Uncertainty-Aware Planning for Disambiguating User Intent in Interactive LLM Agents: Application to Baidu Maps

Deqiang Huang
University of Science and Technology of China
Hefei, Anhui, China
deqianghuang@mail.ustc.edu.cn

Xinjiang Lu*
Frontier Research Department, Baidu Inc.
Beijing, China
luxinjiang@baidu.com

Jingbo Zhou*
Frontier Research Department, Baidu Inc.
Beijing, China
zhoujingbo@baidu.com

Nijia Lu
Baidu Inc.
Beijing, China
lunijia01@baidu.com

Fuxin Li
Baidu Inc.
Beijing, China
lifuxin01@baidu.com

Bo Hong
Baidu Inc.
Beijing, China
hongbo@baidu.com

Chuanming Zhang
Baidu Inc.
Beijing, China
zhangchuanming@baidu.com

Tong Xu*
University of Science and Technology of China
Hefei, Anhui, China
tongxu@ustc.edu.cn

Enhong Chen
University of Science and Technology of China
Hefei, Anhui, China
cheneh@ustc.edu.cn

Abstract

Large language models (LLMs) are revolutionizing user interactions in online map applications by enabling conversational interfaces with intelligent map agents. However, inherent ambiguities and nuances in human communication often lead to incomplete or unclear user instructions. While generating clarification questions can mitigate this issue, existing methods—which rely on prompt-based LLM assessments or supervised fine-tuning (SFT) of LLMs with limited annotated data—struggle to reliably determine when clarification is necessary, resulting in cold-start issues and reduced adaptability. To address this gap, we propose an uncertainty-aware dynamic planning framework for intent disambiguation in interactive agent systems, exemplified by Baidu Maps. Our framework leverages LLMs to dynamically generate agent action sequences while estimating query uncertainty. When uncertainty thresholds are exceeded, targeted clarification questions are triggered. Besides, to efficiently navigate the vast uncertainty space, we introduce a novel beam search-based pruning strategy. Crucially, the framework supports context-aware task planning without any additional model training. We evaluate our approach on Baidu Maps and a public robotic task, focusing on uncertainty management and adaptability. Experiments demonstrate that the framework achieves high precision across diverse LLMs without requiring an additional training phase, highlighting its versatility and robustness. Ablation studies confirm that the framework can significantly improve user intent recognition and task success rates.

*Corresponding author.

KDD '26, Jeju Island, Republic of Korea

ACM ISBN 979-8-4007-2258-5/2026/08
https://doi.org/10.1145/3770854.3783943

CCS Concepts

• **Computing methodologies** → **Natural language processing**; *Planning and scheduling*; *Probabilistic reasoning*; • **Information systems** → **Geographic information systems**.

Keywords

Uncertainty-aware planning; agent systems; beam search; uncertainty quantification; pruning strategies

ACM Reference Format:
Deqiang Huang, Xinjiang Lu, Jingbo Zhou, Nijia Lu, Fuxin Li, Bo Hong, Chuanming Zhang, Tong Xu, and Enhong Chen. 2026. Uncertainty-Aware Planning for Disambiguating User Intent in Interactive LLM Agents: Application to Baidu Maps . In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3770854.3783943

1 Introduction

The integration of large language models (LLMs) into online map applications is fundamentally transforming user interactions, elevating these platforms into intelligent map agent systems [9, 36]. These applications, accessible to users through multiple modalities such as voice and text, support a range of tasks, including obtaining navigation routes, locating points of interest (POIs), and managing taxi or transit services, thereby significantly enhancing the overall user experience.

However, inherent ambiguities and nuances in human communication – particularly pronounced in voice interactions – often lead users to provide incomplete or unclear instructions to online map agents. These systems, designed for natural conversational engagement, frequently struggle to interpret user intent accurately. Such misinterpretations not only compromise response relevance and user satisfaction but also hinder the broader adoption of intelligent agent technologies. To address this challenge, agents must proactively generate context-aware clarification questions to resolve

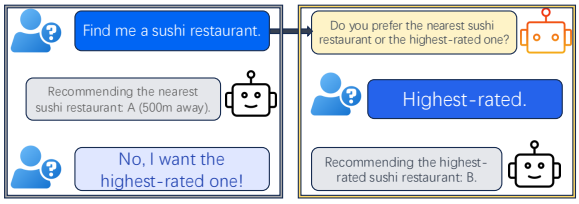

Figure 1: Comparison of responses with and without clarification. Without clarification (left), the agent assumes the user wants the nearest sushi restaurant, leading to possible dissatisfaction. With clarification (right), the system asks for user preference, refining the query for a more accurate recommendation.

ambiguities in time. As demonstrated in Figure 1, this approach refines ambiguous queries and improves response precision.

Recent studies [4, 20] have shown that prompting agents to pose clarification questions enhances their capability to address ambiguous queries. Nevertheless, existing methods face a critical bottleneck: reliably determining when to seek clarification. Present techniques either rely on (1) prompt-based LLM assessments to assess query ambiguity directly or (2) Supervised Fine-Tuning (SFT) [3, 5, 41] using limited annotated datasets. Both approaches have notable drawbacks. LLM-based assessments, while powerful, lack transparency and often prematurely commit to suboptimal interpretations in unclear situations, thus reducing adaptability. Conversely, SFT methods struggle with limited training data, compromising their ability to generalize. These limitations collectively undermine reliable intent recognition and practical application, as systems either over-solicit clarifications (annoying users) or under-solicit them (producing irrelevant responses).

To address these limitations, we propose an uncertainty-aware planning framework that uses LLMs to generate agent action sequences while assessing the uncertainty of user queries. The proposed method improves both performance and interpretability by integrating uncertainty estimation into task planning.We demonstrate the framework's effectiveness through a real-world deployment on Baidu Maps, and highlight its potential to generalize across domains.

Our framework supports context-aware task planning without requiring labeled data or model fine-tuning. By integrating in-context learning with few-shot exemplars and an uncertainty-aware beam search, the system dynamically generates clarification questions tailored to user queries. Uncertainty quantification allows the framework to identify ambiguous inputs requiring further clarification and adapt task planning strategies to diverse contextual demands, significantly broadening its capacity to handle complex or unclear requests. This training-free design ensures immediate compatibility with a wide range of LLMs while mitigating cold-start issues, making the solution both resource-efficient and universally deployable.

In particular, we propose an enhanced beam search algorithm that incorporates uncertainty estimation to refine the pruning strategy. By dynamically adjusting beam width based on uncertainty levels, our approach mitigates error propagation and improves the stability of task planning. This novel design increases the reliability and stability of the generated sequences, ultimately boosting task planning effectiveness.

To evaluate the effectiveness of our framework, we conducted extensive experiments using the Baidu Maps dataset. We also evaluated its generalizability and robustness on the SaGC dataset, a publicly available benchmark for robotic tasks. The experimental results demonstrate that our framework achieves SOTA performance in several key metrics, including uncertainty management and planning efficiency. These results highlight the framework's capacity to handle uncertain inputs effectively, significantly enhancing the capabilities of intelligent map systems. The primary contributions of this paper can be summarized as follows:

- **Training-Free Context-Aware Task Planning**: Our framework operates without additional training, enabling immediate compatibility with various LLMs while avoiding cold start issues. This approach dynamically adapts to context changes in task planning, ensuring generated sequences align with specific application requirements, thereby enhancing cross-model adaptability and reliability.
- **Enhanced Beam Search with Uncertainty Estimation**: We propose an improved beam search algorithm integrating uncertainty quantification to optimize the pruning process. By dynamically adjusting the beam width based on observed uncertainty levels, our approach reduces uncertainty accumulation, enabling more efficient task execution.
- **Comprehensive Evaluation on Benchmark Datasets**: We evaluate the proposed approach on the Baidu Maps and SaGC datasets, showcasing superior performance in uncertainty management and cross-domain adaptability.

In conclusion, our framework integrates uncertainty estimation with advanced beam search strategies to tackle critical challenges in disambiguating user intent for agent applications, and has been successfully deployed in Baidu Maps with demonstrable improvements in clarification precision and user preference.

2 Related Work

2.1 Clarification Question Generation

Clarification Question Generation (CQG) plays a vital role in resolving ambiguities in interactive systems, spanning applications such as code generation [28], legal retrieval [25], open-domain QA [6, 19], and task-oriented dialogue [22, 31]. Retrieval-augmented methods enhance CQG by incorporating external knowledge [28], while frameworks like *CLAM* leverage few-shot prompting to selectively clarify ambiguous queries, improving efficiency [20]. Earlier studies explored template-based generation [37] and discriminative question retrieval [6].

Building on these directions, our work integrates CQG into navigation-specific task planning, producing tailored clarification questions for POI searches and route planning to improve user intent recognition.

2.2 Uncertainty Estimation in Task-Oriented Systems

Uncertainty estimation is vital for task-oriented systems as it influences decision-making. Bayesian models and confidence calibration techniques, such as MC Dropout for approximate Bayesian inference [11] and Temperature Scaling for neural calibration [14, 23],

Table 1: Query Types, Definitions, and Examples

Query Type	Definition	Example Queries
POI (Point of Interest)	Focus on searching for or recommending specific locations.	- "Find the nearest Starbucks." - "What are some popular tourist attractions in Paris?"
Navi (Navigation)	Centers on route planning or navigation tasks.	- "How do I get from home to the airport?" - "What's the fastest route to Central Park?"
Taxi	Deals with ride-hailing or transportation services.	- "Book a taxi to the nearest hospital." - "How much is a ride from Times Square to JFK Airport?"
Transit	Related to public transit information, including metro and bus schedules.	- "Which metro line connects the train station to the airport?" - "What are the bus schedules from downtown to the suburbs?"

are widely used, while entropy-based measures are common in open-domain QA and machine translation [10, 21].

In task-oriented settings, uncertainty-aware clarification has been explored through methods like *CLAM*, which employs selective prompts for ambiguity resolution [20], and *CLARA*, which combines uncertainty quantification with zero-shot classification for improved intent understanding [18, 26, 29].

Our framework integrates uncertainty estimation with adaptive beam search to dynamically optimize response accuracy and efficiency in navigation contexts.

2.3 Adaptive Search Algorithms and Uncertainty Management

Adaptive search improves decision-making under uncertainty by using tree-decoding for clarifications [27], round-trip verification for multi-turn reliability [12], and tree structures for exploration-exploitation balance [40]. *LinearFold* shows beam pruning boosts efficiency without loss of accuracy [17], while *Ask-before-Plan* employs multi-agent collaboration for uncertainty-aware planning [42]. Earlier constrained search strategies [15, 16] also inform our dynamic beam width adjustment for stable, low-latency navigation.

2.4 Evaluation of Uncertainty Handling and Adaptability

Robustness under uncertainty is critical for intelligent systems, with benchmarks covering legal retrieval and robotic planning [24, 39]. Domain-specific methods like *LeClari* use event schemas for clarifications, while others target uncertainty estimation in high-stakes tasks [30].

We evaluate our framework on Baidu Maps and SaGC, showing improved uncertainty management and cross-domain adaptability for real-time, map-based queries.

2.5 Comparison with Related Work

Our work improves on *CLARA* and *CLAM* [20, 29], addressing their limitations while maintaining a training-free design.

- *CLAM* relies on simple prompts to detect ambiguity but struggles with domain-specific cases like robotic tasks and navigation.
- *CLARA* uses greedy sequence generation, which often leads to suboptimal task execution in multi-step reasoning scenarios.

Our approach avoids additional training, leveraging beam-pruning for better search space exploration. Combined with task-specific prompts and uncertainty estimation, it improves accuracy, adaptability, and reliability across diverse domains.

3 Method

In this section, we introduce the proposed uncertainty-aware planning framework for generating clarification questions, to overcome key limitations in intent recognition, uncertainty management, and task planning in intelligent map agent systems. The framework operates in three main stages: intent recognition, uncertainty estimation with dynamic beam pruning, and clarification question generation. Upon receiving a user query related to map services, the system initially uses a zero-shot prompt to determine the query's relevance. For queries identified as map-related, the system classifies them into predefined task categories, estimates uncertainty, and generates customized clarification questions. This process improves the accuracy and reliability of intent interpretation.

3.1 Preliminaries

The clarification question generation framework begins by identifying whether a given user query pertains to map-related tasks, using a zero-shot prompt to perform this initial classification (templates released in our open-source repository (`prompts`)). For non-map-related queries, the system directly invokes the general LLM, a.k.a `ask_llm` function to generate an appropriate response. To provide a clear understanding of the query types, Table 1 summarizes their definitions and illustrative examples.

Each query category has a dedicated prompt template, which allows the sequence generation and uncertainty estimation to be tailored to the specific task. The complete templates and illustrative examples are released in our open-source repository (see `prompts`). This structured classification provides a reliable basis for generating relevant clarification questions and improves both intent recognition and task relevance.

Following categorization, the system performs uncertainty estimation with beam pruning to ensure precision and stability in sequence generation. When a high level of uncertainty is detected, the system proactively generates clarification questions, guiding users to clarify their intent and thereby enhancing overall user experience. Our overall framework is illustrated in Figure 2.

3.2 Task Definition

We define the problem as an uncertainty-aware sequence generation task for interactive map agents. Given a user query x_g describing the user's intent (e.g., "Find the fastest route to the airport")

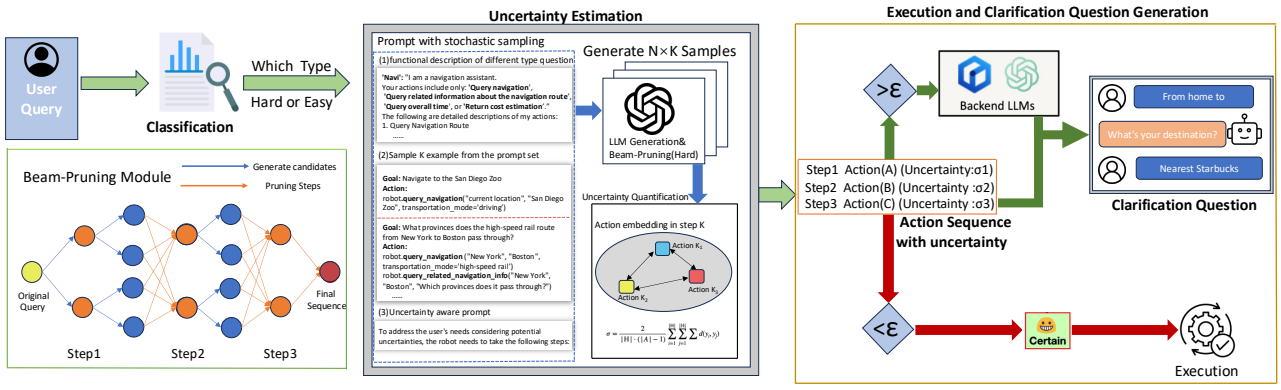

Figure 2: Framework overview of the proposed approach. The system classifies queries by difficulty and applies beam-pruned sampling with dynamic uncertainty-aware pruning for complex cases. If the overall uncertainty exceeds threshold ϵ, a clarification question is generated; otherwise, the action sequence is executed directly to fulfill the user's intent.

and the task context x_s (e.g., task type = navigation), the goal of the agent is to produce either:

- a valid action sequence $Y = [y_1, y_2, \ldots, y_T]$ that accomplishes the intended task, or
- a clarification question q_c if the user's query is ambiguous.

To achieve this, the agent first generates candidate action sequences for the given query and estimates the uncertainty associated with these actions. Based on the overall uncertainty, the agent dynamically decides whether to execute the action sequence directly or generate a clarification question to resolve ambiguity.

This formulation enables the agent to proactively manage ambiguous queries in interactive scenarios, improving both response accuracy and user experience (see Table 2 for a summary of symbols).

Table 2: Summary of symbols used in the task definition

Symbol	Description
x_g	User query
x_s	Task context
y_t	Action at decoding step t
Y	Full action sequence
q_c	Clarification question

3.3 Agent Paradigm

Our framework follows an API-based agent paradigm, where the system interacts with structured service endpoints (e.g., routing APIs, POI search) instead of performing screen-based manipulations. This architecture is well aligned with real-world applications like Baidu Maps, which support intent resolution via backend APIs rather than GUI elements.

In contrast to GUI agents (e.g., Meta-GUI [33]) that operate through visual grounding and pixel-level control to execute tasks, our system translates user intent into sequences of API calls using structured prompt templates. This approach improves reliability, scalability, and interpretability, making it more suitable for production environments.

3.4 Query Classification and Command Dispatch

3.4.1 Query Type Classification. For queries identified as map-related, the system further classifies them into four task categories—POI, Navigation, Taxi (ride-hailing), and Transit—using a lightweight LLM with a zero-shot prompt, enabling fast response and reliable categorization. The exact templates and examples are provided in our open-source repository (see `prompts`). This step ensures that subsequent task planning is aligned with the semantics and structure of each query type.

3.4.2 Command Dispatch Based on Complexity. To further optimize response efficiency, we introduce a *command dispatch* mechanism that routes queries based on their assessed complexity. In brief, simple map queries bypass expensive decoding operations, while complex or ambiguous ones undergo full uncertainty estimation and beam pruning. This mechanism effectively balances efficiency and accuracy, ensuring low-latency handling for straightforward queries and robust processing for challenging cases. Implementation details are provided in Appendix G.

3.5 Beam Pruning with Uncertainty Estimation

To generate accurate sequences for each task category, we propose an enhanced beam-pruning algorithm that integrates uncertainty estimation. This algorithm dynamically adjusts beam width based on real-time uncertainty, effectively controlling error accumulation and improving search efficiency.

3.5.1 Uncertainty Estimation. We design an uncertainty-aware planning approach that quantifies uncertainty at each step of sequence generation. To guide the model, we construct task-specific few-shot prompts that define the valid action and state spaces for each query type. The full prompt specifications for the four task categories—Navigation, Ride-Hailing, Public Transit, and POI—are released in our open-source repository (see `prompts`). Each template constrains the action set and provides goal–action exemplars tailored to its domain, enabling consistent and reliable plan generation.

At each decoding step, the model is prompted with: *"From this, predict the next action considering the role of the robot and the ambiguity of the goal."* The model generates H candidate actions, from which the most frequent is selected. This selected action y_t is then appended to the prompt for the next round, forming an **iterative refinement prompt**—a growing history of previously selected actions that informs future predictions.

Example (Simplified):

- Round 1 Prompt: *"Goal: Find the nearest coffee shop. From this, predict the next action."*
- Model Output: `robot.search_poi(near_current_location, 'Coffee Shop', '', 3)`
- Round 2 Prompt: *"Goal: Find the nearest coffee shop. Previous actions: [search_poi(...)] From this, predict the next action."*

This iterative process continues until a full sequence is constructed or a clarification is triggered due to high uncertainty.

Uncertainty estimation occurs at each step through k-sampling. Specifically, for a given decoding prompt, the model samples k candidate actions using top-p sampling. These candidates reflect plausible alternatives under the same context and are used to assess the model's confidence. A high degree of variation among them indicates higher uncertainty.

Each sampled action is then encoded into an embedding vector using `zh_core_web_lg` for Chinese queries [8] and `en_core_web_lg` for English queries [7]. A pairwise distance matrix is computed to quantify the divergence among the k outputs. The step-wise uncertainty score σ is calculated as the average pairwise distance between embeddings, capturing semantic variability across samples.

Our formulation builds on prior work such as CLARA [29], which models uncertainty using pairwise embedding distance to capture semantic variability.Formally, σ is computed as:

$$\sigma = \frac{2}{A} \sum_{i,j} \|g(y_i) - g(y_j)\|^2,$$

where $y_i = f(x_g, x_s, c_i)$, $A = 2H \cdot (H-1)$, and $c_i \sim p(C)$ denotes a sampled latent context. The function $g(\cdot)$ maps each output to an embedding space to capture its essential semantic content. This distance matrix effectively measures uncertainty at each step, with greater variance indicating higher uncertainty.

To compute an overall uncertainty score for the query, we average the step-wise uncertainty scores across all rounds of sequence generation. Experimental results validate that higher overall uncertainty scores correlate with more complex or ambiguous queries, supporting our hypothesis that unstable generation reflects high query uncertainty. This method thus provides both quantifiable metrics and interpretability, enabling the system to identify queries that may require clarification and adjust responses accordingly. We validated the correlation between ambiguity and uncertainty empirically (Appendix D), observing a strong alignment between average uncertainty scores and manually labeled ambiguity levels.

3.5.2 Adaptive Beam Size Adjustment. There is a key issue during uncertainty estimation, an incorrect action chosen early in a sequence can distort the model's uncertainty score, leading to compounding errors. This problem is particularly evident when the model begins with an incorrect action, which skews subsequent predictions and increases randomness in the following steps due to high accumulated uncertainty.

For instance, in the SaGC dataset, when the intended sequence for "*Cook and serve bacon and toast on a plate*" is `grab(bacon)`, `heat(bacon)`, `grab(bread)`, `heat(bread)`, `plate(bacon)`, `plate(bread)`, a greedy search can mistakenly produce an initial sequence such as `grab(bacon)`, `grab(bread)`. This deviation causes subsequent steps to diverge from the intended goal, illustrating the risk of early missteps in constrained search spaces.

To mitigate this, we implement beam search, which expands the search space by considering multiple sequence paths. However, even with a moderate beam size (e.g., 2), the computational complexity grows exponentially with sequence length, leading to increased response times.

To address the trade-off between search efficiency and accuracy, we propose a beam pruning algorithm that combines the focus of greedy search with the exploratory power of beam search. This approach retains only the top candidate paths at each step, limiting computational costs while preserving high-quality outcomes. Building on this pruning approach, we further enhance beam search by adjusting the beam size dynamically according to the uncertainty at each step. This adjustment allows for a balanced exploration-exploitation strategy, dynamically scaling the search width based on real-time uncertainty levels. The adaptive beam width adjustment is implemented as follows:

- When the uncertainty score σ is high, the beam width B expands toward $B_{\max}$, increasing the number of candidate paths to capture more potential actions.
- When the uncertainty score σ is low, B contracts toward $B_{\min}$, focusing on fewer, high-confidence paths and thereby improving response efficiency.

The uncertainty score σ is computed as the average pairwise embedding variance among sampled candidate actions at each decoding step. This embedding-level variance intuitively reflects the model's predictive uncertainty. Specifically, we define:

$$\sigma = \bar{\sigma}_t,$$

where $\bar{\sigma}_t$ represents the mean variance of action embeddings at step t. The score σ is then evaluated against a predefined threshold ϵ to determine whether to dynamically expand or shrink the beam width.

To calibrate the threshold ϵ, we use a validation set of 100 real-world map queries and select the value that yields the highest F1-score in detecting ambiguous inputs.

$$B = \begin{cases} B_{\max} & \text{if } \sigma \geq \epsilon \\ B_{\min} & \text{if } \sigma < \epsilon \end{cases}$$

By dynamically adjusting beam width, the model effectively balances the need for exploration under high uncertainty with the efficiency of a narrowed search when uncertainty is low. This approach optimizes accuracy and computational efficiency, ensuring that high-uncertainty steps are met with sufficient flexibility, while stable steps are processed more quickly. The entire process is detailed in Algorithm 1.

Algorithm 1: The executive procedure of adaptive beam width adjustment strategy.

Input: User map query
Result: Final path or clarification question
Intent Recognition and Classification: Determine if the query is map-related;
if *query is not map-related* **then**
  Call `ask_llm` and return response;
else
  Classify query into task category (POI, Navi, Taxi, Transit);
  Initialize beam width $B = B_{\min}$;
  foreach *step in path generation* **do**
    Calculate uncertainty score σ;
    if $\sigma \geq \epsilon$ **then**
      Set $B = B_{\max}$;
    else
      Set $B = B_{\min}$;
    Generate and filter candidate paths;
  if *Overall uncertainty* $\sigma \geq \epsilon$ **then**
    Generate clarification question;
  Output: Final path or clarification question;

3.6 Clarification Question Generation

When the system identifies that a query's uncertainty exceeds a set threshold, it flags the query as needing clarification. This flagging step enables the system to prompt the user for more specific information, ultimately improving intent recognition.

To generate effective clarification questions, the framework uses an LLM guided by a zero-shot prompt (the template is provided in our open-source repository; see `prompts`). The prompt instructs the model to detect ambiguity or missing information in the user query and to produce an appropriate clarification question. With this explicit guidance, the system can dynamically formulate targeted clarifications without additional supervision or examples.

In essence, this process ensures that each clarification question is tailored to the unique context and uncertainty of the original query, encouraging users to provide additional information where needed. By integrating a clarification stage, the framework proactively resolves ambiguities, enhancing the accuracy and relevance of the system's responses.

4 Experiments

4.1 Experiment Design

We evaluate our system regarding the following questions:

- **Q1: How does our framework compare to baseline models in managing map-related queries?** This question evaluates the framework's overall effectiveness in handling complex and uncertain queries. We compare its performance in accuracy and uncertainty management against baseline models CLARA and CLAM [20, 29].
- **Q2: What is the contribution of individual modules to performance and efficiency?** To assess the roles of beam-pruning, instruction dispatch, and dynamic pruning in enhancing the framework. Ablation studies analyze the impact of these modules on accuracy and response time.
- **Q3: How adaptable is the framework across datasets and backbone models without additional training?** This question examines the generalizability of the framework by testing it on the Baidu Maps and SaGC datasets and evaluating its performance across various LLM backbones.

Each question is addressed through targeted experimental setups, with corresponding results and analyses presented, including sensitivity analysis (Appendix E) and implementation details (Appendix F).

4.2 Datasets

To evaluate the effectiveness and generalizability of our framework, we use two benchmark datasets: Baidu Maps and SaGC.

Baidu Maps Dataset: We collected 2,000 real-world user queries via Baidu Maps' online service, covering diverse map-related scenarios. The queries can be categorized into five task intents: POI (Point of Interest, 1230 queries), Navi (Navigation, 554 queries), Transit (241 queries), Taxi (61 queries), and Others (44 queries).

The first four categories represent the primary action space for map-related tasks, with POI being the most frequent ones. The "Others" category includes non-map-related queries or those with unrecognizable intents. This distribution ensures the dataset reflects real-world complexity and variability in map-based interactions.

SaGC Dataset: The SaGC (Situational Awareness for Goal Classification in Robotic Tasks) dataset evaluates situational uncertainty in robotic tasks. It includes 5,006 goal scene pairs annotated with one of three uncertainty levels: certain (1,749 pairs), ambiguous (1,560 pairs), or infeasible (1,697 pairs). Spanning 15 scenarios across three robotic task domains – cooking, cleaning, and massaging – this dataset assesses the framework's ability to generalize across distinct domains and uncertainty levels. The annotations were derived using LLMs, as established in prior work [29].

4.3 Evaluation Metrics

We adopt standard classification (Precision, Recall, F1-score, Accuracy) and efficiency (average response time) metrics to evaluate our framework. Detailed definitions and computation formulas are provided in Appendix H.

4.3.1 Q1: Performance Comparison with Baseline Models on Baidu Maps Dataset. To address Q1, we evaluate our framework's performance on the Baidu Maps dataset, comparing it against two baseline models: CLARA and CLAM. These baselines are selected due to their training-free nature, which enables direct application to datasets lacking annotated training data, such as our Baidu Maps dataset. This characteristic makes them particularly relevant for comparison, as supervised fine-tuning approaches are infeasible in such scenarios.

The evaluation focuses on four key metrics: Recall, Precision, F1-score, and Accuracy. These metrics provide a comprehensive assessment of the models' ability to classify queries accurately and manage complexity. By highlighting these metrics, we aim to demonstrate the advantages of our framework not only in query classification but also in effectively handling complex and ambiguous user inputs under the constraints of a training-free setting.

As shown in Table 3, our framework consistently outperforms the baseline models across all metrics. Specifically, it achieves the highest Recall (**63.2%**), Precision (**81.3%**) F1-score (**71.1%**), and Accuracy (**97.1%**), significantly exceeding the scores of CLARA and CLAM. These results demonstrate the framework's ability to deliver accurate and reliable outputs, even for complex and ambiguous queries. We also evaluate recent clarification methods under few-shot prompting (see Appendix A, Table 10). While these models reach high recall, they suffer from over-clarification, highlighting our model's more balanced precision.

Table 3: Performance comparison on Baidu Maps dataset across key metrics.

Model	Recall (%)	Precision (%)	F1-score (%)	Acc. (%)
Ours	**63.2**	**81.3**	**71.1**	**97.1**
CLARA	36.8	50.4	42.5	92.5
CLAM	52.5	7.8	13.6	76.1

These findings highlight our framework's effectiveness in managing complex queries, achieving high classification performance and robustness in real-world scenarios.

4.3.2 Q2: Module-Wise Impact on Performance and Efficiency. To address **Q2**, we conduct ablation studies and parameter sensitivity experiments to evaluate the impact of key modules—beam pruning, instruction dispatch, and dynamic pruning—on both performance and response time. All experiments in this section use the same inference pipeline as our deployed Baidu Maps system, ensuring consistency and reproducibility [38]. Additionally, we assess the influence of varying beam width values to identify the optimal trade-off configuration for real-world applications.

Table 4: Beam size vs. Accuracy and Average Response Time.

Beam Size	Accuracy (%)	Avg. Time per Query (s)
1	92.5	7.85
2	**97.1**	12.16
3	97.2	15.86

The results of the parameter sensitivity analysis (Table 4) show that increasing the beam width improves accuracy, but the gains saturate beyond Beam=2. Meanwhile, response time grows linearly with beam size, suggesting Beam=2 as the best balance between performance and latency.

Table 5: Module-wise impact on Accuracy and Response Time.

Configuration	Accuracy (%) / Time (s)
CLAM	76.1 / **1.1**
No Beam-pruning	92.5 / 7.85
No Instruction Dispatch	**97.1** / 12.16
No Dynamic Pruning	95.9 / 10.64
All Modules	94.6 / 8.49

The configuration without Instruction Dispatch achieves the highest accuracy (97.1%) but at the cost of significantly increased latency (12.16s). In contrast, our deployed configuration ("All Modules") slightly compromises accuracy (94.6%) while substantially reducing latency (8.49s), achieving a favorable balance between quality and runtime. We attribute this small accuracy drop to the Instruction Dispatch module, which bypasses beam search for low-complexity queries, thereby reducing computational overhead. This design choice is further supported by our user preference study in Appendix C, where "All Modules" received the highest number of votes, confirming its effectiveness for real-world deployment.

4.3.3 Q3: Cross-Domain Adaptability and Backbone Model Consistency. To address **Q3**, we evaluate our framework's adaptability across different datasets and backbone models. Due to time constraints, we focused on mainstream English models for the SaGC dataset and mainstream Chinese models for the Baidu Maps dataset, ensuring a fair comparison in each linguistic context.

Table 6 presents precision scores on the SaGC dataset, highlighting our framework's ability to generalize beyond the Baidu Maps dataset. Our model consistently outperforms the CLARA and CLAM baselines, demonstrating strong cross-domain adaptability.

Table 7 shows Baidu Maps results with different backbone models. Precision stays high across backbones, highlighting the framework's stable, training-free adaptability.

Table 6: Accuracy comparison on SaGC dataset (cross-domain evaluation).

Model	CLAM	CLARA	Ours
Gemma2-9b-it [34]	55.24	69.87	**72.15**
LLAMA3-8B-Instruct [1]	52.56	65.70	**69.39**
Qwen2-7B-Instruct [38]	49.04	64.89	**66.31**

Table 7: Accuracy comparison across backbones on Baidu Maps dataset (training-free adaptability).

Model	CLAM	CLARA	Ours
Qwen2-7B-Instruct	76.1	92.5	**97.1**
GLM4-9B-chat [13]	72.1	88.9	**95.2**

These results confirm the effectiveness of our framework in both cross-domain settings (SaGC) and across multiple backbones (Baidu Maps). The strong performance on SaGC demonstrates its robustness in handling ambiguous queries in a different domain, while the consistent results on various LLMs underscore its versatility and compatibility with multiple architectures.

Overall, these findings validate our framework as a practical and scalable training-free solution for diverse real-world applications. Its adaptability across datasets and backbone models ensures seamless integration into intelligent systems for both English and Chinese contexts.

4.3.4 Pairwise Evaluation of Clarification Question Quality. We conduct a pairwise Good-Same-Bad (GSB) evaluation to assess the quality of clarification question generation against CLAM and

CLARA. For both the Baidu Maps and SaGC datasets, we extracted all queries requiring clarification and generated clarification questions using our framework and the baselines. The quality of these generated questions was then evaluated using Baidu's ERNIE Bot 4.0 model [2]. Human evaluation results (Appendix B) confirm that our clarification questions are preferred over CLAM and CLARA in over 60 percent of cases.

Evaluation Criteria:

- **G (Good)**: Our method is significantly better.
- **S (Same)**: Both methods are comparable.
- **B (Bad)**: The baseline is significantly better.

Results: Table 8 presents the GSB evaluation results, demonstrating that our framework consistently achieves higher Good (G) rates while minimizing Bad (B) ratings across both datasets.

Table 8: GSB Evaluation of Clarification Question Quality

Dataset	Comparison	G (%)	S (%)	B (%)
Baidu Maps	Ours vs. CLAM	**65.5**	33.4	1.0
	Ours vs. CLARA	**67.5**	31.4	1.0
SaGC	Ours vs. CLAM	**40.1**	56.3	3.6
	Ours vs. CLARA	**34.0**	58.7	7.3

Conclusion: Our method consistently outperforms CLAM and CLARA across both datasets, generating higher-quality clarification questions with minimal Bad (B) ratings.

4.4 Online Test: Performance Comparison with Baidu Maps Online Model

Our system is built with the Qwen2-7B-Instruct as the backbone. The inference procedure is deployed as a plugin-based module seamlessly integrated into the existing Baidu Maps' service architecture. It provides online inference through a RESTful API and leverages query caching to improve response speed. In addition, the module supports independent microservice deployment, enabling low-cost and stable clarification question generation without requiring modifications to the core map model.

To evaluate the practical effectiveness of our framework in clarifying ambiguous user intent, we deployed it alongside the Baidu Maps online model and compared their performance using two key classification metrics: Precision and Accuracy. The Baidu Maps online model, a supervised fine-tuned (SFT) system trained on a large volume of annotated data, is specifically designed to handle diverse map-related queries with high accuracy.

The results, shown in Figure 3, highlight that our framework outperforms the Baidu Maps model in both Precision and Accuracy. This indicates its effectiveness in handling ambiguous intents and producing reliable results in map service scenarios. Additionally, our framework achieves this performance with a training-free design, eliminating the need for costly annotated data and extensive training processes, making it a highly efficient and scalable solution.

To further assess the quality of generated clarification questions, we conducted a pairwise GSB (Good–Same–Bad) evaluation comparing our framework and the Baidu Maps online model on 3,281 ambiguous queries collected from real-world logs. The evaluation results are summarized in Table 9.

Table 9: Pairwise GSB evaluation comparing clarification questions generated by our framework and the Baidu Maps online model.

Comparison	Good (G)	Same (S)	Bad (B)
Ours vs. Online Model	2458 (74.92%)	788 (24.02%)	35 (1.07%)

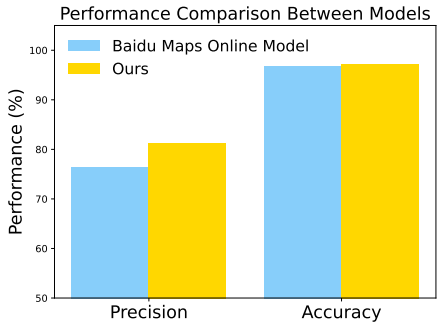

Figure 3: Performance comparisons between our framework and the Baidu Maps online model.

These GSB results demonstrate that our framework produces superior clarification questions in the vast majority of cases, with 74.92% rated as better than the online model and only 1.07% rated as worse. This indicates not only higher precision in understanding ambiguous user intents but also practical effectiveness in delivering targeted, context-aware clarifications in online scenarios. Such consistent performance confirms the scalability and robustness of our approach when integrated into real-world map services.

4.5 Discussion

Our experiments highlight the benefits of integrating beam search into uncertainty estimation for improved planning reliability and efficiency. We provide an extended analysis of dataset-specific implications and parameter tuning insights in Appendix I.

5 Conclusion

This paper introduces an uncertainty-aware task-planning framework for intent disambiguation, addressing the limitations of traditional methods in handling uncertain inputs for LLM agents, such as those used in Baidu Maps. By incorporating uncertainty quantification, independent sequence state management, and adaptive pruning strategies, the framework significantly improves accuracy, response efficiency, and task success rates. Experimental evaluations on the SaGC and Baidu Maps datasets demonstrate SOTA performance, with response times suitable for real-world applications. In addition, the framework also exhibits strong adaptability, generalizing seamlessly across datasets and backbone models without requiring additional training. Its training-free design enables rapid integration into diverse, intelligent systems.

Future work will extend the framework to additional domains and leverage richer context to support more reliable planning. Meanwhile, we will optimize uncertainty quantification to improve robustness and adaptability under dynamic conditions.

Acknowledgments

This work was done while the first author (Deqiang Huang) was an intern at Baidu Inc.This work was supported in part by the National Science and Technology Major Project of China (No. 2023ZD0121104) and by the National Natural Science Foundation of China (Grant No. 62222213,No. 92370204).

References

[1] AI@Meta. 2024. Llama 3 Model Card. (2024). https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[2] Baidu AI Cloud. 2023. Qianfan Large Model Platform - ERNIE Bot 4.0. Available at: https://cloud.baidu.com/product-s/qianfan_home. Accessed: Feb 9, 2025.

[3] Maximillian Chen, Ruoxi Sun, Sercan Ö Arık, and Tomas Pfister. 2024. Learning to Clarify: Multi-turn Conversations with Action-Based Contrastive Self-Training. *arXiv preprint arXiv:2406.00222* (2024).

[4] Yizhou Chi, Jessy Lin, Kevin Lin, and Dan Klein. 2024. CLARINET: Augmenting Language Models to Ask Clarification Questions for Retrieval. *arXiv preprint arXiv:2405.15784* (2024).

[5] Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. 2023. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. *arXiv preprint arXiv:2305.13626* (2023).

[6] Kaustubh D Dhole. 2020. Resolving intent ambiguities by retrieving discriminative clarifying questions. *arXiv preprint arXiv:2008.07559* (2020).

[7] Explosion AI. 2023. en_core_web_lg: English large spaCy model. https://spacy.io/models/en#en_core_web_lg

[8] Explosion AI. 2023. zh_core_web_lg: Chinese large spaCy model. https://spacy.io/models/zh#zh_core_web_lg

[9] Ashley Fernandez and Swaraj Dube. 2023. Core Building Blocks: Next Gen Geo Spatial GPT Application. *arXiv preprint arXiv:2310.11029* (2023).

[10] Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics* 8 (2020), 539–555.

[11] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.

[12] Yifan Gao, Henghui Zhu, Patrick Ng, Cicero Nogueira dos Santos, Zhiguo Wang, Feng Nan, Dejiao Zhang, Ramesh Nallapati, Andrew O Arnold, and Bing Xiang. 2020. Answering ambiguous questions through generative evidence fusion and round-trip prediction. *arXiv preprint arXiv:2011.13137* (2020).

[13] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793* (2024).

[14] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

[15] Liang Huang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 142–151.

[16] Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 1077–1086.

[17] Liang Huang, He Zhang, Dezhong Deng, Kai Zhao, Kaibo Liu, David A Hendrix, and David H Mathews. 2019. LinearFold: linear-time approximate RNA folding by 5'-to-3'dynamic programming and beam search. *Bioinformatics* 35, 14 (2019), i295–i304.

[18] 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*. PMLR, 9118–9147.

[19] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. *arXiv preprint arXiv:2310.14696* (2023).

[20] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769* (2022).

[21] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664* (2023).

[22] Vaibhav Kumar et al. 2020. ClarQ: A large-scale and diverse dataset for clarification question generation. *arXiv preprint arXiv:2006.05986* (2020).

[23] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* 30 (2017).

[24] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 9493–9500.

[25] Bulou Liu, Yiran Hu, Qingyao Ai, Yiqun Liu, Yueyue Wu, Chenliang Li, and Weixing Shen. 2023. Leveraging Event Schema to Ask Clarifying Questions for Conversational Legal Case Retrieval. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1513–1522.

[26] Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650* (2020).

[27] Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. 2021. Joint passage ranking for diverse multi-answer retrieval. *arXiv preprint arXiv:2104.08445* (2021).

[28] Fangwen Mu, Lin Shi, Song Wang, Zhuohao Yu, Binquan Zhang, ChenXue Wang, Shichao Liu, and Qing Wang. 2024. ClarifyGPT: A Framework for Enhancing LLM-Based Code Generation via Requirements Clarification. *Proceedings of the ACM on Software Engineering* 1, FSE (2024), 2332–2354.

[29] Jeongeun Park, Seungwon Lim, Joonhyung Lee, Sangbeom Park, Minsuk Chang, Youngjae Yu, and Sungjoon Choi. 2023. CLARA: classifying and disambiguating user commands for reliable interactive robotic agents. *IEEE Robotics and Automation Letters* (2023).

[30] Pradip Pramanick, Chayan Sarkar, Sayan Paul, Ruddra dev Roychoudhury, and Brojeshwar Bhowmick. 2022. Doro: Disambiguation of referred object for embodied agents. *IEEE Robotics and Automation Letters* 7, 4 (2022), 10826–10833.

[31] Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. *arXiv preprint arXiv:1805.04655* (2018).

[32] Anmol Singhal, Chirag Jain, Preethu Rose Anish, Arkajyoti Chakraborty, and Smita Ghaisas. 2024. Generating Clarification Questions for Disambiguating Contracts. *arXiv preprint arXiv:2403.08053* (2024).

[33] Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. 2022. Meta-gui: Towards multi-modal conversational agents on mobile gui. *arXiv preprint arXiv:2205.11029* (2022).

[34] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118* (2024).

[35] Alberto Testoni and Raquel Fernández. 2024. Asking the right question at the right time: Human and model uncertainty guidance to ask clarification questions. *arXiv preprint arXiv:2402.06509* (2024).

[36] Eren Unlu. 2023. Chatmap: Large Language Model Interaction with Cartographic Data. *arXiv preprint arXiv:2310.01429* (2023).

[37] Jian Wang and Wenjie Li. 2021. Template-guided clarifying question generation for web search clarification. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 3468–3472.

[38] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671* (2024).

[39] Yang Yang, Xibai Lou, and Changhyun Choi. 2022. Interactive robotic grasping with attribute-guided disambiguation. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 8914–8920.

[40] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems* 36 (2024).

[41] Michael JQ Zhang, W Bradley Knox, and Eunsol Choi. 2024. Modeling future conversation turns to teach llms to ask clarifying questions. *arXiv preprint arXiv:2410.13788* (2024).

[42] Xuan Zhang, Yang Deng, Zifeng Ren, See-Kiong Ng, and Tat-Seng Chua. 2024. Ask-before-Plan: Proactive Language Agents for Real-World Planning. *arXiv preprint arXiv:2406.12639* (2024).

A Additional Baseline Comparison Results

To further evaluate our approach under training-free constraints, we implemented few-shot prompting versions of several recent clarification methods. As shown in Table 10, these models achieve high recall but suffer from low precision due to over-clarification. Our method achieves a more balanced performance, particularly in precision and overall accuracy.

Table 10: Performance comparison of few-shot adaptations of recent clarification methods vs. our approach.

Method	Recall (%)	Precision (%)	F1 (%)	Accuracy (%)
Conrap [32]	94.8	6.3	11.8	22.4
QDrawer [35]	78.6	5.7	10.6	27.9
CEP [42]	92.3	9.5	17.2	51.7
Ours	63.2	**81.3**	**71.1**	**97.1**

B Human Evaluation: Clarification Question Quality

To assess the quality of generated clarification questions, we conducted a pairwise human evaluation on 200 ambiguous queries. Each question was evaluated by 20 participants, comparing our method with CLARA and CLAM. As shown in Table 11, our method was preferred by a significant margin.

Table 11: Human preference on clarification question quality (200 ambiguous queries, 20 participants).

Comparison	Ours Preferred (%)	Baseline Preferred (%)
Ours vs CLARA	66.5	33.5
Ours vs CLAM	61.0	39.0

C Human Evaluation: System Configuration Preference

We conducted a second human evaluation to measure user preference over different system configurations, each reflecting a different trade-off between performance and efficiency. Participants were presented with paired outputs and asked which configuration they preferred. Table 12 summarizes the results.

Table 12: Participant preference across system configurations (200 votes).

System Configuration	Votes Received
All Modules	65
No Instruction Dispatch	40
No Beam Pruning	35
No Dynamic Pruning	30
No Uncertainty Estimation	30

D Statistical Validation of Uncertainty and Clarification Quality

We conducted two supporting experiments to validate key claims in our method. First, we manually labeled 200 queries by ambiguity level and computed their average uncertainty scores. As shown in Table 13, uncertainty strongly correlates with query complexity.

Table 13: Average uncertainty score by ambiguity level (200 labeled queries).

Ambiguity Level	Avg. Uncertainty
Clear	2.01
Slightly Ambiguous	4.13
Most Ambiguous	5.49

Second, we performed a two-sample t-test comparing clarification quality between our method and CLARA. The test yields $t = 8.78$, $p < 10^{-16}$, confirming a statistically significant improvement.

E Uncertainty Threshold Sensitivity Analysis

To evaluate the robustness of our method to the choice of uncertainty threshold ϵ, we performed a sensitivity analysis by varying U over a broad range. As shown in Table 14, the system achieves the best F1-Score when $\epsilon = 3.9$, and remains stable across nearby values.

Table 14: F1-Score under different uncertainty thresholds.

Threshold Value	F1-Score
4.5	0.440
4.2	0.573
3.9 (default)	**0.711**
3.6	0.642
3.3	0.605

F Implementation Details

Backbone Model: Qwen2-7B-Instruct (default decoding parameters).
Uncertainty Threshold: $\epsilon = 3.9$ (selected via grid search).
Sampling Parameters: $H = 3$ samples per decoding step, sequence depth = 3.
Beam Width: Beam size = 2 for default settings.

G Command Dispatch Implementation Details

To fully describe the dispatch mechanism in Section 3.4.2, we provide the detailed classification process:

- **Few-shot Examples for Task Relevance:** We use lightweight LLM prompts to distinguish map-related queries (e.g., POI, Navigation) from non-map queries.

- **Complexity Assessment:** Queries are categorized into simple (e.g., direct POI lookup) or complex (e.g., ambiguous routes, multi-criteria POI search) using handcrafted rules and few-shot exemplars.
- **Routing Logic:**
  - Non-map queries are handled by `ask_llm` directly.
  - Simple map queries undergo single-pass uncertainty estimation without beam pruning.
  - Complex queries use full uncertainty estimation with dynamic beam pruning.

This design improves computational efficiency by avoiding redundant decoding for low-ambiguity inputs while allocating resources to complex cases.

H Evaluation Metrics

To comprehensively evaluate the performance of our framework, we employ two core metrics: classification metrics (Precision, Recall, F1-score, and Accuracy) and response time. These metrics jointly capture the model's accuracy and efficiency across tasks.

- **Classification Metrics.** Standard classification metrics are employed to measure the framework's effectiveness in query categorization. These include Precision, Recall, F1-score, and Accuracy, which collectively evaluate the model's predictive performance across different aspects.
  Note: In our context, *Precision* also serves as a proxy for **intervention accuracy**, since it reflects the proportion of clarification triggers that were indeed necessary. A high Precision thus indicates fewer false clarifications (over-clarification), aligning with the goal of balancing under- and over-solicitation.
- **Response Time.** Response time measures the average duration required to generate a response, reflecting the framework's operational efficiency. It is computed as:

$$T_{\text{avg}} = \frac{\sum_{i=1}^{N} T_i}{N}$$

where T_i represents the response time for the i-th query, and N is the total number of queries. Lower T_{avg} values indicate a more efficient framework.

I Extended Discussion

The experimental results demonstrate that integrating beam search algorithms into uncertainty estimation provides substantial advantages in task planning and execution reliability. By leveraging uncertainty quantification, our approach dynamically adjusts pruning thresholds and independently manages sequence states, effectively mitigating the impact of accumulated uncertainty. This mechanism enhances both task success rates and planning efficiency, particularly in scenarios involving complex or ambiguous queries. Additionally, parameter sensitivity analysis highlights that a beam width of 2 achieves the optimal balance between accuracy and response time, reinforcing the importance of carefully tuning parameters for real-world applications.

The performance comparison between the SaGC and Baidu Maps datasets highlights the importance of task-specific adaptations. On SaGC, we observed relatively smaller improvements compared to Baidu Maps. This can be attributed to the experimental setup, where we directly applied the baseline method from CLARA without significant modifications to its prompt design or state space definitions. Our focus for SaGC was to validate the integration and general applicability of beam-pruning within the existing framework. As a result, the improvements were constrained by the limitations of the unmodified baseline setup.

In contrast, the Baidu Maps dataset benefited from carefully designed prompts and specific adaptations to address unique challenges, such as ambiguous intents and incomplete information. These refinements enable the framework to fully leverage beam-pruning and other enhancements, resulting in substantial performance gains. This comparison underscores the importance of tailoring prompts and state space definitions to the specific requirements of each task domain. Future work could explore tailored prompt designs and task-specific adaptations to better address the domain-specific challenges presented by the SaGC dataset.