Noisy Interactive Graph Search: An Uncertainty-Based Approach with Online Modeling of Latent Expertise and Difficulty

Han Linghu
HKUST(GZ)
hlinghu866@connect.hkust-gz.edu.cn

Qianhao Cong
National University of Singapore
cong_qianhao@u.nus.edu

Liang Feng
Chongqing University
liangf@cqu.edu.cn

Lei Chen
HKUST & HKUST(GZ)
leichen@cse.ust.hk

Jing Tang*
HKUST & HKUST(GZ)
jingtang@ust.hk

ABSTRACT

Interactive graph search (IGS) has emerged as a powerful information retrieval paradigm for various applications. Given a hierarchy and an oracle that typically relies on human intelligence, IGS aims to identify the most precise concept for an unknown object while minimizing interaction costs. Most existing algorithms simplify the problem by assuming a perfect oracle that always provides correct answers. Others adopt an idealized noisy oracle that models noises as explicit error rates specified in advance and locate the target with Bayesian inference guided by a node-wise querying strategy. However, in real-world scenarios, the oracle inevitably makes mistakes and prior knowledge of the oracle is often limited. Moreover, the node-wise querying strategy that lacks holistic awareness of the search state and ignores the global hierarchical structure usually yields suboptimal queries. To address these challenges, we introduce IGS-RTA. We first formulate the problem based on search uncertainty, explicitly accounting for the randomness of the search state and hierarchical relations. We then propose a querying strategy that maximizes the expected uncertainty decrement. Our rigorous theoretical analysis establishes a logarithmic upper bound on the query complexity. In addition, to adapt to noisy settings with limited prior knowledge, we analyze oracle expertise and task difficulties, which characterize two groups of meta-factors that influence real query answering. We model their relationships using a probabilistic graphical model and design techniques to estimate these latent factors online. We evaluate IGS-RTA on two real-world datasets against six baselines. Results show that IGS-RTA improves search accuracy by up to 52% while reducing monetary costs by up to 8×.

PVLDB Reference Format:
Han Linghu, Qianhao Cong, Liang Feng, Lei Chen, and Jing Tang. Noisy Interactive Graph Search: An Uncertainty-Based Approach with Online Modeling of Latent Expertise and Difficulty. PVLDB, 19(11): 3119 - 3132, 2026.
doi:10.14778/3836663.3836677

*Corresponding Author

Proceedings of the VLDB Endowment, Vol. 19, No. 11 ISSN 2150-8097.
doi:10.14778/3836663.3836677

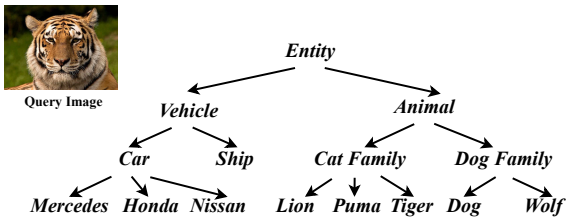


Figure 1: An example of the IGS problem on an image classification hierarchy [13, 35, 51].

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/HandsBro/IGS-RTA.

1 INTRODUCTION

Crowdsourcing leverages human intelligence to solve problems that are challenging for machines. Over the past decade, it has been extensively studied and widely applied as a powerful tool for numerous data-driven tasks [14, 15, 21, 23, 47, 49, 54]. Serving crowdsourcing as the oracle, interactive graph search (IGS) aims to locate the most appropriate concept (i.e., the deepest node) within a hierarchy H for an unlabeled object. In H, each node represents a potential label and each directed edge denotes a subclass-of relationship. During each round of the search, we pose a reachability query such as "is the target node reachable from (i.e., in the category of) x?" to the oracle and proceed based on the returned answer until the target node is identified. One typical application of IGS is data labeling, which aims to generate high-quality datasets for various supervised learning tasks, including image classification [13, 56], product categorization [22, 30, 39], software testing [3], and data filtering [41, 42, 44, 48]. The following example provides an image labeling task as an immediate illustration.

Example 1. Figure 1 illustrates an example hierarchy for image classification, where each node represents a potential target label and each directed edge denotes a "class of" relationship. Suppose the oracle is perfect (i.e., error-free), and we need to assign the correct label *Tiger* to an unlabeled image. We first ask the oracle "is this image in the category of *Entity*?" With a positive answer, we proceed to query node *Vehicle* which cannot reach the target node *Tiger*. After receiving the negative response from the oracle,

we switch to the subgraph about *Animal* and continue by querying nodes {*Animal, Cat Family, Lion, Puma, Tiger*}. Based on the answers {yes, yes, no, no, yes}, we finally assign the label *Tiger* to the image. □

As illustrated in the above example, the naive solution to the IGS problem is to search in a top-down manner. Specifically, we start at the root of the hierarchy for each task and iteratively examine its children until we find the one that receives a positive feedback from the oracle. We then switch to the subgraph under the verified child node and repeat the process until we finally locate the target, which is either a leaf or an internal node with all its children receiving negative responses from the oracle. However, in the worst case, the top-down search may potentially check every node in the hierarchy, leading to prohibitive monetary and time costs in interacting with the oracle.

To reduce interaction costs, Tao et al. [51] propose a binary search framework for the IGS problem. They first extract the heavy-path depth-first search (HPDFS) tree of the hierarchy using heavy-path decomposition, which is a classic graph decomposition algorithm [46]. Then they conduct binary searches on paths within the HPDFS tree to locate the target node. Building on the binary search framework, numerous algorithms [9, 30, 34, 35, 58, 60, 65] have been proposed to further enhance search efficiency. All these approaches assume a perfect oracle that always provides correct answers. However, in practice, the oracle inevitably makes mistakes. Since these methods inherently lack the ability to correct errors and it usually requires a series of queries for a single task, any erroneous answer during the search can lead to an incorrect final result, which significantly degrades their effectiveness.

Majority voting has been widely adopted [4, 6, 8, 10, 50, 62] to mitigate the impact of oracle errors. However, it still suffers from limited effectiveness [8], as it fails to guarantee the correctness of sequential decisions. On the other hand, our experiments show that naively adapting active learning [62] and reinforcement learning techniques [19] to the IGS problem yields sub-optimal performances. Recently, Cong et al. [8] attempt to address the noisy IGS problem using Bayesian inference. They model noise as an explicit error rate e_u for each node u, representing the probability of the oracle making an incorrect decision for the reachability query on node u, and propose a node-wise querying strategy that selects the node with the maximal expected posterior probability. However, modeling noises via explicit error rates oversimplifies real-world conditions. In practice, noises are often influenced by multiple factors, and we typically lack the prior knowledge about when or where the oracle makes mistakes (i.e., hidden error rates). Moreover, the node-wise querying strategy that lacks holistic awareness of the search state and ignores hierarchical structure usually yields suboptimal queries.

Challenges. The main challenges faced by the IGS problem are twofold: (i) How to search effectively under a practical noisy oracle with limited prior knowledge? (ii) How to design more efficient querying strategy?

To address these challenges, we propose IGS-RTA in this paper. We first formulate the noisy IGS problem based on search uncertainty which considers the randomness of the search state. Then we propose a querying strategy that enables holistic reasoning across the hierarchy by maximizing the expected uncertainty decrement in each round. Our rigorous theoretical analysis establishes a lower bound on the expected uncertainty decrement, guaranteeing a logarithmic query complexity for the search. Considering the limited prior knowledge in real-world scenarios, we further propose online estimation techniques. Specifically, rather than assuming pre-specified error rates, we analyze two meta-factors, the oracle expertise and task difficulties, that influence the real query answering. Expertise characterizes oracle-side factors such as topic familiarity and fatigue, while difficulties characterize task-side properties like sibling similarity and node depth. We further model their relationships based on a probabilistic graphical model. To estimate both factors, we design interaction records to collect query data on the fly during the search and further propose an online inference algorithm based on expectation maximization. We evaluate IGS-RTA against six baselines on two real-world datasets, and the experimental results show that IGS-RTA improves search accuracy by up to 52% while reducing interaction costs by up to a factor of 8. In summary, we make the following contributions.

- We study the IGS problem under a practical noisy oracle with limited prior knowledge based on search uncertainty, explicitly accounting for the randomness of the search state. We propose a querying strategy that enables holistic reasoning across the hierarchy by maximizing the expected uncertainty decrement. We further show a logarithmic upper bound on the query complexity with a rigorous theoretical analysis.
- We characterize two meta-factors affecting query responses, i.e., oracle expertise and task difficulty, and model their relations using a probabilistic graphical model. Building on this, we propose online inference techniques to estimate these factors with limited prior knowledge, enabling effective and efficient noisy search.
- We conduct comprehensive evaluations of IGS-RTA against six baseline methods on two real-world datasets. Experimental results demonstrate that IGS-RTA improves search accuracy by up to 52%, while reducing monetary costs by up to a factor of 8.

Roadmap. We organize the paper as follows. Section 2 provides the necessary background of noisy interactive graph search. Section 3 introduces the uncertainty-based querying strategy with the theoretical analysis. Section 4 examines the two meta factors that affect the query answering and details the inference procedure. Section 5 describes the complete design of IGS-RTA. Section 6 reviews related work, and Section 7 evaluates the performance of IGS-RTA. We further discuss remaining opportunities in Section 8. Finally, we conclude in Section 9.

2 PRELIMINARY

A classification hierarchy H is defined as a directed acyclic graph (DAG) denoted by $H = G(V, E)$, where V is a set of n nodes and E is the set of m edges. We assume the hierarchy in the IGS problem is single-rooted. Otherwise we add a dummy root denoted by r and connect it to the original roots in H. We denote a directed edge from u to v by $e(u, v)$. Each edge $e(u, v) \in E$ indicates that v is a sub-concept (i.e., subclass) under u, and it also means v is a more specific concept than u in the hierarchy. Table 1 summarizes the frequently used notations.

Table 1: Notations.

$H = G(V, E)$	a hierarchy where $n = \|V\|$ and $m = \|E\|$
W	the oracle (i.e., a set of workers)
$Q_r(q)$	the reachability query on node q
a	the answer of $Q_r(q)$ from W
α_i	latent expertise of worker i
β_q	latent difficulty of $Q_r(q)$
Z_q	ground truth of $Q_r(q)$
$p(u)$	prior probability of u being the target
ϵ	the failure threshold of termination
A_{iq}	boolean answer of worker i to the query $Q_r(q)$
Ac_{iq}	the probability of worker i correctly answering the query $Q_r(q)$

In IGS, an independent oracle is expected to answer reachability queries issued by IGS algorithms. In this paper, we denote the oracle by W, which consists of a set of crowdworkers. Notably, utilizing classifiers [60] as the oracle in IGS is challenging, as IGS is primarily designed to handle tasks that are difficult for machines yet solvable by humans. We use $Q_r(q)$ to represent the reachability query on node q: "is the target node reachable from (i.e., in the category of) node q?", and the oracle is expected to return either "yes" or "no" denoted by a. Additionally, node u can reach node v (i.e., $u \rightarrow v$) if there exists a path going from u to v. Otherwise, u cannot reach v (i.e., $u \nrightarrow v$). The final answer a returned by the oracle is aggregated from each worker's decision, and we denote the answer of worker $i \in W$ on $Q_r(q)$ by A_{iq}. We also denote a query $Q_r(q)$ and its answer a by (q, a). The ground truth of $Q_r(q)$ is represented by Z_q. In the case with an idealized oracle, we denote the probability of it returning an incorrect a on $Q_r(q)$ by e_q. Specifically, given a query node q and a target node t, the conditional probability of observing the result of (q, a) is

$$p((q,a) \mid t) = \begin{cases} 1 - e_q, & \text{if } (q \rightarrow t \wedge a = \text{yes}) \text{ or } (q \nrightarrow t \wedge a = \text{no}), \\ e_q, & \text{otherwise.} \end{cases}$$

In more practical scenarios, it is difficult for us to determine the reliability of each answer a returned by the oracle with limited prior knowledge. This represents one of the key challenges we aim to address in this work: determining the trustworthiness of each oracle response. In this work, we utilize α_i to denote the latent expertise of worker i, and use β_q to represent the difficulty of the reachability query on node q. A detailed discussion will be provided in Section 4.

For each node $u \in V$, we use $p(u)$ to represent the prior probability of node u being the target node. In this work, we do not require the prior distribution to be given in advance. In its absence, we can simply set it as the uniform distribution (i.e., $\frac{1}{n}$ for each node, where $n = |V|$). Note that some previous work [9, 30] has explored how to learn the hidden prior distribution on the fly. These techniques are orthogonal to our approach but are beyond the scope of this work.

In the traditional setting with a perfect oracle, the primary goal of the IGS problem is to minimize interactions with the oracle, which incur significant time and monetary costs. In the noisy setting, we also need to consider the effectiveness of IGS algorithms. We define the noisy interactive graph search problem as follows.

Definition 1. (*Noisy Interactive Graph Search*) Given a classification hierarchy H, a noisy oracle W, and the prior probability $p(u)$ for each node $u \in V$ being the target node, noisy interactive graph search aims to minimize the interaction costs with the oracle W while maximizing search accuracy.

3 NOISY INTERACTIVE GRAPH SEARCH

In this section, we first introduce our uncertainty-based querying strategy, which considers the randomness of the search state and enables holistic reasoning across the hierarchy. We then provide a lower bound of this strategy.

3.1 Uncertainty-Driven Strategy

We define the search uncertainty in a classification hierarchy H based on randomness as follows.

Definition 2 (Search Uncertainty [7, 38, 59])**.** Given a hierarchy H and the prior distribution, the search uncertainty in H denoted by $Q(H)$ can be represented by Shannon entropy,

$$Q(H) = -\sum_{v \in V} p(v) \log p(v). \tag{1}$$

For each node $v \in V$, we use $p(v)$ to represent the probability of v being the target node, and $\sum_{v \in V} p(v) = 1$. According to the definition, a larger value of $Q(H)$ indicates greater randomness of the search, which means more queries are expected to make the search concentrated. Conversely, a smaller $Q(H)$ implies that the target's location is more clearly inferred.

For each node $u \in V$, according to Bayes' theorem, the posterior probability of node u being the target after observing a sequence of queries $\Omega = \{(q_1, a_1), \ldots, (q_k, a_k)\}$ asked can be computed by

$$p(u|\Omega) = \frac{p(\Omega|u)p(u)}{p(\Omega)}. \tag{2}$$

Furthermore, we consider that queries are independently answered: the answer to a query only depends on the reachability and the error probability, not influenced by other queries [21, 42, 53]. The probability of observing Ω given the target node z is

$$p(\Omega|z) = \prod_{(q,a) \in \Omega} P((q,a)|z). \tag{3}$$

As a result, the posterior probability of node u after observing a new pair (q, a) based on a sequence of queries Ω, denoted by $\Omega \cup \{(q, a)\}$, can be calculated by

$$\begin{aligned} p(u|\Omega \cup \{(q,a)\}) &= \frac{p(\Omega \cup \{(q,a)\}|u)p(u)}{p(\Omega \cup \{(q,a)\})} \\ &= \frac{p((q,a)|u)p(\Omega|u)p(u)}{\sum_{v \in V} p(\Omega \cup \{(q,a)\}|v)p(v)} \\ &= \frac{p((q,a)|u)p(\Omega|u)p(u)/p(\Omega)}{\sum_{v \in V} p((q,a)|v)p(\Omega|v)p(v)/p(\Omega)} \\ &= \frac{p((q,a)|u)p(u|\Omega)}{\sum_{v \in V} p((q,a)|v)p(v|\Omega)} \\ &= \frac{p((q,a)|u)p(u|\Omega)}{p((q,a)|\Omega)}, \end{aligned} \tag{4}$$

where the first equality is by (2), the second equality is by (3) and the definition that $p(\Omega) = \sum_{v \in V} p(\Omega|v)p(v)$ for any Ω, the third

Algorithm 1: Naive Noisy Interactive Graph Search

Input: classification hierarchy H, failure threshold ϵ, prior probability $p(u)$ and error rate e_u for each node $u \in V$

Output: target node

1 $\Omega \leftarrow \emptyset$;
2 $p_{max} \leftarrow \max_v\ p(v)$;
3 $p(v|\Omega) \leftarrow p(v)$ for each node $v \in V$;
4 **while** $p_{max} \le 1 - \epsilon$ **do**
5 $\quad q \leftarrow \arg\max_{q \in V} \Delta Q(H)$;
6 $\quad a \leftarrow$ answer of $Q_r(q)$ from the oracle;
7 $\quad$ compute $p(\Omega \cup \{(q,a)\})$ and $p((q,a)|u)$ for $u \in V$;
8 $\quad$ update the posterior probability for $v \in V$ by Eq. (4);
9 $\quad \Omega \leftarrow \Omega \cup \{(q,a)\}$;
10 $\quad p_{max} \leftarrow \max_{v \in V}\ p(v|\Omega)$;
11 **Return** $\arg\max_{v \in V} p(v|\Omega)$;

equality is by (2), and last equality is by chain rule. We use (4) to characterize the effect of the new query pair (q, a) on the search state given Ω.

Now, we are ready to compute the search uncertainty with respect to the posterior probabilities after querying $\Omega \cup \{(q,a)\}$ by

$$Q(H|\Omega \cup \{(q,a)\}) = -\sum_{v \in V} p(v|\Omega \cup \{(q,a)\}) \log p(v|\Omega \cup \{(q,a)\}). \tag{5}$$

We denote the subgraph of H rooted at q by H_q and the complement graph of H_q by H_q^c. Combining (4) and (5), for each new query pair (q, a) based on Ω, the expected search uncertainty in the hierarchy with respect to the query outcome of a (i.e., $a = yes$ and $a = no$), can be computed by

$$\begin{aligned}
&\mathbb{E}[Q(H|\Omega \cup \{(q,a)\})] \\
&= \sum_{a \in \{yes,no\}} Q(H|\Omega \cup \{(q,a)\}) \cdot p((q,a)|\Omega) \\
&= Q(H|\Omega) - e_q \log e_q - (1-e_q)\log(1-e_q) \\
&\quad + p((q,yes)|\Omega) \log p((q,yes)|\Omega) + p((q,no)|\Omega) \log p((q,no)|\Omega).
\end{aligned} \tag{6}$$

Based on (6), we can now compute the expected uncertainty decrement as,

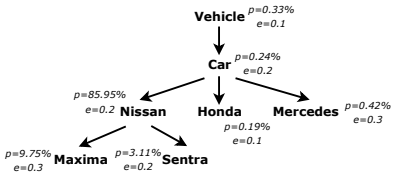
$$\begin{aligned}
\Delta Q(H) = Q(H|\Omega) &- \mathbb{E}[Q(H|\Omega \cup \{(q,a)\})] \\
&- p((q,yes)|\Omega) \log p((q,yes)|\Omega) \\
&- p((q,no)|\Omega) \log p((q,no)|\Omega) \\
&+ e_q \log e_q + (1-e_q)\log(1-e_q).
\end{aligned} \tag{7}$$

Furthermore, the probability of observing (q, yes) (i.e., $p((q, yes))$) and (q, no) (i.e., $p((q, no))$) can be calculated as

$$p((q,yes)|\Omega) = (1-e_q) \sum_{v \in H_q} p(v|\Omega) + e_q \sum_{v \in H_q^c} p(v|\Omega), \tag{8}$$

$$p((q,no)|\Omega) = e_q \sum_{v \in H_q} p(v|\Omega) + (1-e_q) \sum_{v \in H_q^c} p(v|\Omega). \tag{9}$$

For simplicity of expression, we denote $x := \sum_{v \in H_q} p(v) \in (0,1)$ and $y := e_q \in (0,1)$ in the following contents. Combining (7)–(9),



v	OIGS	IGS-RTA
Vehicle	0.0000	0.0000
Car	0.0075	0.0027
Nissan	0.0259	0.0098
Honda	0.0137	0.0034
Mercedes	0.0031	0.0014
Maxima	0.0628	**0.0295**
Sentra	**0.0634**	0.0247

Figure 2: Example of a noisy search. The left hierarchy is a subgraph from ImageNet [13], and the right table shows the details.

we obtain the closed-form expression of the expected uncertainty decrement and the Theorem 1.

$$\begin{aligned}
\Delta Q(H) = &-(x - 2xy + y)\log(x - 2xy + y) \\
&-(1 + 2xy - x - y)\log(1 + 2xy - x - y) \\
&+ y \log y + (1-y)\log(1-y).
\end{aligned} \tag{10}$$

Theorem 1 (Harmless Querying)**.** Given a hierarchy H, the prior probability $p(u)$ of each node $u \in V$ being the target, and the error rate e_u for each node $u \in V$, the expected uncertainty decrement, defined as $\Delta Q(H) = Q(H|\Omega) - \mathbb{E}[Q(H|\Omega \cup \{(q,a)\})]$, with respect to the answer a on $Q_r(q)$, is always non-negative.

Due to space limitations, the proof of this theorem and all remaining proofs are deferred to our technical report [1]. Notably, the uncertainty decrement $\Delta Q(H)$ represents the uncertainty gap (i.e., reduction in randomness) of the search after asking a new query, and is formulated as the difference between the uncertainty from the previous step and the current uncertainty. According to Theorem 1, asking more questions will generally decrease the uncertainty of the search when the error rate of the query node is not equal to 0.5. Based on (10), we propose a greedy strategy that selects the query node q which maximizes the expected decrement in uncertainty. Formally,

$$q = \arg\max_{q \in V} \Delta Q(H). \tag{11}$$

Algorithm 1 presents a naive noisy interactive graph search algorithm, assuming the error rate and prior probability of each node are given in advance. Specifically, in each round we select the node that maximizes $\Delta Q(H)$, query the oracle about $Q_r(q)$, and update posterior probabilities based on the returned answer until p_{max} exceeds the accuracy threshold $1 - \epsilon$ (lines 4-10). Finally, we return $\arg\max_{v \in V} p(v|\Omega)$ as the result.

Complexity Analysis. For ease of analysis, assume there are k rounds of querying in total. The time complexity of Algorithm 1 is $O(k\rho n)$ where $n = |V|$ is the total number of nodes in H since the query selection takes $O(n)$, update takes $O(n)$, and the oracle response time takes $O(\rho)$ in each round. The space complexity is $O(n)$. Notably, the primary goal of IGS algorithms is to reduce interaction costs, which involve both time and monetary expenses.

Example 2. Figure 2 illustrates the divergence in query strategies. Suppose we have an unlabeled image with the ground truth *Nissan*, and we are provided with the hierarchy on the left to label this

image. The prior probability of being the target and the explicit error rate are listed beside each node. The right table shows the posterior increments (OIGS column) and the expected uncertainty decrements (IGS-RTA column). Attracted by the low error rate and the highest posterior increment, OIGS selects *Sentra* in this round. In contrast, IGS-RTA evaluates the global search state and queries *Maxima*, which offers the greatest uncertainty decrement. This divergence leads to different results: IGS-RTA identifies *Nissan* as the target and terminates immediately, whereas OIGS requires three additional queries to reach the same conclusion. □

3.2 Theoretical Guarantees

Following the standard analysis for multivariate discrete functions, Equation (10) indicates that $\Delta Q(H)$ varies non-monotonically with y (i.e., e_q). Specifically, $\Delta Q(H)$ decreases as y ranges from 0 to 0.5, and increases as y rises from 0.5 to 1. Denote $e^* = \min_{q\in V}|e_q - 0.5|$, and we have the following inequality

$$\begin{aligned}\Delta Q(H) \geq & -(x - 2xe^* + e^*)\log(x - 2xe^* + e^*) \\ & -(1 + 2xe^* - x - e^*)\log(1 + 2xe^* - x - e^*) \\ & + e^*\log e^* + (1 - e^*)\log(1 - e^*) \triangleq h(q).\end{aligned}$$

As a result, we have

$$\max_{q\in V}\ \Delta Q(H) \geq \max_{q\in V}\ h(q).$$

Additionally, $h(q)$ increases when x (i.e., $\sum_{v\in H_q} p(v)$) increases from 0 to 0.5 and decreases as x further increases from 0.5 to 1. Therefore, a lower bound of the right hand side of the above inequality can be reached when x achieves the maximal distance to the middle point (i.e., at $q^* = \arg\max_{q\in V} |\sum_{v\in H_q} p(v) - 0.5|$). Hence, there is

$$\max_{q\in V}\ \Delta Q(H) \geq \max_{q\in V}\ h(q) \geq h(q^*) \triangleq \lambda. \tag{12}$$

It is evident that the maximal value of $\sum_{v\in H_q} p(v) - 0.5$ is 0.5 when q is set to the root. However, we do not consider this trivial case, as it is meaningless to ask $Q_r(r)$ which is surely positive since the root node can reach all other nodes in the hierarchy. This is consistent with Equation (10), which equals 0 when q is set to the root. In the following, we establish an upper bound on query complexity by deriving a lower bound λ for the expected uncertainty decrement on nodes in $V \setminus \{r\}$.

Lemma 1. Given a spanning tree of the hierarchy $H = G(V, E)$ and $p(v)$ for each node $v \in V$, the following bound holds:

$$\frac{1 - p^*}{d + 1} \leq x \leq \frac{d + p^*}{d + 1},$$

where x represents $\sum_{v\in H_q} p(v)$, d is the maximum degree of nodes in the spanning tree and $p^* = \max_{v\in V}\ p(v)$.

The upper bound and lower bound of x also imply the symmetry about $x = 0.5$, since they have the same distance to the axis of symmetry (i.e., $x = 0.5$), which is $\frac{d-1+2p^*}{2d+2}$.

Lemma 2. The uncertainty-based strategy proposed in (11) selects a query node q on top of Ω ensuring that the expected uncertainty decrement $\Delta Q(H) = Q(H|\Omega) - \mathbb{E}[Q(H|\Omega \cup \{(q, a)\})] \geq \lambda$, where

$$\lambda = \mathcal{H}(\omega) - \mathcal{H}(e^*) \text{ with } \omega = \frac{\epsilon - 2e^*\epsilon + de^* + e^*}{d + 1},$$

where $\mathcal{H}(x) = -x\log x - (1 - x)\log(1 - x)$ denotes the binary entropy function.

Theorem 2 (Query Complexity)**.** The expected number of queries required to terminate is upper bounded by $O(\frac{\log n}{\lambda})$.

This result establishes a logarithmic query complexity bound for IGS-RTA, theoretically guaranteeing its efficiency. Unlike OIGS [8] which optimizes local posterior probability multiplicatively, our approach ensures a global additive reduction in uncertainty, preventing the search from being trapped in local optima due to noise.

4 ONLINE MODELING

A critical challenge in noisy IGS is that the explicit error rates relied upon in Section 3 are often unavailable in real-world scenarios. In their absence, we must resort to suboptimal heuristics such as the midpoint approach [9], which frequently yields degraded performance. In this section, we characterize and analyze two pivotal meta-factors influencing query response quality and introduce online inference techniques to enable effective search in practical noisy environments.

4.1 Probabilistic Graphical Model

Real-world crowdsourcing decisions are influenced by numerous factors. Nevertheless, it is hard to enumerate all noise sources explicitly. Inspired by the widely adopted expertise-difficulty model for handling noise [15, 17, 26, 28, 29, 62], we utilize worker expertise and task difficulty to characterize two meta-factors that model the correctness of an oracle answering a query. Expertise characterizes oracle-side factors. For instance, workers may exhibit higher focus and accuracy in the morning (i.e., time of day), whereas prolonged task engagement often leads to declining performance (i.e., hours worked). In contrast, task difficulty reflects task-side properties, such as sibling similarity and node depth. Denoted the answer of worker i to $Q_r(u)$ by A_{iu}, we adopt the probabilistic graphical model (PGM) [15, 36, 55] to capture the relationship between the factors and the answers returned by W. Formally, the probability of worker i correctly answering $Q_r(u)$ is

$$p(A_{iu} = Z_u \mid \alpha_i, \beta_u) = \frac{1}{1 + e^{-\alpha_i\beta_u}}. \tag{13}$$

To model the expertise of the oracle W, we assign each worker i a parameter $\alpha_i \in (-\infty, +\infty)$, which characterizes three levels of reliability. A positive α_i indicates better expertise, with larger values reflecting greater proficiency. Conversely, a negative α_i indicates that worker i is adversarial, consistently inverting the correct answer, either intentionally or due to persistent misunderstandings, with more negative values reflecting stronger adversarial tendencies. Finally, when $\alpha_i = 0$, it implies that worker i lacks domain knowledge, and the response is uninformative regardless of the query's difficulty.

On the other hand, the inverted difficulty of the reachability query on node u is modeled by a hidden positive parameter $\beta_u \in (0, +\infty)$. As u's difficulty increases, β_u decreases, which reduces the probability of W correctly answering $Q_r(u)$. Specifically, when β_u is close to 0, $Q_r(u)$ is highly ambiguous such that even domain experts can only answer it by random guessing (i.e., with a 50% chance

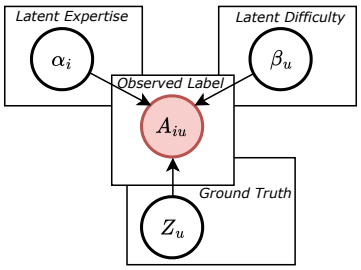


Figure 3: The probabilistic graphical model of worker expertise, task difficulty, ground truth, and observed data.

of being correct). Conversely, as β_u approaches infinity, $Q_r(u)$ becomes so straightforward that any positive worker can answer it correctly. Figure 3 illustrates the structure of our model, depicting the probabilistic relationships among expertise, difficulties, ground truth, and the observed data.

Example 3. According to (13), as α_i varies from $-\infty$ to $+\infty$, the probability of a worker i correctly answering the same reachability query $Q_r(u)$ changes from 0 to 1. Similarly, for a fixed worker with a positive α_i, the probability of correctly answering $Q_r(u)$ increases from 0.5 to 1 as β_u increases from 0 to $+\infty$. □

To estimate the latent factors affecting answer correctness, it is necessary to simultaneously infer both the oracle's hidden expertise and the unknown difficulties of query nodes. Namely, for a reachability query on node u, our goal is to determine the true answer Z_u by aggregating responses from k independent crowdworkers W, while jointly inferring each worker's expertise and task difficulties.

4.2 Data Collection

The first challenge in the estimation is the lack of sufficient data. Previous estimating methods [10, 15, 23, 33, 45, 57] assume that questions and their corresponding answers from different workers are available prior to the inference step. However, this assumption does not hold in the noisy IGS problem. Specifically, in each round of IGS algorithms, a single query node u will be selected, and the oracle answers the reachability query $Q_r(u)$ to determine whether the target node can be reached from u. Based on the oracle's feedback, we repeat this process: selecting a query node, posing the reachability query, and obtaining the oracle's response. Since query node selection depends on the uncertainty updated based on previous queries, it is not feasible to pre-determine all query nodes and collect worker responses in advance for estimation. Furthermore, as shown in our experimental results in Section 7, the average number of queries for one task is limited. Hence, even if we could collect all queries and their answers in advance, we may still suffer from the poor performance due to insufficient data.

To address this challenge, we propose Interaction Records for noisy IGS inference based on the PGM model. The core idea is to learn both the oracle's expertise and the nodes' difficulties from all historical interactions. Formally, the Interaction Records are defined as follows.

Algorithm 2: IGS-RTA

Input: hierarchy H, number of workers k, failure threshold ϵ, prior probability $p(u)$ for each $u \in V$
Output: target node q

$\alpha_i \leftarrow 1$ for each worker $i \in W$;
$\beta_u \leftarrow 1$ for each node $u \in V$;
$IR \leftarrow \emptyset$;
for *task* $t \in T$ **do**
  $\Omega \leftarrow \emptyset$;
  $p_{max} \leftarrow \max_{v \in V} p(v)$;
  $p(v|\Omega) \leftarrow p(v)$ for each node $v \in V$;
  while $p_{max} \leq 1 - \epsilon$ **do** `// Stage 1:query selection`
    for *each worker* $i \in W$ **do**
      for *each node* $j \in V$ **do**
        compute the estimated Ac_{ij} by Eq. (13);
    estimate $\tilde{e}_j$ for each node j by Eq. (18);
    $q \leftarrow \arg\max_{v \in V} \Delta Q(H)$;
    $A_q \leftarrow \emptyset$; `// Stage 2:decision & estimation`
    for *worker* $i \in W$ **do**
      $A_q \leftarrow A_q \cup \{A_{iq}\}$;
    $IR \leftarrow IR \cup \{(t, q, A_q)\}$;
    while Q *not converged* **do**
      update Q function by Eq. (16);
      update $\boldsymbol{\alpha}, \boldsymbol{\beta}$ by Eq. (17);
    $a \leftarrow$ answer of $Q_r(q)$ from IR; `// Stage 3:update`
    compute $\tilde{e}_q$ based on $A_q, a, \boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ by Eq. (19);
    compute $p(\Omega \cup \{(q, a)\})$ and $p((q, a)|u)$ for $u \in V$;
    update the posterior probability for $v \in V$ by Eq. (4);
    $\Omega \leftarrow \Omega \cup \{(q, a)\}$;
    $p_{max} \leftarrow \max_{v \in V} p(v|\Omega)$;
  Return $\arg\max_{v \in V} p(v|\Omega)$;

Definition 3. (Interaction Records) Interaction Records store all historical query pairs across tasks and are defined as:

$$IR = \{(t_1, u_1, A_{u_1}), (t_2, u_2, A_{u_2}), \ldots, (t_n, u_m, A_{u_m})\},$$

where t_i denotes the i^{th} task, u_i represents query nodes selected during task t_i, and A_u corresponds to the oracle's answers to $Q_r(u)$. We call each tuple in IR as a record.

According to the PGM, whether $Q_r(u)$ can be correctly answered depends on the difficulty of node u and the expertise of W only, and it is not related to the tasks (i.e., target nodes). Therefore, this task independence allows us to utilize IR to accumulate all past interactions and leverage historical data for real-time parameter estimation. Each time a new query is issued and answered by the oracle, we insert the new record into IR and then perform inference, as described below, using all available historical records until the parameter estimates converge.

4.3 Inference

We employ the expectation-maximization (EM) to compute the maximum likelihood estimates of the latent expertise $\boldsymbol{\alpha}$, unknown

difficulties $\boldsymbol{\beta}$, and hidden ground truth $\boldsymbol{Z}$, based on the collected *IR*. The inference details are presented as follows. Specifically, we set the initial values $\boldsymbol{\alpha}^{\mathbf{0}}$ and $\boldsymbol{\beta}^{\mathbf{0}}$, and iteratively perform the following two steps until convergence.

4.3.1 E-Step. In the E step, we compute the standard Q function with the parameter obtained from the previous iteration. Specifically, denote parameters acquired in the i^{th} round by $\boldsymbol{\alpha}^{\boldsymbol{i}}$ and $\boldsymbol{\beta}^{\boldsymbol{i}}$, in the $(i+1)^{th}$ round E step, we need to compute

$$Q(\boldsymbol{\alpha}, \boldsymbol{\beta} | \boldsymbol{\alpha}^{\boldsymbol{i}}, \boldsymbol{\beta}^{\boldsymbol{i}}) = \sum_{Z} \log p(\mathbf{A}, Z | \boldsymbol{\alpha}, \boldsymbol{\beta}) p(Z | \mathbf{A}, \boldsymbol{\alpha}^{\boldsymbol{i}}, \boldsymbol{\beta}^{\boldsymbol{i}}). \tag{14}$$

Formally, the Q function is defined as the expectation of the log-likelihood $\log p(\mathbf{A}, \mathbf{Z} | \boldsymbol{\alpha}, \boldsymbol{\beta})$ with respect to the posterior probability of $\mathbf{Z}$, denoted by $p(\mathbf{Z} | \mathbf{A}, \boldsymbol{\alpha}^{\boldsymbol{i}}, \boldsymbol{\beta}^{\boldsymbol{i}})$, given the observed data $\mathbf{A}$ and the current parameters $\boldsymbol{\alpha}^{\boldsymbol{i}}$ and $\boldsymbol{\beta}^{\boldsymbol{i}}$. Denoted the set of k answers for $Q_r(u)$ by $\mathbf{A}_u = \{l_{ij} \mid j = u\}$, we can then compute the posterior probability $p(Z | A, \boldsymbol{\alpha}^{\boldsymbol{i}}, \boldsymbol{\beta}^{\boldsymbol{i}})$ by

$$p(Z_u | \mathbf{A}, \boldsymbol{\alpha}^{\boldsymbol{i}}, \boldsymbol{\beta}^{\boldsymbol{i}}) \propto p(Z_u) \prod_{i} p(A_{iu} | Z_u, \alpha_i^i, \beta_u^i). \tag{15}$$

Notably, according to the conditional independence in the PGM, we have $p(Z_u | \boldsymbol{\alpha}^{\boldsymbol{i}}, \beta_u^i) = p(Z_u)$.

Combining (14) and (15), we update the Q function in the E step by calculating

$$Q(\boldsymbol{\alpha}, \boldsymbol{\beta} | \boldsymbol{\alpha}^{\boldsymbol{i}}, \boldsymbol{\beta}^{\boldsymbol{i}}) = \sum_{u} \mathbb{E}[\log p(Z_u)] + \sum_{i,u} \mathbb{E}[\log p(A_{iu} | Z_u, \alpha_i, \beta_u)]. \tag{16}$$

4.3.2 M-Step. With the Q function $Q(\boldsymbol{\alpha}, \boldsymbol{\beta} | \boldsymbol{\alpha}^{\boldsymbol{i}}, \boldsymbol{\beta}^{\boldsymbol{i}})$ updated in the previous E step, we maximize it to obtain the parameters for the $(i+1)^{th}$ round by computing

$$\boldsymbol{\alpha}^{\boldsymbol{i+1}}, \boldsymbol{\beta}^{\boldsymbol{i+1}} = \arg\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} Q(\boldsymbol{\alpha}, \boldsymbol{\beta} | \boldsymbol{\alpha}^{\boldsymbol{i}}, \boldsymbol{\beta}^{\boldsymbol{i}}). \tag{17}$$

We compute $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to locally maximize the Q function in each iteration with gradient ascent.

5 IGS-RTA

Integrating the analysis in Section 3 and Section 4, we introduce IGS-RTA as follows. Algorithm 2 provides the complete procedure.

We initialize α_i for each worker i and β_u for each node u with the prior value of 1 (lines 3-4), reflecting a prior belief that the probability of correctly answering $Q_r(u)$ exceeds 0.5. This encodes the general assumption that crowdworkers (i.e., the oracle) are generally reliable, as they are compensated for their responses.

Query Selection. For each incoming task t, to effectively select a query node q, we need to estimate the proxy error rate for each node $j \in V$, denoted by $\tilde{e}_j$, as discussed in Section 3. Specifically, for each node $j \in V$, we go through each worker $i \in W$ and calculate the individual accuracy Ac_{ij} by (13) (lines 9-11). Then for each node j, we estimate the error rate $\tilde{e}_j$ (line 12) by

$$\tilde{e}_j = 1 - \sum_{m=\lceil k/2 \rceil}^{k} \sum_{\substack{S \subseteq \{1,2,\ldots,k\} \\ |S|=m}} \prod_{i \in S} Ac_{ij} \prod_{i \notin S} (1 - Ac_{ij}). \tag{18}$$

Now with the prior probabilities and estimated error rates, we select query node q by our greedy strategy (line 13). As aforementioned, the prior probability of each node being the target is not essential for our algorithm. When unavailable, we can simply assign $\frac{1}{n}$ to each node.

Decision-Making & Estimation. After selecting the query node, we proceed to acquire answers to $Q_r(q)$ and estimate $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. We first ask each worker i about $Q_r(q)$, and add the answer A_{iq} into the record (lines 13-16). Then we insert the new record into *IR* (line 17). With the expanded *IR*, we conduct inference until the Q function converges, during which we update each worker's expertise and difficulty of node u for $u \in IR$ (lines 18-20). Notably, there may be multiple records about the same node u in *IR*, and we compute β_u by averaging over these records.

Posterior Update. After the inference step, we retrieve the answer to $Q_r(q)$ from *IR* (line 21) and calculate $\tilde{e}_q$ based on the updated parameters (line 22), where a is the aggregated oracle answer. With uninformative priors, this becomes,

$$\tilde{e}_q = 1 - \frac{p(A_q | Z_q = a) p(Z_q = a)}{\sum_{z \in \{yes, no\}} p(A_q | Z_q = z) p(Z_q = z)}. \tag{19}$$

Finally, we update the posterior probability of each node being the target, update the query set, and repeat the process until the target is identified (lines 23-27).

Complexity Analysis. Supposing k rounds of queries are needed for each task, the space complexity of Algorithm 2 is $O(n + tk)$ where t denotes the total number of tasks. Denoted the convergence time of the learning component by γ and the interaction time by ρ, the time complexity of Algorithm 2 is $O(k\gamma\rho n)$, where $n = |V|$.

Theoretical Analysis. We further discuss the overall query comlexity of Algorithm 2 as follows.

Lemma 3 (EM Estimation Bound [11, 18])**.** Let e_q and $\hat{e}_q$ denote the true and estimated error rates, respectively. Given a proper initialization via an effective crowdsourcing pre-screening mechanism to ensure identifiability, the estimation error is bounded by $|\hat{e}_q - e_q| \leq 2\sqrt{\frac{\log k}{k}}$ with a probability of at least $1 - O(1/k)$, where k denotes the total number of accumulated interactions.

This bound directly follows from Gao and Zhou [18], thus we omit the detailed proof for brevity. Furthermore, the uncertainty decrement $\Delta Q(H)$ is Lipschitz continuous with respect to e_q.

Lemma 4 (Lipschitz Continuity)**.** Let e_q denote the true error rate of a crowdsourcing worker. Provided that the error rates are bounded away from absolute 0 and 1 (i.e., $e_q \in [\eta, 1 - \eta]$ for a small constant $\eta > 0$), the expected uncertainty decrement function $\Delta Q(H; e_q)$, which is composed of Shannon entropy functions, has bounded derivatives and is locally L-Lipschitz continuous with respect to e_q.

Based on Lemmas 3 and 4, we derive the following theorem on query complexity.

Theorem 3 (Query Complexity)**.** Let k denote the total number of accumulated interactions. Assuming a proper initialization via an effective crowdsourcing pre-screening mechanism to guarantee identifiability, and provided that k is sufficiently large such that $\lambda > 4L\sqrt{\frac{\log k}{k}}$, Algorithm 2 achieves a total query complexity of

$O\left(\frac{\log n}{\lambda - 4L\sqrt{\frac{\log k}{k}}}\right)$ with probability at least $1 - O(1/k)$. Here, λ represents the lower bound established under ground-truth error rates in Lemma 2.

6 RELATED WORK

6.1 Interactive Graph Search

The interactive graph search problem studies target locating in a given hierarchy using an oracle and has been studied in recent years [8, 9, 30, 31, 34, 35, 51, 58, 60, 65]. Since Tao et al. [51] formally introduced the foundational heavy-path depth-first search (HPDFS) framework, a plethora of variants have been proposed. These extensions explore diverse dimensions, including multi-choice queries [30], budget-constrained multi-target search [58, 65], cost-effective search with priors [9], multiway algorithms [34], taciturn oracles [35], example-guided similarity search [60], and upward search [31]. However, most of these methods inherently assume a perfect oracle. Under the noisy setting, Cong et al. [8] propose OIGS. Nevertheless, it relies on an idealized static error prior that is typically unattainable in practice, which is a critical limitation our work explicitly addresses.

6.2 Active Learning and Truth Inference

Active learning and truth inference in crowdsourcing operate primarily with full data availability. Existing techniques range from direct computation (e.g., majority voting and medians) [17, 43] to probabilistic models that optimize and estimate single-value worker expertise [2, 28, 29, 64]. To capture complex dependencies among workers, tasks, and ground truths, advanced probabilistic graphical models have been developed by incorporating diverse skills [15, 36, 55], multi-dimensional worker probabilities [12, 25, 32, 57], and confusion matrices [11, 27, 45, 52]. Furthermore, several advanced models move beyond isolated worker evaluation to simultaneously infer task difficulty [15, 36, 55, 61].

6.3 Online Task Assignment

Online task assignment dynamically allocates a set of candidate questions to suitable workers [5, 20, 23, 24, 26, 33, 37, 63]. Common strategies evaluate workers by injecting gold-standard questions, thereby reserving subsequent tasks for highly accurate individuals [15, 20, 24, 26, 33]. Since optimal assignment is NP-hard, heuristic systems like QASCA [63] integrate evaluation metrics directly into the assignment process. Additionally, graph-based estimation models, such as the one proposed by Fan et al. [15], utilize task similarities to drive greedy assignment strategies.

7 EXPERIMENTS

7.1 Setup

We evaluate IGS-RTA by comparing it against six baseline IGS methods on two real-world datasets. We implement our approach in Python, and the source code of the baselines is kindly provided by their authors. Experiments are conducted on a single machine with Intel Xeon(R) 8377C@3.0GHz, 1TB RAM.

Table 2: Datasets statistics.

Type	Dataset	# nodes	Max Deg.	Height
DAG	*ImageNet*	27,714	402	13
Tree	*Amazon*	29,240	225	10

Datasets. We conduct our evaluations on two datasets, which are widely used in previous work [8, 9, 30, 35, 51, 58, 60, 65]. Table 2 summarizes the statistics of the two datasets.

- *ImageNet* [13]. This is a large-scale image dataset organized hierarchically based on the structure of WordNet [16]. Each category within this hierarchy is represented as a tag referred to as a "synset", with subcategories listed within each tag.
- *Amazon* [22]. This dataset represents the product hierarchy at Amazon, organized in a tree structure. It includes details about products sold on the platform. Each record contains a field "categories", which can be interpreted as a path from the root of the hierarchy to the specific product category.

For both datasets, we randomly select 1000 tasks as the testing sets.

Query Assignment. On crowdsourcing platforms, each worker is assigned a unique Worker ID, which facilitates reliable tracking and identification across tasks. Additionally, worker inclusion and exclusion can be managed via *Qualifications*, a built-in mechanism that enables requesters to condition task access on prior participation or performance. Following standard crowdsourcing practices, we employ a fixed worker pool and assign queries to each worker.

Competing Algorithms. We compare our method against six baseline methods listed below.

- **Topdown**: The straightforward search that initiates from the root of the hierarchy.
- **IGS** [51]: The conventional method that extracts HPDFS trees using heavy-path decomposition and conducts binary search on the HPDFS trees.
- **AIGS** [9]: The cost-effective IGS algorithm that conducts binary searches based on the middle points, assuming prior distribution of the hierarchy is available.
- **TS-IGS** [60]: The recently proposed IGS method equipped with target sensitive binary searches.
- **IGS-Tacit** [35]: The recent IGS algorithm designed under a taciturn oracle.
- **OIGS** [8]: The state-of-the-art noisy IGS algorithm. It assumes the error rate e_u of the oracle incorrectly answering $Q_r(u)$ for each $u \in V$ is available in advance.

For the five methods, Topdown, IGS, AIGS, TS-IGS, and IGS-Tacit, which are designed under a perfect oracle, we adapt them to the noisy setting by incorporating majority voting [10, 50, 62], which is widely adopted in crowdsourcing platforms [8, 51].

Evaluation Metrics. We employ the following three metrics to evaluate the performance of each method.

- **Round**: The average number of interactions with the oracle required per task.

Table 3: Experimental results on the *ImageNet* dataset.

Adv. # : Total #	*TopDown*			*IGS*			*AIGS*			*TS-IGS*			*IGS-Tacit*			*OIGS*			IGS-RTA		
	Round	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*
1:3	14.49	43.47	1%	13.83	41.49	1%	15.07	45.21	1%	17.89	53.69	1%	8.01	60.49	1%	123.66	371.00	94%	60.01	180.03	93%
1:5	16.54	82.72	4%	16.97	84.87	3%	17.12	85.62	7%	19.68	98.42	4%	8.46	113.12	4%	78.20	391.01	95%	34.53	172.68	95%
2:5	13.47	67.35	0%	12.59	62.95	0%	13.97	69.87	0%	16.69	83.49	0%	7.90	94.81	0%	199.29	996.49	66%	39.76	198.81	93%
1:7	17.95	125.67	12%	19.05	133.35	9%	18.47	129.27	14%	21.38	149.68	8%	8.65	164.91	9%	64.33	450.34	95%	20.26	141.86	95%
2:7	15.06	105.43	2%	15.04	105.30	2%	15.79	110.50	3%	18.29	128.01	1%	8.07	145.52	1%	100.75	705.26	92%	27.80	194.61	93%
3:7	13.03	91.23	0%	11.69	81.82	0%	13.59	95.13	0%	16.39	114.79	0%	7.82	126.95	0%	232.59	1628.17	36%	30.76	215.36	88%

Table 4: Experimental results on the *Amazon* dataset.

Adv. # : Total #	*TopDown*			*IGS*			*AIGS*			*TS-IGS*			*IGS-Tacit*			*OIGS*			IGS-RTA		
	Round	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*	*Round*	*Cost*	*Accuracy*
1:3	14.90	44.70	0%	15.09	45.27	0%	13.54	40.65	3%	19.23	57.69	1%	7.05	40.35	1%	90.34	271.02	93%	50.11	150.34	92%
1:5	18.03	90.18	3%	17.98	89.91	2%	15.71	78.59	9%	21.91	109.58	3%	7.93	85.28	3%	60.00	300.04	94%	25.56	127.83	94%
2:5	13.58	67.93	0%	11.89	59.46	0%	11.33	56.66	1%	17.50	87.53	0%	6.74	58.47	0%	151.91	759.55	73%	25.81	129.07	95%
1:7	20.17	141.21	7%	20.94	146.58	6%	17.70	123.92	14%	24.30	170.12	5%	8.41	136.16	7%	49.64	347.51	95%	17.30	121.13	95%
2:7	16.41	114.92	2%	15.53	108.72	1%	14.15	99.02	5%	19.89	139.26	1%	7.31	101.49	2%	75.07	525.49	93%	20.92	146.48	94%
3:7	12.85	89.98	0%	10.85	75.99	0%	10.31	72.19	2%	16.82	117.78	0%	6.42	72.44	0%	179.39	1255.77	46%	22.41	156.87	91%

- **Cost**: The average total monetary cost to complete a task, assuming each worker response incurs a unit cost (e.g., $1).
- **Accuracy**: The fraction of correctly solved tasks, reflecting the effectiveness in noisy settings.

Note that IGS algorithms primarily aim to minimize oracle interactions to reduce monetary costs. Since crowdsourcing latency (e.g., recruitment and response collection) typically exceeds local computation by orders of magnitude, wall-clock time is rarely a bottleneck. In the experiments, the initialization priors and ground truth parameters for both the oracle expertise and task difficulties are set as follows.

- $\boldsymbol{\alpha}$: The prior for each α is set to 1, indicating that crowdworkers are expected to be reliable and non-adversarial since they are compensated per decision [8, 23, 40].
- $\boldsymbol{\alpha}^*$: The ground truth expertise of each worker, drawn from -1, 1, 3. Specifically, -1, 1, and 3 correspond to adversarial, normal, and skilled workers, respectively.
- $\boldsymbol{\beta}$: The prior for each β is set to 1, reflecting a medium difficulty for each $Q_r(u)$.
- $\boldsymbol{\beta}^*$: The ground truth for each node's difficulty β^* is determined by computing e^{δ_u} for each node u, where δ_u is sampled from the standard normal distribution.

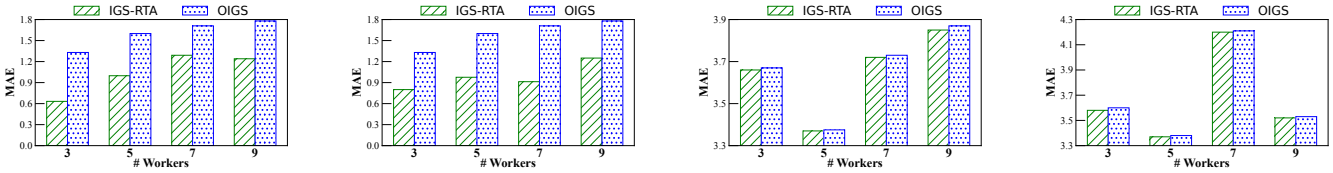
Note that while the ground truth parameters (i.e., $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$) govern the oracle's responses, they remain hidden from the algorithms. Unless otherwise stated, we initialize the oracle W as a group consisting of one skilled worker, one adversarial worker, and the rest as normal workers. To increase the ratio of adversarial workers, we replace normal workers with adversarial ones. For the baseline OIGS, which requires explicit error rates, we derive e_u for each node $u \in V$ via Equation (13) using the priors. The termination threshold ϵ is set to 0.9 by default.

7.2 Experimental Results

Exp-1: Performance Overview. Tables 3 and 4 report the overall performance, where IGS-RTA consistently outperforms all baselines. Methods based on the perfect oracle (Topdown, IGS, AIGS, TS-IGS, IGS-Tacit) fail to search effectively on *ImageNet*, with accuracies below 14% even under the lowest adversarial ratio. While majority voting provides marginal accuracy gains (7%–13%) as normal workers increase, it remains inadequate. Since IGS relies on sequential decisions, a single error can mislead the entire search. Interestingly, adding normal workers slightly increases query rounds for these baselines, as voting corrects early errors and allows deeper searches. Conversely, higher adversarial ratios prematurely terminate searches, severely degrading accuracy and group decision-making effectiveness.

In contrast, OIGS and IGS-RTA achieve significantly better results via Bayesian inference. Specifically, IGS-RTA improves accuracy by up to 52% while reducing costs by 7.5x compared to OIGS. OIGS relies on static, hard-to-obtain error priors, making it fragile in real-world scenarios. IGS-RTA overcomes this via online estimation, dynamically learning worker expertise and node difficulty, and an uncertainty-based strategy that enables holistic hierarchical reasoning rather than local posterior maximization. Furthermore, our estimation component potentially reduces costs by rejecting unreliable responses [40]. For OIGS and IGS-RTA, a lower adversarial ratio significantly reduces query rounds, as fewer errors allow the Bayesian inference to converge faster. Conversely, higher adversarial ratios inevitably increase costs and degrade accuracy, though IGS-RTA exhibits substantially stronger robustness than OIGS.

The results on the *Amazon* dataset are consistent with those on *ImageNet*, with IGS-RTA improving search accuracy by up to 45% and reducing cost by up to 8×. Table 5 shows the results of our ablation study, where OIGS denotes the node-wise strategy [8]



(a) α *MAE on ImageNet* (b) α *MAE on Amazon* (c) β *MAE on ImageNet* (d) β *MAE on Amazon*

Figure 4: Mean absolute errors of the estimated oracle expertise and task difficulty on both datasets.

Table 5: Ablation study on the *Amazon* dataset: OIGS denotes the node-wise strategy [8] and ES denotes our estimation component.

Method	Rounds	Accuracy
OIGS	119.68	82%
IGS-RTA w/o ES	120.33	86%
IGS-RTA	25.56	94%

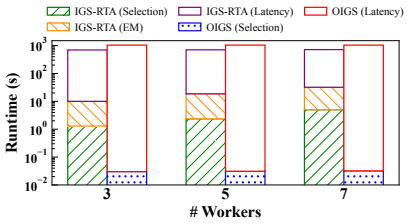


Figure 5: Average query runtime of IGS-RTA and OIGS on *Amazon*.

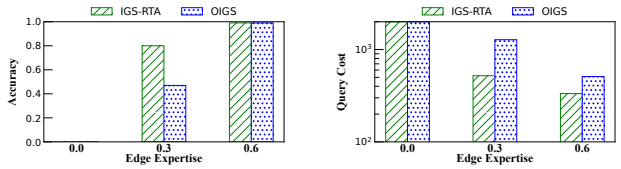
and ES denotes our estimation component. Our uncertainty-based querying strategy (i.e., IGS-RTA w/o ES) achieves a 4% accuracy improvement over the traditional node-wise approach while maintaining a comparable search cost. Furthermore, by integrating the estimation component, IGS-RTA can further reduce the total search cost by 4.7× while simultaneously improving accuracy by 8%. These results validate that both the uncertainty-based strategy and the inference component are indispensable to IGS-RTA's efficiency and robustness.

Exp-2: Estimation Performance. To further demonstrate the estimation performance of IGS-RTA, we compute the mean absolute error (MAE) between the estimated expertise (i.e., α) and the ground truth (i.e., α^*), as well as between the estimated difficulty (i.e., β) and its ground truth (i.e., β^*). Figure 4 shows the overall results. In general, IGS-RTA can effectively estimate both on the two datasets.

For the expertise estimation, as shown in Figure 4a and 4b, IGS-RTA achieves significantly smaller MAEs compared to OIGS on both datasets across varying numbers of workers. This improvement is remarkable and highlights the effectiveness of our online estimation techniques in recognizing workers with varying expertise and making better decisions by monitoring worker behaviors. Additionally, IGS-RTA successfully identifies all adversarial workers with negative parameters in various scenarios: 1 out of 3, 2 out of 5, 3 out of 7, and 4 out of 9. By distinguishing adversarial workers, normal workers, and skilled workers, IGS-RTA naturally assigns greater weight to the latter two groups while reducing trust in adversarial workers, thereby mitigating the negative effects brought by noises and reducing potential errors during the search.

On the other hand, IGS-RTA also improves difficulty estimation on both datasets. As shown in Figure 4c and 4d, IGS-RTA achieves a 0.01 reduction in MAE across both datasets with varying numbers of workers. This improvement is noteworthy given the following challenges. First, both datasets contain a large number of nodes (i.e., over 27,000), hence it is almost impossible to perform comprehensive estimation with limited costs. Second, queries are generated specifically to locate target nodes, meaning that most nodes do not have sufficient data for accurate estimation. According to Table 3 and 4, the average number of queries per task for IGS-RTA is less than 60, which is significantly smaller than the total number of nodes. Moreover, it is inevitable for some nodes to be queried repeatedly within a single task and across different tasks, further restricting the estimation scope of query node difficulties.

Exp-3: Empirical Runtime Scalability Analysis. We provide a detailed runtime comparison between IGS-RTA and OIGS on the *Amazon* dataset. Specifically, the initial setup consists of three workers: one expert, one normal, and one adversarial. We increase the group size by adding one normal worker and one adversarial worker in each incremental step. As shown in Figure 5, the selection time of OIGS remains constant regardless of the worker size. This is because OIGS relies on static priors and lacks the mechanism to dynamically update the expertise and difficulties. Consequently, it performs a single scan over the hierarchy to select query nodes based on its node-wise strategy. In contrast, IGS-RTA exhibits a higher but manageable computational overhead. Specifically, its query selection time increases from 1.3s to 5s as the worker size grows. This overhead arises because IGS-RTA dynamically refines error rates based on inferred worker expertise and task difficulty before each selection, causing the computational cost to scale with the number of workers. Additionally, as the worker size increases from 3 to 7, the EM convergence time rises from 8.7s to 27s. This trend is expected, as a higher ratio of adversarial workers necessitates more iterations for the model to converge and accurately identify malicious behaviors. On the other hand, IGS algorithms [8, 9, 30, 35, 51, 58, 60, 65] primarily aim to minimize oracle interactions to reduce monetary costs. In real-world crowdsourcing, the latency for worker recruitment and response collection (often taking hours or days) is orders



(a) *Accuracy* (b) *Cost*

Figure 6: Performance evaluation under edge cases on the *Amazon* dataset.

Table 6: Sensitivity analysis of different initializations on the *Amazon* dataset.

Init.	Rounds	Accuracy	MAE
positive	17.30	95%	0.33
random	26.49	96%	0.42
negative	20.23	95%	0.46

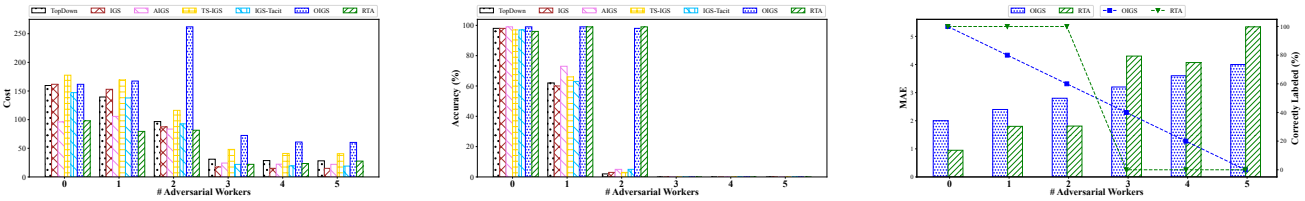
of magnitude higher than the local computation time. To further illustrate this, we have included a case study using Large Language Models (LLMs) as the oracle (see Exp-9 for more details). Even in this setup, the API interaction time exceeds 500 seconds per task (as shown by Latency in Figure 5), and this latency is expected to be substantially higher when engaging human crowdworkers, due to the inherent overhead of worker selection, task assignment, and result collection. Therefore, the local algorithmic execution time is rarely the performance bottleneck in practice.

Exp-4: Edge Case Analysis. We also include experiments to investigate the performance of IGS methods in the edge case where the bound becomes loose. Specifically, to drive the error rate towards 0.5, we approach the expertise parameter $\boldsymbol{\alpha}$ to 0. This represents a scenario where crowdworkers have little domain knowledge and answer queries via random guessing, yielding limited information gain. Figure 6 shows the experimental results. When the expertise is exactly 0, leading to the edge case where error rates are 0.5, both OIGS and IGS-RTA fail to search effectively. Specifically, they get stuck on the initial task, querying endlessly without reaching a termination condition. The intuition is straightforward: if crowdworkers answer queries via pure guessing, their responses provide no information gain to reduce the uncertainty of the search. Similarly, when the expertise parameter $\boldsymbol{\alpha}$ is set to 0.3 which is a less extreme scenario where the error rate is 0.45, both methods are negatively impacted. Specifically, OIGS still struggles to search effectively, yielding an accuracy below 50% and incurring a cost of over 1,000 interactions. In contrast, IGS-RTA demonstrates significantly higher stability, highlighting the robustness of our uncertainty-based selection and estimation components. We also evaluate a setting where $\boldsymbol{\alpha} = 0.6$ (i.e., an error rate of 0.35). Under this condition, both methods achieve high accuracy. However, IGS-RTA does so at a substantially lower cost.

Exp-5: Sensitivity Analysis. We conduct experiments to evaluate the stability of IGS-RTA across three different expertise initializations ($\boldsymbol{\alpha}$): (a) Positive initialization: We initialize $\boldsymbol{\alpha}$ to 1, reflecting the assumption that workers are pre-screened by the platform and motivated by financial compensation. (b) Random initialization: We draw $\boldsymbol{\alpha}$ from a normal distribution $\mathcal{N}(1, 1)$ to provide a random starting point. (c) Negative initialization: We initialize $\boldsymbol{\alpha}$ to -1 to test the robustness of IGS-RTA under an extreme worst-case scenario. Table 6 presents the results, demonstrating that IGS-RTA maintains stable performance across all three initializations. Specifically, it consistently achieves high accuracy regardless of the starting conditions. As expected, positive initialization incurs the lowest average cost (i.e., interaction rounds), while the other two require slightly more iterations. Furthermore, IGS-RTA successfully identifies adversarial workers in all three settings. The MAEs remain comparable: lowest under positive initialization and highest under negative initialization. This minor variance is reasonable, as the negative initialization deviates significantly from the ground-truth expertise.

Exp-6: Impact of Adversarial Workers. We also investigate scenarios with extreme noise levels by maintaining a fixed pool of 5 workers and incrementally replacing skilled workers with adversarial ones (from 0 to 5), and the results are shown in Figure 7. In general, performances of the seven methods can be generally divided into two cases: when the number of adversarial workers is in $[0, 2]$ and when it is in $[3, 5]$. The former case implies that the group is dominated by skilled workers while the latter indicates the dominance of adversarial workers. As shown in Figures 7a and 7b, IGS-RTA consistently incurs the lowest cost. When the number of adversarial workers is between 0 and 2, IGS-RTA and OIGS maintain robust accuracies exceeding 90%, whereas conventional baselines degrade rapidly. Notably, IGS-RTA achieves this resilience at a significantly lower cost than OIGS. However, when adversarial workers dominate (3 to 5), the accuracy of all methods understandably collapses. Interestingly, the costs also drop significantly in this phase, as the continuous erroneous answers trigger premature search termination. Figure 7c explains this phase shift by tracking the Mean Absolute Error (MAE) and worker identification accuracy. Our online inference detects adversaries based on their deviations from mainstream behavior. Thus, it performs accurately when skilled workers form the majority (≤ 2 adversaries) but inverts when adversaries dominate (≥ 3), misclassifying skilled workers as malicious. While resolving this inversion without prior knowledge of the adversarial ratio is non-trivial, such extreme scenarios (adversarial ratio $> 50\%$) are rare in practice due to platform-level filtering [40]. Furthermore, this inversion practically benefits the system by quickly terminating meaningless searches in highly corrupted environments, thereby minimizing resource waste.

Exp-7: Comparison with Active and Reinforcement learning. We compare IGS-RTA with AL-based and RL-based methods to further illustrate its superiority, with the results reported in Figure 8. Specifically, we introduce two active learning [62] and reinforcement learning [51] baselines: IGS-AL and IGS-UCB. We set the number of workers to 3, including 1 adversarial worker. In details, IGS-AL dynamically estimates each worker's accuracy based on the aggregated decisions per round, and updates their voting weights accordingly. For IGS-UCB, we assign a UCB score to each worker



(a) Cost (b) Accuracy (c) Estimation

Figure 7: Impact of adversarial workers on cost, accuracy, and real-time estimation on the *Amazon* dataset.

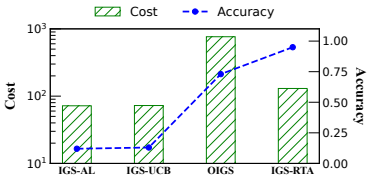


Figure 8: Performance evaluation of active learning and reinforcement learning extensions on the *Amazon* dataset.

Table 7: Case study on the *Amazon* dataset.

Method	Round	# API Calls	Accuracy (%)	Latency (s)
OIGS	97.23	291.69	73.33	1039.93
IGS-RTA	59.60	178.80	83.33	686.69

w, defined as $UCB_t(w) = R_t(w) + \sqrt{\frac{2\ln t}{N_t(w)}}$. Here, $R_t(w)$ denotes the number of correct answers provided by w prior to step t, and $N_t(w)$ represents the total number of times w has been selected. Notably, since the ground truth of queries is inaccessible during the search process, we compute the rewards based on the aggregated responses. As the results indicate, these straightforward combinations suffer from limited effectiveness. Specifically, IGS-AL fails to achieve satisfactory search effectiveness, highlighting the inherent drawback of naively applying group decisions to sequential decision-making. Similarly, IGS-UCB achieves search accuracy and costs comparable to those of IGS-AL. On the other hand, while OIGS outperforms AL/RF-based baselines in accuracy at the expense of higher costs, it is strictly dominated by IGS-RTA, which achieves superior accuracy with substantially lower overhead.

Exp-8: Case Study. To evaluate the practical performance of IGS-RTA, we conduct a case study using Large Language Models as the oracle, while keeping other experimental settings consistent with Amazon Mechanical Turk. Specifically, we employ GPT-4o, GPT-5, and Gemini-2.5-pro to simulate three workers with varying levels of expertise. To represent the least experienced worker, we prompt GPT-4o to answer each query randomly, while allowing the other two models to generate responses based on their standard capabilities. All models are accessed via their respective APIs. We conduct this experiment on the *Amazon* dataset comprising descriptions of 30 random products. As illustrated in Table 7, IGS-RTA significantly reduces the number of API calls compared to OIGS, thereby lowering the monetary cost, while simultaneously achieving a 10% improvement in accuracy. The performance gap is expected to be more pronounced as the adversarial ratio increases and the hierarchy expands. This case study validates the cost-effectiveness of IGS-RTA in handling real-world IGS tasks.

8 DISCUSSION

Expertise Modeling. Extending our framework with fine-grained, dynamic, and context-aware expertise modeling, such as capturing learning effects and domain specialization, is a promising direction. However, this introduces new challenges, including estimating dynamic models and precisely partitioning task hierarchies for context-awareness. Overcoming these challenges to further enhance IGS-RTA is an important avenue for future research.

Query Assignment. Matching task difficulty with worker expertise (e.g., assigning harder queries to experts) can further improve our framework. Implementing this requires highly accurate, real-time tracking of both metrics and optimal assignment strategies. Furthermore, such "difficulty-based" assignments necessitate dynamic cost models, such as tiered pricing based on worker expertise. Developing IGS methods with adaptive assignment remains a promising future direction.

9 CONCLUSION

This paper proposes IGS-RTA for the noisy interactive graph search problem. We introduce an uncertainty-based querying strategy that enables holistic hierarchical reasoning and achieves a theoretically guaranteed logarithmic query complexity. To address limited prior knowledge of the oracle, we dynamically estimate oracle expertise and task difficulty online. Extensive experiments validate the superiority of IGS-RTA over state-of-the-art IGS methods.

ACKNOWLEDGMENTS

This work is partially supported by National Key R&D Program of China under Grant No. 2024YFA1012700, by the National Natural Science Foundation of China (NSFC) under Grant No. 62402410, by Guangdong Provincial Project (No. 2023QN10X025), and by Research and Development of Heterogeneous Computing Interconnection and Scheduling Software Stack (YF202400000003). We also thank the anonymous reviewers for their insightful and constructive suggestions on an earlier version of the paper.

REFERENCES

[1] 2026. IGS-RTA: Technical Report. https://github.com/HandsBro/IGS-RTA.

[2] Bahadir Aydin, Yavuz Selim Yilmaz Yavuz Selim Yilmaz, Yaliang Li, Qi Li, Jing Gao, and Murat Demirbas. 2014. Crowdsourcing for multiple-choice question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28. 2946–2953.

[3] Yosi Ben-Asher, Eitan Farchi, and Ilan Newman. 1999. Optimal search in trees. *SIAM J. Comput.* 28, 6 (1999), 2090–2102.

[4] Tej Bahadur Chandra, Kesari Verma, Bikesh Kumar Singh, Deepak Jain, and Satyabhuwan Singh Netam. 2021. Coronavirus disease (COVID-19) detection in chest X-ray images using majority voting based classifier ensemble. *Expert systems with applications* 165 (2021), 113909.

[5] Xi Chen, Qihang Lin, and Dengyong Zhou. 2013. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *International conference on machine learning*. PMLR, 64–72.

[6] Ziqi Chen, Liangxiao Jiang, and Chaoqun Li. 2022. Label augmented and weighted majority voting for crowdsourcing. *Information Sciences* 606 (2022), 397–409.

[7] Reynold Cheng, Jinchuan Chen, and Xike Xie. 2008. Cleaning uncertain data with quality guarantees. *Proceedings of the VLDB Endowment* 1, 1 (2008), 722–735.

[8] Qianhao Cong, Jing Tang, Kai Han, Yuming Huang, Lei Chen, and Yeow Meng Chee. 2022. Noisy Interactive Graph Search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 231–240.

[9] Qianhao Cong, Jing Tang, Yuming Huang, Lei Chen, and Yeow Meng Chee. 2022. Cost-effective algorithms for average-case interactive graph search. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1152–1165.

[10] Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics* 10 (2022), 92–110.

[11] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28, 1 (1979), 20–28.

[12] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*. 469–478.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[14] Eyal Dushkin and Tova Milo. 2018. Top-k sorting under partial order information. In *Proceedings of the 2018 International Conference on Management of Data*. 1007–1019.

[15] Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. 2015. icrowd: An adaptive crowdsourcing framework. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1015–1030.

[16] Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

[17] Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. 2011. CrowdDB: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. 61–72.

[18] Chao Gao and Dengyong Zhou. 2013. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. *arXiv preprint arXiv:1310.5764* (2013).

[19] Aurélien Garivier and Olivier Cappé. 2011. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*. JMLR Workshop and Conference Proceedings, 359–376.

[20] Naman Goel and Boi Faltings. 2019. Crowdsourcing with fairness, diversity and budget constraints. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 297–304.

[21] Stephen Guo, Aditya Parameswaran, and Hector Garcia-Molina. 2012. So who won? Dynamic max discovery with the crowd. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*. 385–396.

[22] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[23] Danula Hettiachchi, Vassilis Kostakos, and Jorge Goncalves. 2022. A survey on task assignment in crowdsourcing. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–35.

[24] Panagiotis G Ipeirotis and Evgeniy Gabrilovich. 2014. Quizz: targeted crowdsourcing with a billion (potential) users. In *Proceedings of the 23rd international conference on World wide web*. 143–154.

[25] David Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative learning for reliable crowdsourcing systems. *Advances in neural information processing systems* 24 (2011).

[26] Asif R Khan and Hector Garcia-Molina. 2017. Crowddqs: Dynamic question selection in crowdsourcing systems. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1447–1462.

[27] Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *Artificial Intelligence and Statistics*. PMLR, 619–627.

[28] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. 2014. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment* 8, 4 (2014), 425–436.

[29] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. 2014. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 1187–1198.

[30] Yuanbing Li, Xian Wu, Yifei Jin, Jian Li, and Guoliang Li. 2020. Efficient algorithms for crowd-aided categorization. *Proceedings of the VLDB Endowment* 13, 8 (2020), 1221–1233.

[31] Han Linghu, Qianhao Cong, Yuming Huang, Shangqi Lu, Liang Feng, and Jing Tang. 2025. LLM-Powered Interactive Graph Search: A Scalable and Practical Approach. *Proceedings of the ACM on Management of Data* 3, 6 (2025), 1–26.

[32] Qiang Liu, Jian Peng, and Alexander T Ihler. 2012. Variational inference for crowdsourcing. *Advances in neural information processing systems* 25 (2012).

[33] Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, and Meihui Zhang. 2012. Cdas: a crowdsourcing data analytics system. *arXiv preprint arXiv:1207.0143* (2012).

[34] Shangqi Lu, Wim Martens, Matthias Niewerth, and Yufei Tao. 2023. Partial Order Multiway Search. *ACM Transactions on Database Systems* 48, 4 (2023), 1–31.

[35] Shangqi Lu, Ru Wang, and Yufei Tao. 2025. Interactive Graph Search Made Simple. *Proceedings of the ACM on Management of Data* 3, 3 (2025), 1–25.

[36] Fenglong Ma, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. 2015. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 745–754.

[37] William Moulton Marston. 2013. *Emotions of normal people*. Routledge.

[38] Luyi Mo, Reynold Cheng, Xiang Li, David W Cheung, and Xuan S Yang. 2013. Cleaning uncertain data for top-k queries. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. IEEE, 134–145.

[39] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 188–197.

[40] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. 2010. Running experiments on amazon mechanical turk. *Judgment and Decision making* 5, 5 (2010), 411–419.

[41] Aditya Parameswaran, Stephen Boyd, Hector Garcia-Molina, Ashish Gupta, Neoklis Polyzotis, and Jennifer Widom. 2014. Optimal crowd-powered rating and filtering algorithms. *Proceedings of the VLDB Endowment* 7, 9 (2014), 685–696.

[42] Aditya G Parameswaran, Hector Garcia-Molina, Hyunjung Park, Neoklis Polyzotis, Aditya Ramesh, and Jennifer Widom. 2012. Crowdscreen: algorithms for filtering data with humans. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. 361–372.

[43] Aditya Ganesh Parameswaran, Hyunjung Park, Hector Garcia-Molina, Neoklis Polyzotis, and Jennifer Widom. 2012. Deco: declarative crowdsourcing. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. 1203–1212.

[44] Runwen Qiu and Jing Tang. 2025. Efficient Approximate Nearest Neighbor Search via Hemi-Sphere Centroids Graph. *Proceedings of the ACM on Management of Data* 3, 6 (2025), 1–26.

[45] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *Journal of machine learning research* 11, 4 (2010).

[46] Daniel D Sleator and Robert Endre Tarjan. 1981. A data structure for dynamic trees. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*. 114–122.

[47] Yifan Song, Xiaolong Chen, Wenqing Lin, Jia Li, Chen Zhang, Yan Zhou, Lei Chen, and Jing Tang. 2024. Efficient Graph Embedding Generation and Update for Large-Scale Temporal Graph. *Proceedings of the VLDB Endowment* 18, 4 (2024), 929–942.

[48] Yifan Song, Xingjian Tao, Zhicheng Yang, Yihong Luo, and Jing Tang. 2026. EHRAG: Bridging Semantic Gaps in Lightweight GraphRAG via Hybrid Hypergraph Construction and Retrieval. In *Findings of the Association for Computational Linguistics: ACL 2026*. 24638–24652.

[49] Yifan Song, Fenglin Yu, Yihong Luo, Xingjian Tao, Siya Qiu, Kai Han, and Jing Tang. 2026. Mitigating Structural Overfitting: A Distribution-Aware Rectification Framework for Missing Feature Imputation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Australia) *(SIGIR '26)*. Association for Computing Machinery, New York, NY, USA, 1644–1655. https://doi.org/10.1145/3805712.3809587

[50] Gopal S Tandel, Ashish Tiwari, and Omprakash G Kakde. 2021. Performance optimisation of deep learning models using majority voting algorithm for brain tumour classification. *Computers in Biology and Medicine* 135 (2021), 104564.

[51] Yufei Tao, Yuanbing Li, and Guoliang Li. 2019. Interactive graph search. In *Proceedings of the 2019 International Conference on Management of Data*. 1393–1410.

[52] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. 2014. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*. 155–164.

[53] Vasilis Verroios and Hector Garcia-Molina. 2015. Entity resolution with crowd errors. In *2015 IEEE 31st international conference on data engineering*. IEEE, 219–230.

[54] Vasilis Verroios, Hector Garcia-Molina, and Yannis Papakonstantinou. 2017. Waldo: An adaptive human interface for crowd entity resolution. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1133–1148.

[55] Peter Welinder, Steve Branson, Pietro Perona, and Serge Belongie. 2010. The multidimensional wisdom of crowds. *Advances in neural information processing systems* 23 (2010).

[56] Steven Euijong Whang, Peter Lofgren, and Hector Garcia-Molina. 2013. Question selection for crowd entity resolution. *Proceedings of the VLDB Endowment* 6, 6 (2013), 349–360.

[57] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in neural information processing systems* 22 (2009).

[58] Zheng Wu, Xuliang Zhu, Yixiang Fang, Jianliang Xu, and Xin Huang. 2024. Interactive Graph Search for Multiple Targets on DAGs. *Proceedings of the VLDB Endowment* 18, 4 (2024), 1091–1103.

[59] Chen Jason Zhang, Lei Chen, Yongxin Tong, and Zheng Liu. 2015. Cleaning uncertain data with a noisy crowd. In *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 6–17.

[60] Zhuowei Zhao, Junhao Gan, Jianzhong Qi, and Zhifeng Bao. 2024. Efficient Example-Guided Interactive Graph Search. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 342–354.

[61] Zhou Zhao, Furu Wei, Ming Zhou, Weikeng Chen, and Wilfred Ng. 2015. Crowd-Selection Query Processing in Crowdsourcing Databases: A Task-Driven Approach.. In *EDBT*. 397–408.

[62] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment* 10, 5 (2017), 541–552.

[63] Yudian Zheng, Jiannan Wang, Guoliang Li, Reynold Cheng, and Jianhua Feng. 2015. QASCA: A quality-aware task assignment system for crowdsourcing applications. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1031–1046.

[64] Dengyong Zhou, Sumit Basu, Yi Mao, and John Platt. 2012. Learning from the wisdom of crowds by minimax entropy. *Advances in neural information processing systems* 25 (2012).

[65] Xuliang Zhu, Xin Huang, Byron Choi, Jiaxin Jiang, Zhaonian Zou, and Jianliang Xu. 2021. Budget constrained interactive search for multiple targets. *Proceedings of the VLDB Endowment* 14, 6 (2021), 890–902.