



Robust Truth Inference in Crowdsourcing under Adversarial Attacks via Graph Neural Networks

Authors: Arif Dağ¹, Simay Sahin², Mehmet Karaköse¹

¹ *Fırat University, Elazığ, Türkiye*

² *Tilburg University, Tilburg, The Netherlands*

Corresponding Author: dg.arifdag@gmail.com

Received: February, 2026 Published: March, 2026

ARTICLE INFO

Keywords:

Adversarial attacks; Crowdsourcing; Graph neural networks; Label aggregation; Truth inference

© 2026 by the Authors. This open-access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license, making research freely available to the public and supporting a greater global exchange of knowledge and human experiments.



ABSTRACT

Crowdsourcing is widely used to collect labels for machine learning, but open participation also allows spammers, colluders, and Sybil-style attackers to create persuasive yet incorrect consensus. This paper studies robust truth inference under such attacks with a label-aware graph neural network that represents workers and tasks as a bipartite graph. The proposed framework combines edge-label-aware message passing, an auxiliary worker-trust head, and adaptive use of task-content features. Rather than relying on worker-maliciousness labels during training, the primary model is trained only with task supervision and selects between content-enabled and no-content variants on validation data. Evaluation uses a held-out train/validation/test protocol on simulated cifar_binary, imdb, and newsgroups labeling tasks under realistic and oracle threat models. We compare against majority voting, weighted majority voting, Dawid-Skene, the binary KOS baseline where applicable, MMSR, content-only baselines, and collusion/Sybil defenses adapted from prior work. We also validate on two public real crowdsourcing benchmarks, relevance-2 and relevance-5. On these real benchmarks, the adaptive GNN reaches 81.85% and 90.80% accuracy, respectively, and significantly outperforms the classical and robust aggregation baselines considered in this study. In simulation, the method is competitive with the strongest fair content-aware baseline, improves substantially over a fixed-content GNN on newsgroups, and remains stronger than classical crowd-only aggregation on the attack-sensitive cifar binary setting. Ablation analysis shows that task content helps on cifar binary and imdb but hurts on newsgroups, motivating adaptive content selection instead of a fixed multimodal design. Overall, the results support a qualified claim: graph-based robust aggregation can work without worker-maliciousness labels, but its gains are dataset-dependent and are strongest when relational evidence and task semantics complement each other.

1.0 Introduction

Crowdsourcing has become a key mechanism for producing large-scale labelled datasets, enabling many machine learning pipelines to be trained with human annotations at relatively low cost and high speed. At the same time, the openness of crowdsourcing platforms creates a persistent quality paradox: the workforce can include both honest contributors and unreliable or malicious workers, and the platform must estimate correct labels from conflicting submissions. Therefore, most systems rely on truth inference (label aggregation) to infer the true label for each task and model worker reliability (Jin et al. 2020; Zheng et al. 2017).

Crowdsourcing platforms mediate interactions between requesters who publish tasks and a large pool of workers who submit responses (Figure 1). In this setting, the platform typically aggregates multiple noisy labels per item and returns feedback to the requester, often under limited supervision. The open nature of participation makes crowdsourcing attractive and scalable, but it also creates opportunities for strategic manipulation through

coordinated or Sybil-style participation. As illustrated in Figure 1 below, malicious workers can inject correlated responses that appear consistent yet push the inferred consensus away from the true label. These risks motivate robust inference mechanisms that leverage both behavioral signals and auxiliary information to mitigate adversarial influence.

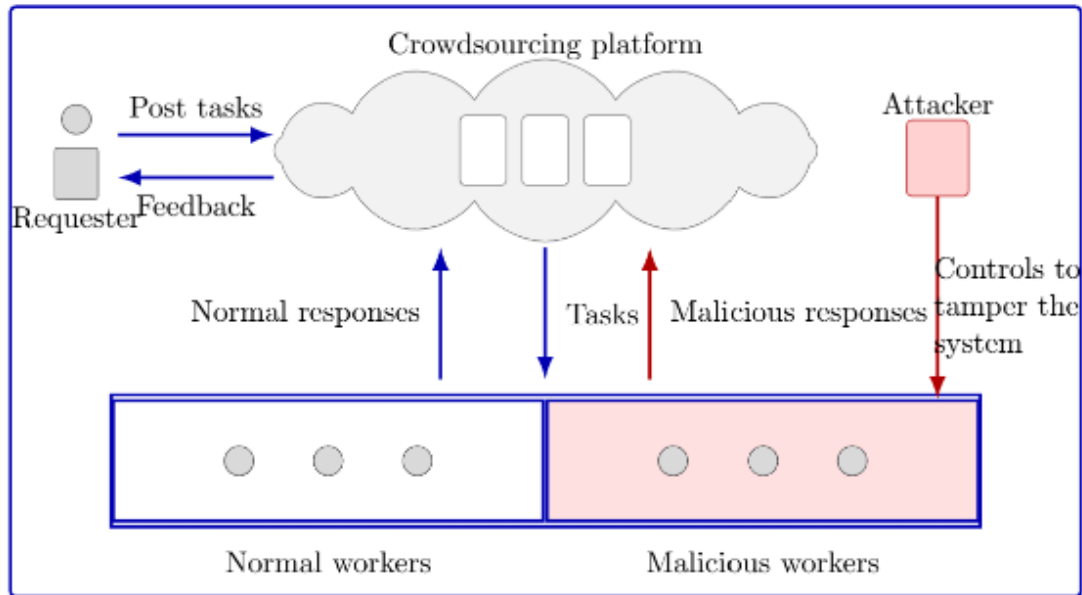


Figure 1: Conceptual view of a crowdsourcing platform with normal and malicious workers.

A central difficulty is that adversarial behavior in crowdsourcing is rarely purely random. Besides accidental noise and spammers (Gadiraju et al. 2015), attackers can deliberately poison labels to degrade aggregation and downstream learning (Checco, Bates, and Demartini 2020; P. P. Chen, Sun, and Z. Chen 2021; Tahmasebian et al. 2020; Miao et al. 2018). Advanced threats such as collusion and Sybil-style participation can generate highly correlated responses that mimic high confidence agreement while pushing the inferred ground truth in the wrong direction (P. P. Chen, Sun, Fang, et al. 2018; Y. Wang, K. Wang, and Miao 2020). These behaviors systematically violate the assumptions behind simple baselines like majority voting (Tao et al. 2019) and can also break the independence assumptions that underlie many probabilistic models. Related robustness concerns also arise in sensitive domains, such as medical imaging, where deepfake manipulation motivates the development of efficient detection methods (Karaköse, Yetiş, and Çeçen 2024).

These observations motivate a relational perspective. The crowdsourcing process naturally forms a bipartite interaction graph where worker nodes and task nodes are connected by edges representing submitted labels. Coordinated adversaries are not only bad individuals; they are often identifiable as structural or group-level patterns (e.g., unusually dense agreement clusters or repeated co-labeling behavior). Graph Neural Networks (GNNs) are designed to learn from such graph-structured data via iterative message passing (Kipf and Welling 2017; Gilmer et al. 2017; Z. Wu et al. 2021; Zhou et al. 2020). In addition, incorporating task-specific content (e.g., ResNet features for images (He et al. 2016) or TF-IDF-style text representations) provides a semantic anchor that helps prevent the system from being driven solely by an attacker-induced consensus.

In this paper, we propose a GNN-based framework for robust truth inference under adversarial attacks. The primary model predicts task labels without requiring ground-truth worker-maliciousness labels during training, while an auxiliary worker-trust head is retained as a risk-scoring signal rather than the main source of supervision.

Our main contributions in this study can be summarized as follows:

- We formulate robust truth inference in adversarial crowdsourcing as a label-aware relational learning problem on a worker–task interaction graph, and we introduce an adaptive-content GNN variant that can choose whether task features should be used for a given dataset.
- We use a held-out evaluation protocol with train/validation/test splits, realistic and oracle threat models, fair content-aware baselines, and paper-faithful adversarial baselines adapted from prior collusion and Sybil-defense work.
- We provide evidence from both simulation and two public real crowdsourcing benchmarks, showing that the proposed method is strong on real answer-aggregation tasks and competitive, but not uniformly dominant, in controlled adversarial simulation.

2.0 Literature Review

2.1 Truth inference and reliability modeling

Early aggregation approaches range from majority voting to probabilistic models that estimate worker error patterns, such as the classic EM-based method of Dawid and Skene (1979). Subsequent work studied iterative and optimization-based aggregation and highlighted when and why aggregation succeeds (Zheng et al. 2017; Whitehill et al. 2009; Karger, Oh, and Shah 2011). Robust spectral alternatives have also been proposed for adversarial settings; for example, M-MSR estimates worker skill via robust rank-one matrix completion and then performs weighted aggregation (Ma and Olshevsky 2020). Despite strong performance under random noise, these models can be brittle when adversaries adapt their behavior.

2.2 Adversarial and strategic behavior in crowdsourcing

Adversarial threats in crowdsourcing have been studied from both algorithmic and system perspectives. Checco et al. analyze attacks on crowdsourcing quality control and discuss the limitations of standard aggregation under strategic manipulation (Checco, Bates, and Demartini 2020). Several works study data poisoning in crowdsourcing settings and show how attackers can design responses to maximize harm while remaining difficult to detect (P. P. Chen, Sun, and Z. Chen 2021; Tahmasebian et al. 2020; Miao et al. 2018). Beyond individual poisoning, collusion and Sybil-style behaviors introduce highly correlated submissions; collusion-proof inference and Sybil-resilient truth discovery have therefore been investigated using statistical and system-level defenses (P. P. Chen, Sun, Fang, et al. 2018; Y. Wang, K. Wang, and Miao 2020; Song, Liu, and Zhang 2021). In addition, worker classification mechanisms and adaptive platform workflows (e.g., trust scoring and task allocation) have been proposed as practical mitigation strategies (Kurup and Sajeew 2025).

2.3 Graph-based learning for security and trust

Graph modeling is well suited for capturing relational patterns, including abnormal substructures, and coordinated behavior. Surveys describe how GNNs learn node and edge representations via message passing and have become a standard tool for learning from graph-structured data (Z. Wu et al., 2021; Zhou et al., 2020). More recently, GNN-based methods have been used for graph anomaly detection and trust evaluation (Kim et al. 2022; Luo et al. 2025), and they have also been explored for Sybil detection (Heeb, Plesner, and Wattenhofer 2024). In crowdsourcing specifically, graph-based aggregation methods can exploit latent worker–task correlations and richer relation structure to improve label aggregation quality (H. Wu et al. 2022). Recent GNN aggregation work has also shown that limited truth injection can materially improve label aggregation accuracy (Ying et al. 2024). Relative to that line of work, the present study does not claim a new generic aggregation primitive; instead, it focuses on adversarial robustness, label-aware bipartite message passing, adaptive content use, and evaluation under explicit no-worker-label, weak-gold,

realistic-attack, and oracle-attack regimes. Public tooling has also made shared answer-aggregation benchmarks easier to reuse in reproducible evaluation pipelines, notably through Crowd-Kit (Ustalov, Pavlichenko, and Tseitlin 2024). Similar coordination and trust challenges arise in multi-agent reinforcement learning and swarm robotics, where distributed DRL methods are used for collaborative behavior (Tan and Karaköse 2021; Bar and Karaköse 2024).

3.0 Methodology

3.1 System Overview

Figure 2 illustrates the overall workflow considered in this study. First, the platform assigns tasks to workers and collects their labels. Next, a GNN-based inference module aggregates labels while estimating worker trustworthiness. The resulting trust scores can optionally be fed back into an active task assignment module to steer future task allocation.

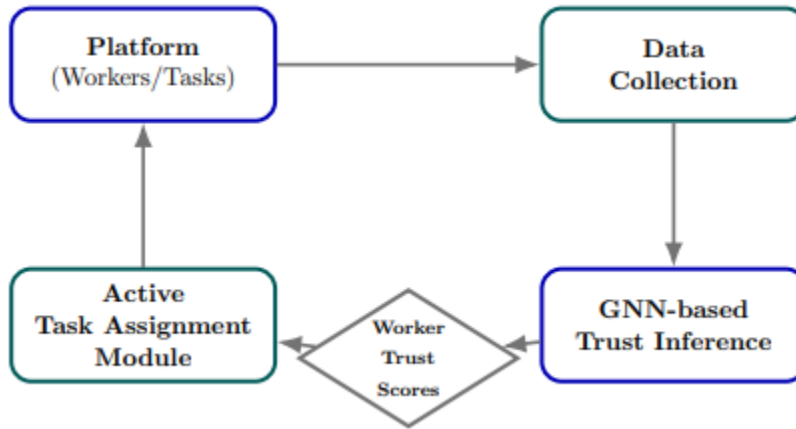


Figure 2: End-to-end workflow: platform (workers/tasks), data collection, GNN-based trust inference, and optional active task assignment using estimated worker trust scores.

3.2 Graph Construction

We represent the crowdsourcing process as a heterogeneous bipartite graph $G = (V, E)$ with two node types: worker nodes V_W and task nodes V_T . An edge $(w_i, t_j) \in E$ exists if worker w_i provides a label for task t_j . Each edge carries an observed label $\ell_{ij} \in \{1, \dots, C\}$ (or a one-hot vector), which serves as the main supervision signal for the aggregation model.

This representation makes adversarial coordination observable: collusion and Sybil-style behavior tend to produce groups of workers that co-label many tasks and exhibit unusually high agreement patterns, which become structural signatures on G .

3.3 Node and Edge Features

Edge features. For a C -class task, the observed label on an edge is encoded as a one-hot vector $\mathbf{x}_{ij}^e \in \{0,1\}^C$. For binary tasks, we use $C = 2$.

Task node features. Task features provide a semantic “reality anchor” to reduce the risk of converging to an attacker-induced consensus. For image tasks, we extract a fixed-length embedding using a pretrained CNN backbone (e.g., ResNet) (He et al. 2016); for text tasks, we use bag-of-words style representations such as TF-IDF vectors. We denote the initial task feature as $\mathbf{x}_j^t \in \mathbb{R}^{d_t}$.

Worker node features. Workers are initialized with learnable embeddings $\mathbf{x}_i^w \in \mathbb{R}^{d_w}$ (randomly initialized and learned end-to-end). These embeddings are updated by message passing and are expected to encode behavioral patterns such as reliability and coordination.

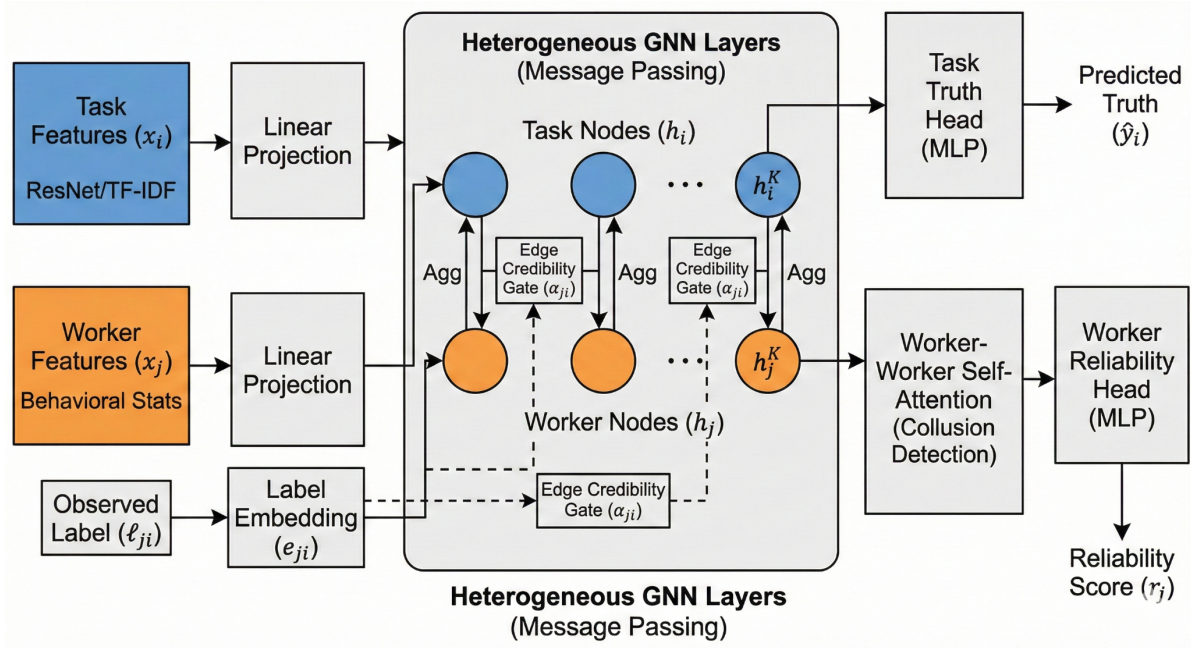


Figure 3: Schematic heterogeneous GNN architecture for robust truth inference.

The diagram above summarizes the core label-aware bipartite message-passing pipeline with edge gating, worker-worker attention, and task / worker readout heads. In the implemented model, worker-side inputs are learnable embeddings rather than fixed precomputed behavioral statistics, task content features are used only when available, and the worker-trust head is auxiliary in the primary no-worker-label setting.

3.4 GNN-based Truth Inference

Figure 3 summarizes the implemented architecture, including the label-aware bipartite message-passing layers, the edge-credibility gate, the worker-worker attention block, and the task / worker readout heads.

We learn node representations via message passing on G (Gilmer et al. 2017; Wu et al. 2021). Throughout, we index workers by i and tasks by j . Implementation-wise, task features and worker embeddings are first mapped to a common hidden dimension via Linear projections. We then apply a PyTorch Geometric HeteroConv with SAGEConv (GraphSAGE) on both relations (worker \rightarrow task and task \rightarrow worker), using mean aggregation.

Let $\mathbf{h}_i^{(0)}$ and $\mathbf{h}_j^{(0)}$ denote the projected worker and task embeddings, respectively. At layer k , the update aggregates label-aware neighbor messages by mean over the corresponding relation:

$$\begin{aligned} a_{ij}^{(k)} &= \sigma \left(MLP_{\alpha} \left[\mathbf{h}_i^{(k-1)} \parallel \mathbf{h}_j^{(k-1)} \parallel \mathbf{x}_{ij}^e \right] \right), \\ \mathbf{m}_{i \rightarrow j}^{(k)} &= a_{ij}^{(k)} \phi_t \left(\mathbf{h}_i^{(k-1)}, \mathbf{x}_{ij}^e \right), \\ \mathbf{h}_j^{(k)} &= \psi_t \left(\mathbf{h}_j^{(k-1)}, \frac{1}{|\mathcal{N}(j)|} \sum_{i \in \mathcal{N}(j)} \mathbf{m}_{i \rightarrow j}^{(k)} \right), \end{aligned}$$

and symmetrically

$$\begin{aligned} \mathbf{m}_{j \rightarrow i}^{(k)} &= a_{ij}^{(k)} \phi_w \left(\mathbf{h}_j^{(k-1)}, \mathbf{x}_{ij}^e \right), \\ \mathbf{h}_i^{(k)} &= \psi_w \left(\mathbf{h}_i^{(k-1)}, \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{j \rightarrow i}^{(k)} \right). \end{aligned}$$

Edge “credibility attention”. The scalar gate $a_{ij}^{(k)} \in (0, 1)$ is applied directly to the message on edge (i, j) , so suspicious labels can be attenuated rather than treated identically to all other observations. In this formulation, the gate is part of the actual message-passing computation rather than a diagnostic quantity.

Worker–worker collusion module. To capture coordinated behavior beyond bipartite propagation, we apply a worker–worker self-attention block using MultiheadAttention (scaled dot-product attention with softmax over workers). This auxiliary block helps encode repeated co-labeling and agreement patterns that may not be visible from a single worker–task edge in isolation.

Mapping from equations to implementation. In the implemented forward pass, ϕ_t and ϕ_w correspond to label-aware message maps from the sender representation and one-hot edge label into the shared hidden space, while ψ_t and ψ_w correspond to the GraphSAGE-style update functions realized by the relation-specific SAGEConv blocks inside HeteroConv. Each pass therefore follows the same sequence used in the code: (i) project task features and worker embeddings to the hidden space, (ii) compute a_{ij} and gated worker→task / task→worker messages, (iii) apply the bipartite updates and then the worker–worker attention block, and (iv) read out task logits and optional worker-trust scores from the final node states.

3.5 Outputs and Learning Objectives

The model produces two outputs: (i) a task label distribution $\hat{y}_j = \text{softmax}(\text{MLP}_t(\mathbf{h}_j^{(K)}))$ and (ii) a worker trust score $\hat{z}_i = \sigma(\text{MLP}_w(\mathbf{h}_i^{(K)}))$.

For tasks with known ground truth on the training split, we optimize a task-level cross-entropy loss L_{task} . The primary no-worker-label model, GNN_NO_WORKER_GT, sets $\lambda = 0$ and does not use ground-truth worker-maliciousness labels. A weak-gold variant (GNN_WEAK_GOLD) uses a small gold-task subset for calibration, while a fully supervised worker-label variant is retained only as a controlled upper bound. The total objective is

$$L = L_{task} + \lambda L_{worker},$$

where L_{worker} is optional and $\lambda \geq 0$.

When $\lambda = 0$, the worker-trust head is not directly supervised. In that primary setting, \hat{z}_i should therefore be interpreted as an auxiliary exploratory risk score derived from the shared representation rather than as a calibrated malicious-worker probability.

Because the effect of task content is dataset-dependent, we also define an adaptive-content variant (GNN_ADAPTIVE_CONTENT). In this setup, two models are trained, one with task content and one without, and the better variant is selected using validation accuracy. This adaptive model is the primary method reported in the experiments.

3.6 Adversarial Worker Simulation

To evaluate robustness under controlled conditions, we inject synthetic adversarial workers into an otherwise honest population. We implement behaviors motivated by the adversarial crowdsourcing literature (Checco, Bates, and Demartini 2020; Miao et al. 2018; P. P. Chen, Sun, Fang, et al. 2018) and evaluate two knowledge regimes:

Realistic attackers: no access to the true label, using heuristic behaviors such as random labeling, constant-label spam, and partially targeted corruption.

Oracle attackers: access to the true label, enabling flip, item-dependent, and coordinated collusion strategies that act as upper-bound stress tests.

We vary the adversarial ratio $\mu \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ to quantify performance degradation under increasing attack strength.

3.7 Training and Inference

We train the model end-to-end with full-graph optimization on the training split, tune hyperparameters and content usage on the validation split, and report only held-out test performance. At inference time, the aggregated label for each task is $\arg \max_c \hat{y}_{j,c}$. The worker-head output \hat{z}_i can be inspected as an auxiliary soft risk indicator, but because the primary models do not use direct worker-maliciousness supervision and no real malicious worker calibration set is available, we do not present it as a fully validated deployment score in the main empirical claims.

3.8 Experimental Setup

We conduct experiments in a controlled simulation setting where each task has an underlying ground-truth label and worker responses are generated by a mixture of honest and adversarial worker models. All primary simulation results use held-out train/validation/test splits and report the realistic attack regime; oracle-attack results are used as additional stress tests.

Datasets. We consider three representative labeling scenarios: (i) `cifar_binary` (CIFAR-10 cat vs. dog subset) (Krizhevsky 2009), (ii) `imdb` (binary sentiment classification) (Maas et al. 2011), and (iii) `newsgroups` (multi-class text classification subset) (Lang 1995).

Baselines. We compare against majority voting (MV), weighted majority voting (WMV), Dawid–Skene (D&S) (Dawid and Skene 1979), the binary KOS message-passing baseline (Karger, Oh, and Shah 2011), and the robust spectral baseline MMSR (Ma and Olshevsky 2020). To separate the contributions of task content from those of graph aggregation, we also include content-aware baselines: `TASK_ONLY`, `TASK_PLUS_MV`, and `TASK_PLUS_DS`. Finally, we report adapted adversarial defenses: `PROCAP_BIC`, a pairwise collusion-screening baseline in the style of Song et al. (Song, Liu, and Zhang 2021), and `TDSSA_STYLE`, a standard-task/Sybil defense baseline in the style of Wang et al. (Y. Wang, K. Wang, and Miao 2020). `TDSSA_STYLE` is fit using train-side standard tasks only and is therefore treated as a weak-gold comparator rather than a no-gold baseline. The primary neural comparisons are `GNN_NO_WORKER_GT`, `GNN_WEAK_GOLD`, and `GNN_ADAPTIVE_CONTENT`; the fully supervised worker-label variant is reported only as an upper-bound diagnostic.

Metrics. Task-level aggregation quality is reported as accuracy and macro-F1. For paired comparisons on simulated settings, we summarize mean accuracy differences between the primary method and selected baselines. Unless stated otherwise, reported Δ values are absolute accuracy differences (e.g., $\Delta = 0.129$ corresponds to 12.9 percentage points). For the real crowdsourcing benchmarks, we additionally use paired bootstrap confidence intervals and p-values on held-out task predictions. Because these paired tests are reported for a targeted set of planned comparisons rather than for a full familywise screening procedure, the p-values are unadjusted and interpreted descriptively, along with effect sizes and confidence intervals.

Real crowdsourcing benchmarks. To reduce reliance on simulation alone, we also evaluate on two public answer-aggregation benchmarks distributed through Crowd-Kit (Ustalov, Pavlichenko, and Tseitlin 2024): `relevance-2` (binary relevance judgments) and `relevance-5` (five-class relevance judgments). These datasets provide real crowd annotations and known ground truth, enabling out-of-simulator validation.

Reproducibility: architecture, optimization, and split policy. The main neural model uses two label-aware worker–task message-passing layers with hidden size 96, a label embedding width of 24, edge gating, and a four-head worker-attention block. Training uses Adam with learning rate 3×10^{-3} , weight decay 10^{-4} , gradient clipping at 1.0, a maximum of 40 epochs, and early stopping on validation accuracy with patience 6. For simulated datasets, the original training partition is split stratified 80/20 into train/validation subsets and the original test partition is kept as held-out test data. For the weak-gold regime, 10% of training tasks are selected as a stratified gold subset and the remaining training tasks use a D&S-

derived pseudo-target term. For the Crowd-Kit benchmarks, tasks are split stratified into 64/16/20 train/validation/test subsets with seed 42; worker identities are aligned across splits for transfer-style baselines; and task-content features are disabled because the public answer files do not provide the corresponding raw content features used in the simulation datasets. The relevance-5 benchmark is capped at 20,000 tasks for runtime control.

Reproducibility: seeds, repeats, and confidence intervals. Unless stated otherwise, global random seeds are set to 42 (NumPy, TensorFlow, and Python’s random); the simulator uses `default_rng(42)`. For Step 5 accuracy confidence intervals, we use a nonparametric bootstrap over tasks with $n_{boot} = 1000$ resamples and a 95% CI ($\alpha = 0.05$), using RNG seed 42. For Step 6 regime sweeps, each configuration is repeated with `SEEDS = {101,102,103,104,105}`; we report the mean across seeds together with an approximate 95% CI computed as $1.96 \text{ std}/\sqrt{n}$ (with $n = 5$), and we additionally report per-setting bootstrap CIs where applicable.

3.8.1 Simulator Details

This subsection specifies the simulator configuration used to generate worker labels and the resulting worker–task graph.

Datasets and task features. We use fixed train/test splits and treat each item as a crowdsourcing task. Task content features are extracted once and kept fixed during truth inference.

`cifar_binary` (Krizhevsky 2009): 10,000 training tasks and 2,000 test tasks, $C = 2$ (cat/dog), task feature dimension $d_t = 2048$ (ResNet).

`imdb` (Maas et al. 2011): 3,000 training tasks and 500 test tasks, $C = 2$ (negative/positive), $d_t = 1000$ (TF-IDF).

`newsgroups` (Lang 1995) (4-class subset): 2,257 training tasks and 1,502 test tasks, $C = 4$ (alt.atheism, comp.graphics, sci.med, soc.religion.christian), $d_t = 1000$ (TF-IDF).

Crowdsourcing graph size and labeling density. Unless stated otherwise, each simulation instance contains $|V_W| = 30$ persistent workers shared across the train/validation/test splits. For each robustness setting μ , we create $|W_{adv}| = \lfloor \mu |V_W| \rfloor$ adversarial workers and $|W_{hon}| = |V_W| - |W_{adv}|$ honest workers. Each worker labels a random 60–90% subset of tasks, which yields a dense worker–task graph while preserving worker identity across splits for the transfer-style baselines. Accordingly, the simulator is intended to represent a dense, stable-assignment regime rather than a sparse or high-churn platform.

Task difficulty. Each task t_j is assigned a scalar difficulty $d_j \in [0,1]$ (higher means harder), sampled i.i.d. from `Uniform(0,1)`. This difficulty affects both honest worker noise and some adversary strategies.

Honest worker model. Each honest worker w_i is assigned a base reliability above the chance level $1/C$, sampled around a dataset-level honest-accuracy target. For task t_j with difficulty d_j and worker expertise e_j , the probability of returning the correct label is modeled as

$$p_{correct}(i, j) = \text{clip}\left(\frac{1}{C} + \left(\rho_i - \frac{1}{C}\right)r(d_j, e_i), \frac{1}{C} + \delta_c, 0.995\right)$$

where $r(d_j, e_i)$ is a retention factor that decreases with task difficulty but preserves a positive margin above chance, and δ_c is a small class-count-dependent safety margin. When an honest worker is incorrect, it chooses uniformly among the remaining $C - 1$ labels. This chance-aware calibration avoids an unrealistic anti-correlated binary-worker regime on binary tasks.

Adversary strategies (implemented regimes). The simulator uses two disjoint attack pools that match the notebook implementation. In both regimes, random chooses a label uniformly at random from $\{0, \dots, C - 1\}$ and always one class always outputs a constant target label. The remaining strategies depend on the attacker-knowledge regime:

Realistic pool: `proxy_flip` replaces the current honest-worker majority proxy for a task with a sampled alternative label, and `collusion_proxy` targets the hardest 30% of tasks and, with probability 0.85, coordinates colluders onto a shared non-proxy label; otherwise it outputs a random label. These attacks do not observe the true task label.

Oracle pool: `oracle_flip` samples a non-true label using the ground-truth label y_j , `oracle_item_dependent` uses y_j on tasks with $d_j > 0.45$ with probability 0.8 and otherwise outputs a random label, and `oracle_collusion` targets the hardest 30% of tasks and, with probability 0.85, coordinates colluders onto a shared non-true label. These attacks explicitly observe y_j and are used only in the oracle stress-test regime.

Accordingly, the realistic pool is proxy-based and deployment-oriented, whereas the oracle pool is label-aware and should be interpreted as an upper bound on attacker strength.

3.8.2 Threat Model

We consider an adversary that can inject a fraction μ of malicious workers into the worker population. In our robustness experiments, μ is swept over $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Adversarial workers are heterogeneous: each adversary is assigned a strategy uniformly from a mixture of attack behaviors (random, always-one-class, flip/partial-flip, item-dependent, and collusion), so the platform observes a mix of noisy and targeted manipulations rather than a single fixed pattern.

Attacker capabilities. Adversaries can submit labels on the same assignment graph as honest workers. Since each worker labels a random 60–90% subset of tasks, attacks are only partially observed: an adversary does not label every task, and any given task may be influenced by only a subset of adversaries. In the collusion setting, a subset of adversaries can additionally coordinate on a targeted subset of tasks, defined as the hardest 30% tasks according to the simulator’s difficulty variable; this induces correlated, task-targeted label manipulation.

Attacker knowledge. The realistic regime does not expose y_j to the attacker and uses only random, always_one_class, proxy_flip, and collusion_proxy. The oracle regime exposes y_j and replaces the proxy-based attacks with oracle_flip, oracle_item_dependent, and oracle_collusion. We therefore interpret the realistic regime as the primary deployment-oriented threat model and the oracle regime as an upper-bound stress test rather than as the main practical scenario.

Attacker goal. The primary attacker objective is to reduce overall task-label accuracy across the dataset. For collusion, the attacker additionally aims to maximize harm on a targeted subset of tasks (the hardest items), which is relevant for stress-testing robustness on difficult instances.

4.0 Results and Analysis

We evaluate the proposed method under increasing adversarial participation by sweeping the adversarial worker ratio $\mu \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The primary model is GNN_ADAPTIVE_CONTENT; GNN_NO_WORKER GT is reported to isolate the no-worker-label setting, and GNN_WEAK_GOLD is included as an auxiliary weak-supervision reference.

Unless otherwise stated, experiments follow the controlled simulation protocol described in the previous section (datasets, baselines, metrics, seeds/repeats, and threat model). We emphasize realistic attacks in the main text because they better reflect deployment conditions; oracle attacks are retained as upper-bound stress tests.

4.1 Robustness Under Increasing Adversarial Participation

Figure 4 summarizes the realistic-attack robustness curves, and Table 1 reports representative held-out test accuracy at $\mu = 0.0$ and $\mu = 0.5$.

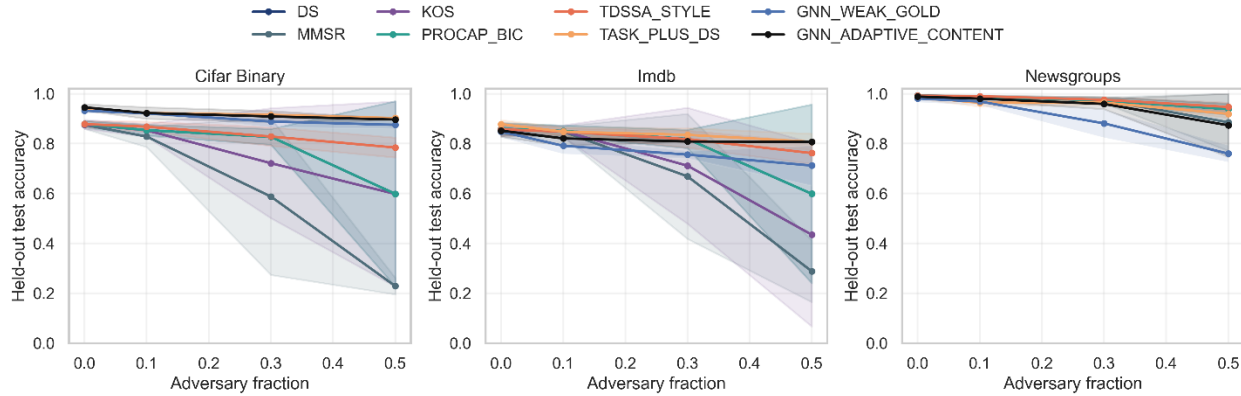


Figure 4: Robustness curves under increasing adversarial ratio μ

The results support a qualified interpretation. On `cifar_binary`, `GNN_ADAPTIVE_CONTENT` and `TASK_PLUS_DS` are essentially tied on mean realistic-attack accuracy (91.84% vs. 91.88%), and both clearly outperform crowd-only D&S (78.95%). On `imdb`, `TASK_PLUS_DS` remains the strongest fair baseline (84.22%), while `GNN_ADAPTIVE_CONTENT` is competitive but lower (82.20%). On `newsgroups`, adaptive content selection materially improves the GNN relative to the fixed-content and no-worker-GT variants, yet classical and weak-gold crowd-centric methods remain strongest on average (TDSSA_STYLE: 97.62%, D&S: 97.29%).

Table 1: Representative held-out test accuracy (mean %) under realistic attacks at $\mu = 0.0$ and $\mu = 0.5$.

Dataset	Model	$\mu = 0.0$	$\mu = 0.5$
cifar_binary	GNN_ADAPTIVE_CONTENT	94.55	89.65
	GNN_NO_WORKER_GT	94.55	89.65
	TASK_PLUS_DS	93.48	90.23
	D&S	87.83	59.85
	TDSSA_STYLE	87.83	78.43
	TASK_ONLY	87.60	87.60
imdb	GNN_ADAPTIVE_CONTENT	85.20	80.67
	GNN_NO_WORKER_GT	83.67	80.67
	TASK_PLUS_DS	87.80	80.73
	D&S	85.93	59.87
	TDSSA_STYLE	85.73	76.20
	TASK_ONLY	76.40	76.40
newsgroups	GNN_ADAPTIVE_CONTENT	98.85	87.35
	GNN_NO_WORKER_GT	94.81	76.72
	TASK_PLUS_DS	98.65	91.94
	D&S	99.20	93.82
	TDSSA_STYLE	99.20	94.70
	TASK_ONLY	71.50	71.50

Paired comparisons reinforce this dataset-dependent picture. Under realistic attacks, GNN_ADAPTIVE_CONTENT significantly outperforms D&S on cifar_binary ($\Delta = 0.129$, $p = 0.026$), is statistically indistinguishable from TASK_PLUS_DS on that dataset ($\Delta \approx 0$, $p = 0.939$), and is significantly below TASK_PLUS_DS on imdb ($\Delta = -0.020$, $p = 0.007$). On newsgroups, the adaptive GNN is below D&S and TDSSA_STYLE on average, but the differences are not significant at the conventional 0.05 level ($p = 0.169$ and $p = 0.120$, respectively). Accordingly, the paper does not claim that the method is uniformly best; instead, it is competitive under realistic attacks and particularly strong when relational evidence and content features are complementary.

4.2 Real Crowdsourcing Benchmark Validation

To address the simulation-only limitation, we evaluate the models on two public answer aggregation benchmarks available through Crowd-Kit (Ustalov, Pavlichenko, and Tseitlin 2024). These datasets provide out-of-simulator evidence for aggregation quality and transfer to real crowd annotations, but they do not reproduce the simulated attack generator. Figure 5 summarizes the accuracy comparison, and Table 2 reports accuracy and macro-F1.

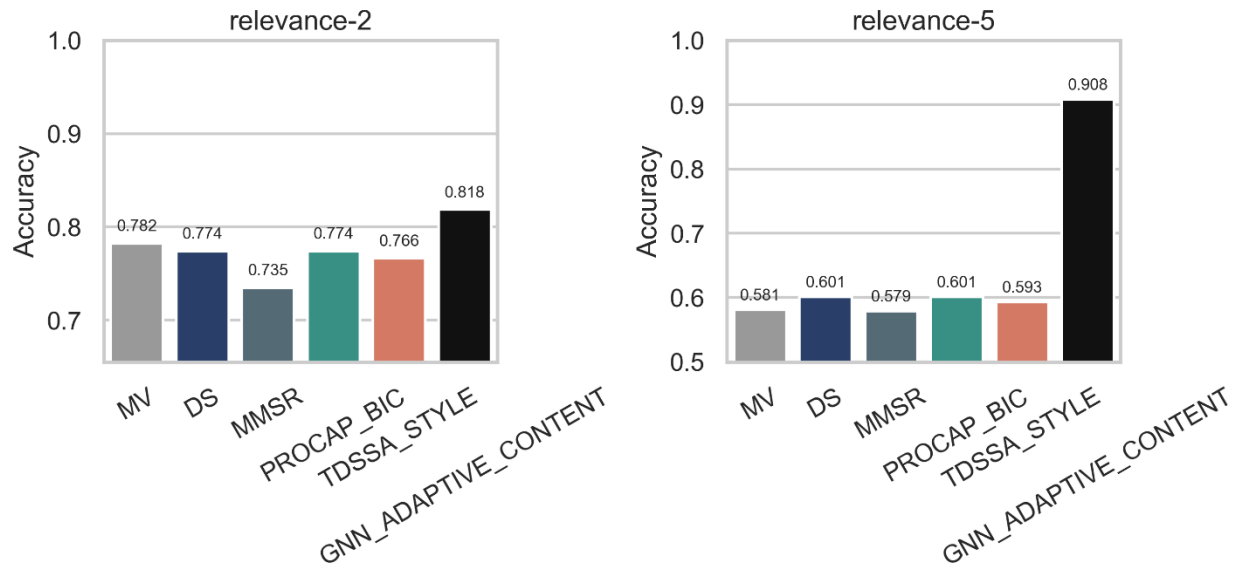


Figure 5: Performance on public real crowdsourcing benchmarks.

On relevance-2, the highest accuracy is achieved by GNN_WEAK_GOLD (82.14%), while the practical no-worker-label variants GNN_ADAPTIVE_CONTENT and GNN_NO_WORKER_GT tie at 81.85% with 80.98 macro-F1. We emphasize the latter pair as the main practical result because they do not require worker-maliciousness labels. On relevance-5, GNN_ADAPTIVE_CONTENT and GNN_NO_WORKER_GT again coincide, both reaching 90.80% accuracy and 72.90 macro-F1, substantially above D&S and PROCAP_BIC (60.08%) and above TDSSA_STYLE (59.28%). Paired bootstrap tests on both datasets show significant improvements of GNN_ADAPTIVE_CONTENT over all listed non-neural baselines ($p = 0.001$ across comparisons in the external significance analysis). The identical PROCAP_BIC and D&S scores on both datasets arise because the pairwise collusion screen does not retain a nontrivial worker grouping there, so the grouped inference stage reduces to the same singleton-worker aggregation structure as D&S.

Table 2: Performance on public real crowdsourcing benchmarks (held-out test; mean %).

Dataset	Model	Accuracy	Macro-F1
relevance-2	GNN_ADAPTIVE_CONTENT	81.85	80.98
	GNN_NO_WORKER_GT	81.85	80.98
	GNN_WEAK_GOLD	82.14	81.47
	MV	78.12	78.12
	D&S	77.38	77.07
	MMSR	73.46	48.99
	PROCAP_BIC	77.38	77.07
	TDSSA_STYLE	76.64	76.32
relevance-5	KOS	58.83	58.41
	GNN_ADAPTIVE_CONTENT	90.80	72.90
	GNN_NO_WORKER_GT	90.80	72.90
	GNN_WEAK_GOLD	90.50	72.66
	MV	58.08	55.93
	D&S	60.08	57.59
	MMSR	57.90	55.20
	PROCAP_BIC	60.08	57.59
TDSSA_STYLE	59.28	56.99	

4.3 Ablation Study

Figure 6 reports an ablation analysis across datasets, illustrating how content, edge gating, and worker-attention components influence performance.

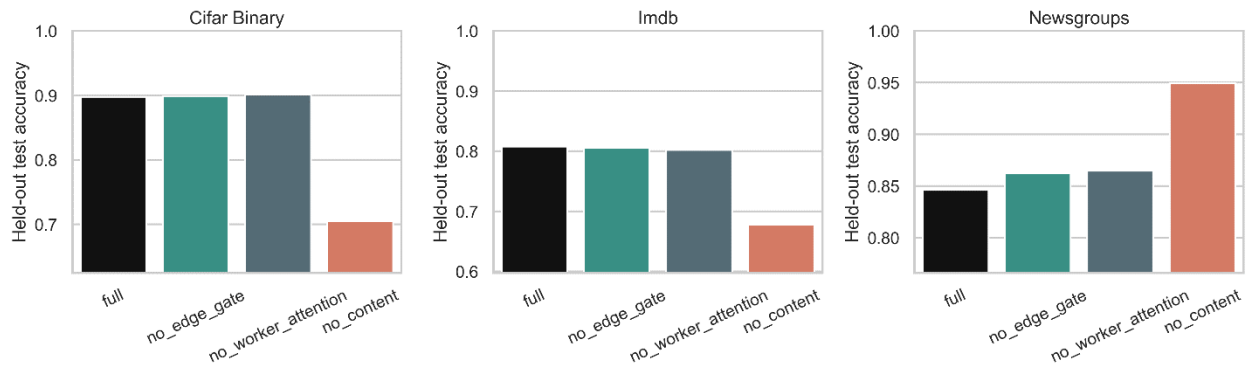


Figure 6: Ablation results across datasets.

The ablation study clarifies that task content is not uniformly beneficial. Removing content reduces accuracy from 89.8% to 70.5% on `cifar_binary` and from 80.8% to 67.8% on `imdb`, confirming that semantic anchoring is valuable on those datasets. In contrast, `newsgroups` improve from 84.6% to 94.9% when content is removed, which explains why the adaptive selector chooses the no-content variant in almost all realistic `newsgroups` runs. Edge gating and worker attention provide smaller but still meaningful refinements relative to this dominant content effect.

5.0 Discussion

This section interprets the empirical findings and clarifies where the proposed approach helps most. The overall picture is mixed but coherent: graph-based aggregation is effective when relational structure and task semantics complement each other, but the strongest baseline can still be dataset-specific.

5.1 Relational modeling under coordinated attacks

Majority-style aggregation and classical probabilistic models can fail when a coordinated group generates a consistent but incorrect consensus. The label-aware bipartite GNN addresses this by combining local labels, worker-task neighborhood structure, and worker-worker coordination cues. This relational view is most beneficial in the settings where adversarial pressure substantially degrades crowd-only aggregation, such as the binary `cifar_binary` simulation and the two real crowdsourcing benchmarks. The real-benchmark results are especially important here because they show transfer beyond the simulator, even though they should be interpreted as real-data aggregation validation rather than as a direct replay of the simulated attack process.

5.2 Why task-content anchoring matters

The ablation results show that task content should be treated as a controllable signal rather than a universally beneficial input. On `cifar_binary` and `imdb`, content acts as a semantic anchor and improves robustness. On `newsgroups`, however, the content branch hurts performance because the crowd signal is already strong and coherent enough that the additional text features introduce more noise than benefit. This is precisely why the adaptive-content variant is useful: it allows the model to keep content when it helps and to discard it when the validation split indicates that the no-content graph is stronger.

5.3 Worker trust scores and practical use

The worker-head outputs should be interpreted as auxiliary risk scores rather than as definitive malicious-worker labels. In the primary $\lambda = 0$ setting, these values are not directly supervised and are therefore not calibrated maliciousness probabilities. Practically, they may still be useful for exploratory soft interventions such as down-weighting suspicious labels, routing additional verification to low-trust workers, assigning hard tasks to high-trust workers, or triggering targeted audits via gold tasks. This auxiliary role is more defensible than presenting worker detection as the main validated outcome of the method.

5.4 Dataset dependence and statistical evidence

Improvements are not uniform across datasets, which is why the manuscript avoids universal claims. On `cifar_binary`, the adaptive GNN is significantly better than D&S under realistic attacks and essentially tied with the strongest fair content-aware baseline. On `imdb`, the adaptive GNN remains competitive, but `TASK_PLUS_DS` is significantly better. On `newsgroups`, the adaptive GNN closes much of the gap created by a fixed-content design, yet D&S and `TDSSA_STYLE` remain stronger on average. The real-benchmark results partially offset this limitation by showing clear gains on `relevance-2` and `relevance-5`. Together, these findings support a narrower claim: the method is robust and adaptable, but not uniformly dominant across all crowdsourcing regimes.

5.5 Limitations and threats to validity

This work still has important limitations. First, most stress testing remains simulator-based, even though we include two real crowdsourcing benchmarks. Second, the simulator uses simplified worker confusion behavior, persistent worker identities, and a relatively dense labeling graph; these assumptions are useful for controlled evaluation but are more favorable than sparse, temporally evolving, or high-churn real platforms. Third, the public benchmarks strengthen transfer and real-aggregation validation, but they do not reproduce the simulated attack generator and therefore should not be read as direct real-world

adversarial benchmarks. Fourth, some directly relevant adversarial baselines are implemented as adaptations rather than official released code. Finally, the primary models do not use worker-maliciousness labels during training, but the worker-trust head still lacks direct validation against real malicious-worker annotations. In addition, the primary models use task ground-truth labels on the training split; accordingly, we do not present them as fully unsupervised truth-inference algorithms. Instead, we frame them as supervised robust aggregation models for settings where historical labeled tasks are available during training, with separate weak-gold experiments reported for more limited supervision. These limitations justify cautious conclusions and motivate further real-data validation.

5.6 Future directions

Future work should validate the framework on additional real-world crowdsourcing datasets, especially benchmarks with known adversarial or temporal dynamics. It would also be valuable to explore stronger text and image encoders under the same held-out protocol, semi-/self-supervised trust objectives, and online task-allocation policies that use trust scores while accounting for robustness, cost, and fairness.

6.0 Conclusion

This paper studied truth inference in crowdsourcing under adversarial behavior and presented a GNN-based framework that models worker–task interactions as a bipartite graph while adaptively using task content features. The evaluation shows that the main method can be trained without worker-maliciousness labels, remains competitive under realistic simulated attacks, and performs strongly on two public real crowdsourcing benchmarks.

The results support three conclusions. First, graph-based relational modeling is useful for robust aggregation, but its benefit is dataset-dependent rather than universal. Second, task content should be treated adaptively: it is valuable on `cifar_binary` and `imdb`, but harmful on `newsgroups`. Third, the strongest empirical support for the framework comes from real answer-aggregation benchmarks, where it clearly outperforms the classical and robust baselines considered in this study, although those benchmarks provide more direct transfer validation than adversarial attack replay.

Overall, the findings indicate that adaptive integration of task semantics with graph-based relational inference yields a practical and robust truth-inference pipeline, if claims are scoped carefully: the gains are dataset-dependent and strongest when relational evidence and task semantics complement each other, while classical aggregation can still remain competitive or superior on some coherent multiclass crowdsourcing regimes.

Conflict of Interest

The authors declare that they have no competing interests.

References

- [1] Y. Jin et al. “A technical survey on statistical modelling and design methods for crowdsourcing quality control”. In: *Artificial Intelligence* 287 (2020), p. 103351.
- [2] Y. Zheng et al. “Truth inference in crowdsourcing: Is the problem solved?” In: *Proceedings of the VLDB Endowment* 10.5 (2017), pp. 541–552.
- [3] U. Gadiraju et al. “Understanding malicious behavior in crowdsourcing platforms: The case of online surveys”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI)*. 2015, pp. 1631–1640.
- [4] A. Checco, J. Bates, and G. Demartini. “Adversarial attacks on crowdsourcing quality control”. In: *Journal of Artificial Intelligence Research* 67 (2020), pp. 375–408.

- [5] P. P. Chen, H. L. Sun, and Z. Chen. "Data poisoning attacks on crowdsourcing learning". In: APWeb/WAIM Joint International Conference on Web and Big Data. Springer, 2021, pp. 164–179.
- [6] F. Tahmasebian et al. "Crowdsourcing under data poisoning attacks: A comparative study". In: IFIP Annual Conference on Data and Applications Security and Privacy. Springer, 2020, pp. 310–332.
- [7] C. Miao et al. "Attack under disguise: An intelligent data poisoning attack mechanism in crowdsourcing". In: Proceedings of the 2018 World Wide Web Conference (WWW). 2018, pp. 13–22.
- [8] P. P. Chen et al. "Collusion-proof result inference in crowdsourcing". In: Journal of Computer Science and Technology 33 (2018), pp. 351–365.
- [9] Y. Wang, K. Wang, and C. Miao. "Truth discovery against strategic sybil attack in crowdsourcing". In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD). 2020, pp. 95–104.
- [10] D. Tao et al. "Domain-weighted majority voting for crowdsourcing". In: IEEE Transactions on Neural Networks and Learning Systems 30.1 (2019), pp. 163–174. doi: 10.1109/TNNLS.2018.2836969.
- [11] M. Karaköse, H. Yetiş, and M. Çeçen. "A New Approach for Effective Medical Deepfake Detection in Medical Images". In: IEEE Access 12 (2024), pp. 52205–52214. doi: 10.1109/ACCESS.2024.3386644.
- [12] T. N. Kipf and M. Welling. "Semi-supervised classification with graph convolutional networks". In: International Conference on Learning Representations (ICLR). arXiv:1609.02907. 2017.
- [13] J. Gilmer et al. "Neural message passing for quantum chemistry". In: Proceedings of the 34th International Conference on Machine Learning (ICML). 2017, pp. 1263–1272.
- [14] Z. Wu et al. "A comprehensive survey on graph neural networks". In: IEEE Transactions on Neural Networks and Learning Systems 32.1 (2021), pp. 4–24.
- [15] J. Zhou et al. "Graph neural networks: A review of methods and applications". In: AI Open 1 (2020), pp. 57–81.
- [16] K. He et al. "Deep residual learning for image recognition". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, pp. 770–778.
- [17] A. P. Dawid and A. M. Skene. "Maximum likelihood estimation of observer error-rates using the EM algorithm". In: Journal of the Royal Statistical Society: Series C (Applied Statistics) 28.1 (1979), pp. 20–28.
- [18] J. Whitehill et al. "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise". In: Advances in Neural Information Processing Systems (NeurIPS). 2009.
- [19] D. Karger, S. Oh, and D. Shah. "Iterative learning for reliable crowdsourcing systems". In: Advances in Neural Information Processing Systems (NeurIPS). 2011.
- [20] Qianqian Ma and Alex Olshevsky. "Adversarial Crowdsourcing Through Robust RankOne Matrix Completion". In: Advances in Neural Information Processing Systems (NeurIPS). 2020. arXiv: 2010.12181 [cs.LG]. url: <https://arxiv.org/abs/2010.12181>.
- [21] C. Song, K. Liu, and X. Zhang. "Collusion Detection and Ground Truth Inference in Crowdsourcing for Labeling Tasks". In: Journal of Machine Learning Research 22.190 (2021), pp. 1–45.
- [22] A. R. Kurup and G. P. Sajeev. "Classifying workers for mitigating adversarial attacks in crowdsourcing". In: IEEE Access 13 (2025), pp. 142713–142727. doi: 10.1109/ACCESS.2025.3598463.
- [23] H. Kim et al. "Graph Anomaly Detection with Graph Neural Networks: Current Status and Challenges". In: IEEE Access 10 (2022). also available as arXiv:2209.14930, pp. 111820–111829. doi: 10.1109/ACCESS.2022.3211306.
- [24] T. Luo et al. "Graph Neural Networks for Trust Evaluation: Criteria, State-of-the-Art, and Future Directions". In: IEEE Network 39.4 (2025), pp. 37–46. doi: 10.1109/MNET.2025.3551068.
- [25] S. Heeb, A. Plesner, and R. Wattenhofer. "Sybil detection using graph neural networks". In: arXiv preprint arXiv:2409.08631 (2024). arXiv: 2409.08631.
- [26] Hanlu Wu et al. "Exploiting Heterogeneous Graph Neural Networks with Latent Worker/Task Correlation Information for Label Aggregation in Crowdsourcing". In: ACM Transactions on Knowledge Discovery from Data 16.2 (2022), 27:1–27:18. doi: 10.1145/3460865.

- [27] Zijian Ying et al. "A Little Truth Injection But a Big Reward: Label Aggregation With Graph Neural Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.5 (2024), pp. 3169–3182. doi: 10.1109/TPAMI.2023.3338216.
- [28] Dmitry Ustalov, Nikita Pavlichenko, and Boris Tseitlin. "Learning from Crowds with Crowd-Kit". In: *Journal of Open Source Software* 9.96 (2024), p. 6227. doi: 10.21105/joss.06227.
- [29] Z. Tan and M. Karaköse. "On-Policy Deep Reinforcement Learning Approach to Multi Agent Problems". In: *Interdisciplinary Research in Technology and Management*. CRC Press, 2021, pp. 369–376. doi: 10.1201/9781003202240-58.
- [30] N. F. Bar and M. Karaköse. "Collaborative approach for swarm robot systems based on distributed DRL". In: *Engineering Science and Technology, an International Journal* 53 (2024), p. 101701. doi: 10.1016/j.jestch.2024.101701.
- [31] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Tech. rep. Computer Science Department, University of Toronto, 2009.
- [32] A. L. Maas et al. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the ACL-HLT*. 2011, pp. 142–150.
- [33] K. Lang. "NewsWeeder: Learning to Filter Netnews". In: *Proceedings of the International Conference on Machine Learning (ICML)*. 1995, pp. 331–339.