li, L.; and Zeng, D. 2025. Uncertainty Unveiled: Can Exposure to More In-context Examples Mitigate Uncertainty for Large Language Models? arXiv:2505.21003.

Yarowsky, D. 1993. One Sense per Collocation. In *Proceedings of the Workshop on Human Language Technology*.

Zhang, H.; Zhu, C.; Wang, X.; Zhou, Z.; Yin, C.; Li, M.; Xue, L.; Wang, Y.; Hu, S.; Liu, A.; Guo, P.; and Zhang, L. Y. 2025. BadRobot: Jailbreaking Embodied LLMs in the Physical World. In *The Thirteenth International Conference on Learning Representations*.

Zhang, M. J. Q.; and Choi, E. 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)*.

Reproducibility Checklist

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA)
Yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) Yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) Yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) Yes

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) Partly

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) No

2.4. Proofs of all novel claims are included (yes/partial/no) No

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) Yes

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) Yes

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) Yes

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) Yes

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) Yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) Yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) NA

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) NA

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) Yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) Yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) Yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) Partial

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) Partial

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) Partial

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) Yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) Partial

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) Partial

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) Partial

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) Yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) No

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) Yes

4.12. The significance of any improvement or decrease in

performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) No

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) Partial