Algorithm 1 Data Construction Pipeline

```
Require: target $A$, distractor $B$
$F_A \leftarrow$ attributes of $A$;   $F_B \leftarrow$ attributes of $B$
Build ambiguous $Q$ from $F_A \cap F_B$
Add context $C$ from $F_A \setminus Q$
Validate $(Q, C)$:
while not finished do
    if candidate set too large or $Q$ answerable then
        refine $Q$
    else if answer not unique then
        refine $C$
    else if cross-validation fails then
        repair $Q$ or $C$
return finalized instance $(Q, C, A)$
```

3.1 Task Overview

As shown in Table 1, each instance comprises an ambiguous question, a context containing distinctive attributes, the correct answer, and a distractor (a popular alternative sharing attributes with the target). The context is hidden from agents but available to a simulated user responder. Agents receive only the ambiguous question and operate with three actions: `search` to retrieve web information, `interact` to propose clarifying questions, and `answer` to provide the final response. The simulated responder replies with ”yes,” ”no,” or ”I don’t know” based solely on context information. Through this process, agents must recognize ambiguity, gather disambiguating details via interaction, and identify the correct answer. Implementation details for both agents and responders are provided in Appendix A.1 and Appendix A.2.

3.2 Data construction and Verification

Our construction methodology draws inspiration from BrowseComp’s answer-first approach (Wei et al., 2025), but fundamentally shifts focus from search complexity to ambiguity resolution. The central challenge in constructing such a benchmark is creating questions that appear reasonable yet systematically lack information for confident resolution. We observe that user ambiguity is particularly pronounced when dealing with similar concepts that share overlapping attributes, it is in these scenarios that additional clarification becomes truly necessary rather than merely helpful.

This observation leads us to design a systematic target-distractor methodology. We deliberately pair an target entity with a similar popular entity (the distractor), crafting questions using only their shared attributes while hiding distinctive information as context. This construction ensures that: (1) questions admit multiple plausible interpretations including the popular distractor, making direct answering unreliable; (2) the target answer possesses all described attributes, ensuring verifiability; and (3) distinctive attributes hidden in context provide clear disambiguation paths through interaction. Algorithm 1 formalizes this pipeline, which we detail in the following subsections alongside our two-stage verification process.

3.2.1 Construction Process

Annotators receive the following instruction:

> *“You need to find a pair of entities that are similar but differ in popularity. Use their shared attributes to construct an ambiguous question, and reserve the remaining distinctive attributes to form the context.”*

Following this instruction, the construction proceeds in four steps: **(1) Entity Selection**: annotators identify a lesser-known target and a popular distractor sharing overlapping characteristics; **(2) Attribute Categorization**: attributes are classified as shared (common to both) or distinctive (unique to target); **(3) Question Formulation**: only shared attributes are used to create questions admitting multiple plausible candidates; **(4) Context Formation**: distinctive attributes are reserved as context, ensuring question-context pairs uniquely identify the target while questions alone remain ambiguous.

3.2.2 Verification Process

We implement a two-stage verification protocol to ensure data quality and interaction necessity.

Stage 1: Completeness Verification. Independent annotators validate three requirements: (1) the target answer must possess all attributes described in both the question and context, (2) the question-context combination must admit only one valid answer with no plausible alternatives, and (3) instances where annotators identify valid alternative answers are discarded and reconstructed.

Stage 2: Interaction Necessity Validation. We verify that questions truly require interaction through two complementary checks. First, we manually confirm questions cannot be confidently resolved through direct web search, checking the first five Google result pages. Second, we conduct automated testing with three capable models (GPT-5, GPT-5-mini, Claude-Sonnet-4) across 5-round trials where models have access to search but no interaction. Questions successfully answered by two or more models without interaction are flagged as insufficiently ambiguous and undergo revision to strengthen their ambiguity.

3.3 Data Statistics

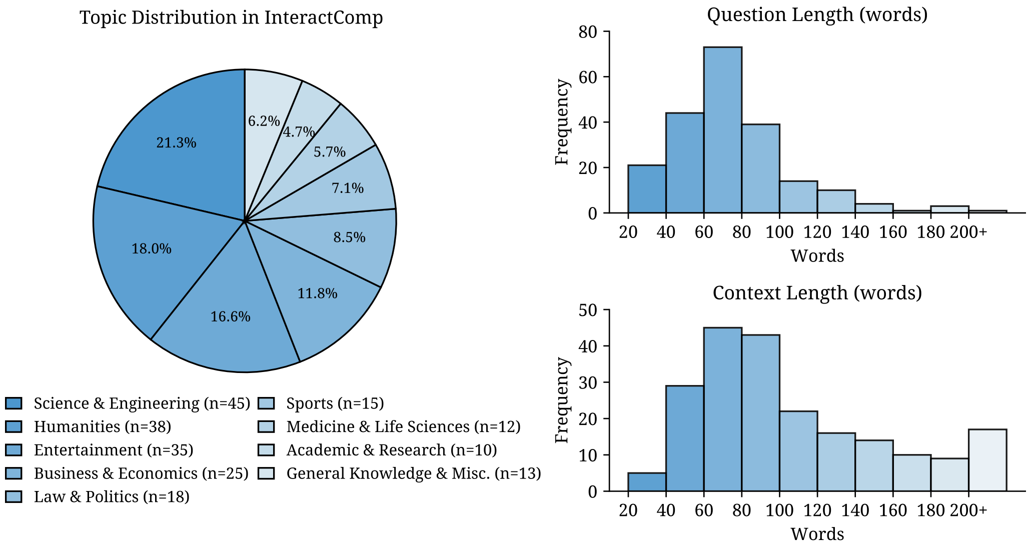

Figure 2: Topic distribution and question/context length statistics in InteractComp.

In this section, we present statistics on the topic distribution, question and context length distribution of our curated InteractComp dataset.

Topic distribution. Figure 2 presents the distribution of samples across 9 topic domains in the InteractComp dataset. The most represented categories include Science & Engineering (21.3%), Humanities (18.0%), and Entertainment (16.6%). The dataset also features Business & Economics (11.8%), Law & Politics (8.5%), and Sports (7.1%). Conversely, domains like Medicine & Life Science (5.7%), Academic & Research (4.7%), and General Knowledge & Misc. (6.2%) have fewer samples.

Question and Context Length distribution. Figure 2 illustrates the distribution of question and context lengths in the InteractComp dataset. Question length predominantly ranges between 40 to 80 words, with the majority falling within this interval. Context length shows a broader distribution, typically spanning from 40 to over 200 words, with peak frequency in the 60-100 word range. These distributions demonstrate that questions are concise yet informative, while contexts provide comprehensive disambiguation information.

Language distribution. The INTERACTCOMP dataset comprises bilingual instances with English accounting for 139 samples (66.19%) and Chinese contributing 71 samples (33.81%), enabling evaluation of interaction capabilities across different linguistic contexts.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

To systematically evaluate agent capabilities across different interaction paradigms, we design a controlled experimental framework that isolates and measures the incremental contribution of core agent capabilities: knowledge recall, information retrieval, and interactive clarification.

Agent Architecture: We employ the ReAct framework (Yao et al., 2023) as our base architecture, implementing three complementary configurations: (1) *Answer-only*: direct response generation testing pure knowledge recall, (2) *Answer+Search*: incorporating web search for information retrieval, and (3) *Answer+Search+Interact*: adding interactive clarification through interact with responder. This design enables measurement of capability increments while maintaining architectural consistency. To further investigate interaction behavior, we implement a forced-interaction variant for ablation studies that requires minimum interaction thresholds before answer generation. Implementation details are provided in Appendix A.1.

Models: We evaluate across diverse model families including proprietary models (GPT-4o-mini, GPT-4o, GPT-4.1, GPT-5, OpenAI o3, Grok-4, Doubao-1.6, Claude-Sonnet-4, Claude-Opus-4, Claude-3.5-Sonnet) and open-weight models (GLM-4.5, Kimi-K2, Deepseek-V3.1, Deepseek-R1, Qwen3-235B-A22B, Qwen2.5). Following established benchmarking practices, we standardize parameters where supported: temperature=0.6, top_p=0.95.We employ GPT-4o (temperature=0.0) as our grader, providing ground truth, agent response, and question context for binary correctness judgments.We implement responder simulation using GPT-4o (temperature=1.0) that provides structured feedback when agents employ the *interact* action.

Metrics: We evaluate agents across five key dimensions: (1) **Interaction Metrics**: Round (average number of conversation turns) and percentage of rounds where interact actions are used (IR) measuring behavioral patterns and action utilization; (2) **Performance Metrics**: Accuracy (Acc.) measuring the percentage of correctly answered queries, and Calibration Error (C.E.) measuring confidence calibration using 5 confidence bins; and (3) **Cost**: measured in USD reflecting computational resources usage for practical deployment considerations.

4.2 MAIN RESULTS

Table 2 presents comprehensive results across 17 models, revealing striking patterns in how different architectures handle ambiguous queries. The results expose fundamental limitations even in state-of-the-art systems, with the highest-performing model (GPT-5) achieving only 13.73% accuracy, demonstrating the benchmark's challenging nature.

Diverse Interaction Patterns Across Models. Models exhibit dramatically different interaction strategies, creating distinct behavioral profiles. GPT-4o-mini stands out as an extreme case: it asks questions in 73.95% of available rounds, by far the highest interaction rate, yet achieves only 7.14% accuracy—close to GLM-4.5 which barely interacts (0.25% IR). This suggests that excessive questioning without clear purpose can be counterproductive. Conversely, DeepSeek-R1 demonstrates more balanced behavior with 44.72% IR yielding 13.08% accuracy, the highest among open-weight models, indicating that willingness to interact can translate to better performance when used effectively.

Calibration Quality Correlates with Interaction Patterns. A remarkable finding is that models with higher interaction rates often exhibit superior calibration. GPT-4o-mini's aggressive questioning strategy, while not improving accuracy, results in dramatically better calibration (37.44 CE) compared to low-interaction models like Doubao-1.6 (84.35 CE). This pattern suggests that interaction, even when not optimally targeted, helps models develop more realistic confidence assessments about their knowledge limitations.