DeCo: Defect-Aware Modeling with Contrasting Matching for Optimizing Task Assignment in Online IC Testing

Lo Pang-Yun Ting, **Yu-Hao Chiang**, **Yi-Tung Tsai**, **Hsu-Chao Lai** and **Kun-Ta Chuang**
Dept. of Computer Science and Information Engineering, National Cheng Kung University, Taiwan
{lpyting, yhchiang, yttsai, hclai}@netdb.csie.ncku.edu.tw,
ktchuang@mail.ncku.edu.tw

Abstract

In the semiconductor industry, integrated circuit (IC) processes play a vital role, as the rising complexity and market expectations necessitate improvements in yield. Identifying IC defects and assigning IC testing tasks to the right engineers improves efficiency and reduces losses. While current studies emphasize fault localization or defect classification, they overlook the integration of defect characteristics, historical failures, and the insights from engineer expertise, which restrains their effectiveness in improving IC handling. To leverage AI for these challenges, we propose ***DeCo***, an innovative approach for optimizing task assignment in IC testing. *DeCo* constructs a novel defect-aware graph from IC testing reports, capturing co-failure relationships to enhance defect differentiation, even with scarce defect data. Additionally, it formulates defect-aware representations for engineers and tasks, reinforced by local and global structure modeling on the defect-aware graph. Finally, a contrasting-based assignment mechanism pairs testing tasks with QA engineers by considering their skill level and current workload, thus promoting an equitable and efficient job dispatch. Experiments on a real-world dataset demonstrate that *DeCo* achieves the highest task-handling success rates in different scenarios, exceeding 80%, while also maintaining balanced workloads on both scarce or expanded defect data. Moreover, case studies reveal that *DeCo* can assign tasks to potentially capable engineers, even for their unfamiliar defects, highlighting its potential as an AI-driven solution for the real-world IC failure analysis and task handling.

1 Introduction

As the demand for integrated circuits (ICs) continues to grow across numerous specialized sectors, it is essential in the semiconductor industry to minimize the time from production to shipment while maintaining quality at every stage. When manufacturers produce and deliver samples to customers, they receive numerous return material authorizations (RMAs) [K *et al.*, 2023]. RMAs play a crucial role in the development of the next IC phase, as customer-reported issues are analyzed by the quality assurance (QA) and test verification teams. Addressing these issues directly affects IC design improvements and production efficiency, making RMA handling a key competitive factor.

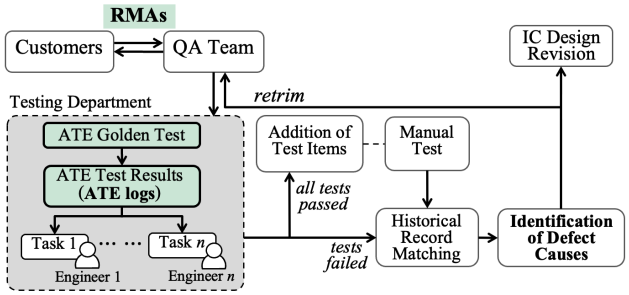

Figure 1: The RMA processing flow.

The RMA handling process is shown in Figure 1. In the testing department, RMA first undergoes golden version testing through the automatic test equipment (ATE) [Lee *et al.*, 2003], where each RMA is regarded as a unique task assigned to an engineer. The ATE generates a test report (ATE logs) containing results for various test items. Engineers analyze these reports to identify IC defect types, using historical RMA records and experience to diagnose the root cause. Once determined, they report findings to the QA team, aiding IC design improvements.

Current studies concentrate on automating fault localization within IC testing [Fan *et al.*, 2020; Fan *et al.*, 2023], with the goal of pinpointing potential fault locations or defect types during manufacturing, or improving the detection yield during the design phase to reduce both cost and time [He and Wang, 2007; Lang *et al.*, 2022]. However, focusing solely on fault localization may lead to critical limitations. ❶ Specifically, certain defect types exhibit similar failure characteristics, and when defect data are scarce, identifying them becomes inherently challenging. As such, relying solely on automated fault localization is inadequate, as it does not utilize contextual insights from previous cases. Engineers with prior experience in similar failure scenarios are still often required to ensure accurate defect diagnosis. ❷ In additions, exist-

File Category	Variable	Description	Example
RMA information	$rma.id$	The individual numeric identifier for the received RMA from customers.	43219Q
	$rma.start$	The date (year, month, and day) on which an RMA is received.	2020-11-25
	$rma.close$	The date (year, month, and day) on which an RMA was completed.	2020-12-20
	$rma.eng$	The name of the engineer responsible for handling the RMA.	Teila
	$defect.type$	The primary defect type causing the IC failure in an RMA.	#EIPD
ATE logs of an RMA	$test.id$	The individual identifier for the test item.	1022
	$test.name$	The name for the test item.	Ripple_L04_-VCC
	$measure$	The measured value of the test item.	0.225 dB
	$test.result$	The test results of the test item.	passed

Table 1: Overview of the data structure of an RMA and its ATE logs.

ing approaches overlook the structural relationships between defects, historical failure logs, and engineers' workload and expertise, which hinder effective RMA handling by leading to suboptimal task assignments and inefficient defect resolution. These limitations reveal a research gap: existing work focuses on defect detection, but lacks research on optimizing RMA handling to reduce manual diagnosis time and enhance task allocation efficiency as shown in Figure 1.

Addressing this gap is essential for reducing resolution time and enhancing overall IC testing workflows. To address the identified challenges and enhance RMA handling, we propose **DeCo** (**De**fect-Aware Modeling with **Co**ntrasting Matching), a novel framework designed to optimize RMA task assignment by maximizing task-handling success rates, while ensuring engineer workloads remain manageable. Specifically, to overcome the first limitation, we design a **defect-aware graph** based on ATE logs of each RMA task, structuring co-failure relationships among RMAs. This graph structure encodes hidden dependencies among defects, enabling more accurate defect differentiation even when ATE logs for defect types are scarce. Additionally, we formulate **defect-aware representations** for engineers and RMA tasks based on their relationships with defect types. To further reinforce representations, we design local and global structure modeling on the defect-aware graph, effectively capturing failure characteristics. These two modeling approaches can be deployed independently or integrated to provide a more comprehensive understanding of defect characteristics, directly addressing the second limitation. Finally, a **contrasting-based assignment** is introduced to match new RMA tasks with the most suitable engineers based on their representations and engineers' current workloads, ensuring optimal task assignment and improving overall efficiency in RMA handling.

Our key contributions are summarized as follows:

- We introduce *DeCo*, a novel framework for optimizing RMA task assignment, maximizing task-handling success rates while balancing engineer workloads.
- To capture the failure characteristics of RMA tasks, we construct a defect-aware graph and design both local and global structure modeling to improve defect-aware task representations, using a contrastive-based mechanism to refine task-to-engineer assignments.
- We evaluate *DeCo* on a real-world dataset under both scarce and expanded ATE log conditions. Results demonstrate that *DeCo* achieves the highest success rate while keeping workloads manageable across different simulation scenarios, highlighting its potential as an AI-driven solution for intelligent IC failure analysis and RMA handling.

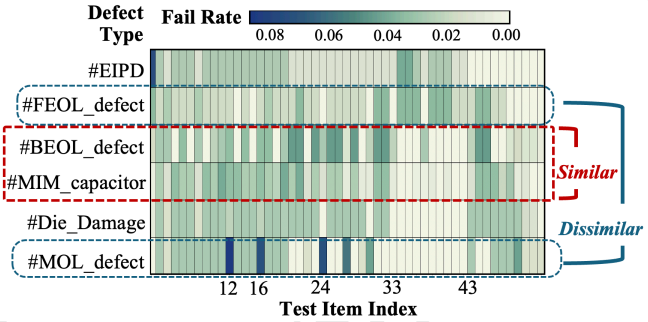

Figure 2: Fail rates of each defect type of IC modules on different test items in ATE logs.

2 Preliminaries

2.1 Data Analysis

Real-World Dataset

For this study, we use real data from a semiconductor company, including RMAs for IC failures and corresponding ATE logs from 2020 to 2023, handled by four engineers. Table 1 lists key data fields, with certain fields (e.g., $rma.id$, $rma.eng$, $test.id$) anonymized. Engineers categorize RMAs into $defect.type$ after analyzing ATE log results. Each ATE log contains multiple test items, and an RMA fails a test item if its measured value ($measure$) exceeds the normal range.

Analysis on Failure Characteristics

We perform a preliminary data analysis to validate that different defect types may share similar failure characteristics, as mentioned in Sec. 1. Figure 2 shows the fail rate of each test name across defect types in all ATE logs. While fail rates vary, some defect types exhibit similar failure characteristics, whereas others differ significantly.

Similar Characteristics. Defect types "*#BEOL_defect*" and "*#MIM_capacitor*" (highlighted by the red dotted line in Figure 2) exhibit nearly zero fail rates between test indices 33 to 43 and share similar fail rate distributions in other regions,

particularly in indices 24 to 33. "*#MIM_capacitor*" refers to issues in metal-insulator-metal capacitors, such as capacitance deviations or dielectric defects, while "*#BEOL_defect*" involves back-end-of-line manufacturing problems, including metal interconnect failures, via defects, or material delamination. The similarities between these two defect types arise from their shared origins in manufacturing challenges, such as material impurities, process misalignments, or contamination, which can lead to comparable failure characteristics.

Dissimilar Characteristics. Some defect types exhibit distinct failure characteristics. Specifically, "*#FEOL_defect*" and "*#MOL_defect*" (highlighted by the blue dotted line in Figure 2) show substantial differences in fail rate distributions. "*#FEOL_defect*" has higher fail rates at indices 30 to 43, while "*#MOL_defect*" is significantly higher at indices 12, 16, and 24. This discrepancy arises because the type "*#FEOL_defect*" occurs in the Front-End-Of-Line (FEOL) stage, affecting transistor performance, whereas "*#MOL_defect*" originates in the Middle-Of-Line (MOL) stage, impacting signal transmission and interconnect resistance. As these defects arise at different manufacturing stages, their failure characteristics differ.

This finding shows that defect types vary in their associations with failure characteristics. Capturing these characteristics in ATE logs is crucial for assigning RMA tasks to suitable engineers. Accurate task matching not only enhances successful task handling but also reducing inefficiencies from misassignments, helping minimize the overall workload needed to complete all tasks.

2.2 Problem Definition

Definition 1 (Task): In this study, each RMA initiated by a customer is regarded as an unique task that requires handling. The set of tasks is denoted as $\mathcal{V}_{task}$.

Problem Statement: Given a set of tasks $\mathcal{V}_{task}$ and a set of engineers E, each task $t \in \mathcal{V}_{task}$ is associated with its ATE log testing results and defect type, and each engineer $e \in E$ has records of previously handled RMAs. Note that new RMA tasks arrive sequentially and are deemed successfully accomplished if they are finished prior to their designated close dates (variable $rma.close$ in Table 1). Our goal is to assign new tasks to appropriate engineers to maximize the success rate while ensuring workloads remain manageable[1].

3 Proposed Method: DeCo

The architecture of *DeCo* comprises two main components. First, it learns representations for engineers and RMA tasks based on their relationships with defect types, incorporating both local and global structure modeling to capture failure characteristics from ATE logs (Sec. 3.1). Then, based on the learned representations, a contrastive approach is used to assign engineers to new RMA tasks by identifying the most suitable assignment (Sec. 3.2).

[1]"Success Rate" and "Workload" are defined as the ratio of successfully accomplished tasks and the average number of tasks handled per engineer, respectively. For ease of presentation, details are described in Sec. 4.1.

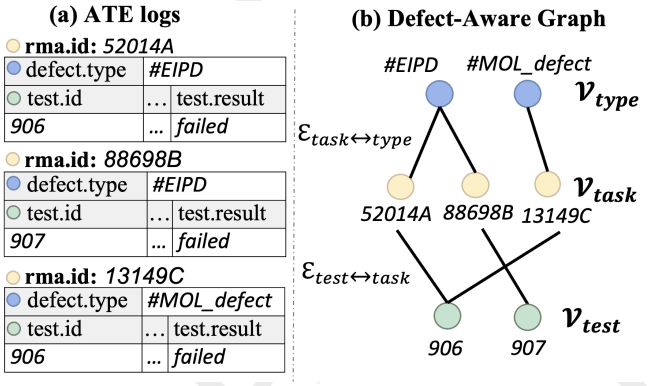

Figure 3: Defect-aware graph construction from ATE logs.

3.1 Defect-Aware Representation

Engineer Representation

To evaluate the ability of each engineer in handling different defect types of ICs, we first formulate the normalized representation $\bar{\mathbf{e}}_i$ for each engineer $e_i \in E$, utilizing the data from RMAs they have previously handled. Given the defect type set $\mathcal{V}_{type}$, let $\mathcal{P}_{i,p}$ and $\mathcal{N}_{i,p}$ denote the average processing time and the count of defect type $p \in \mathcal{V}_{type}$ handled by engineer $e_i \in E$, respectively. An engineer e_i's ability to handle defect type p is defined as follows, where shorter processing time and more handling experience indicate higher ability:

$$\text{ability}(i,p) = \text{avg}\left(1 - \frac{\mathcal{P}_{i,p}}{\sum_{e_j} \mathcal{P}_{j,p}/|E|}\ ,\ \frac{\mathcal{N}_{i,p}}{\sum_{e_j} \mathcal{N}_{j,p}/|E|}\right). \tag{1}$$

Based on the defined ability $a_{i,p}$ of engineer e_i, the normalized defect-aware representation $\bar{\mathbf{e}}_i \in \mathbb{R}^{|\mathcal{V}_{type}|}$ of engineer e_i is formulated as follows:

$$\bar{\mathbf{e}}_i = \left[\frac{\text{ability}(i,p)}{\sum_{p'} \text{ability}(i,p')}\right]_{p\in\mathcal{V}_{type}}. \tag{2}$$

Task Representation

To preserve ATE log information and represent new tasks, we construct a defect-aware graph based on ATE logs and *encode each node within it* to capture failure characteristics.

Definition 2 (Defect-Aware Graph): Given the task set $\mathcal{V}_{task}$, defect type set $\mathcal{V}_{type}$, and the test item set $\mathcal{V}_{test}$, the relationships between these entities are defined as follows. If a test item $m \in \mathcal{V}_{test}$ fails in the ATE logs of task $t \in \mathcal{V}_{task}$, an edge $\varepsilon_{m\leftrightarrow t}$ is formed, defining the edge set $\mathcal{E}_{test\leftrightarrow task} = \{\varepsilon_{m\leftrightarrow t} \mid m \in \mathcal{V}_{test}, t \in \mathcal{V}_{task}\}$. If a task $t \in \mathcal{V}_{task}$ belongs to defect type $p \in \mathcal{V}_{type}$, an edge $\varepsilon_{t\leftrightarrow p}$ is formed, defining the edge set $\mathcal{E}_{task\leftrightarrow type} = \{\varepsilon_{t\leftrightarrow p} \mid t \in \mathcal{V}_{task}, p \in \mathcal{V}_{type}\}$. Therefore, the defect-aware graph $\mathcal{G}$ is formulated as follows:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}) \begin{cases} \mathcal{V} = \mathcal{V}_{test} \cup \mathcal{V}_{task} \cup \mathcal{V}_{type}\ \ (nodes) \\ \mathcal{E} = \mathcal{E}_{test\leftrightarrow task} \cup \mathcal{E}_{task\leftrightarrow type}\ \ (edges) \\ \mathcal{X} = \{x_v | v \in \mathcal{V}\}\ \ (attributes) \end{cases}, \tag{3}$$

where $\mathcal{X}$ is an attribute set, and each $x_v \in \mathcal{X}$ represents a node attribute of "*test item*", "*task*" or "*defect type*". Figure 3 illustrates an example of graph construction.

The observation in Figure 2 highlights that different defect types exhibit varying degrees of correlation in their failure characteristics. To effectively capture these characteristics and encode each node in the defect-aware graph $\mathcal{G}$, we design two types of modeling approaches: *local structure modeling* and *global structure modeling*. These approaches can be applied individually or in combination. Generally, integrating both improves performance by capturing diverse structural information. In some cases, such as when ATE logs are extensive, global structure modeling alone better captures high-level defect relationships since it leverages broader connections. Conversely, local structure modeling alone may yield better results. By flexibly applying local and global structural information, *DeCo* can provide a more comprehensive representation of defect-aware relationships.

Local structure modeling. For local structure modeling, we represent the characteristics of each node in the defect-aware graph $\mathcal{G}$ by exploring its local neighborhood. This allows us to encode fine-grained, proximity-based relationships among test items, tasks, and defect types. To achieve this, we redesign the classic model DeepWalk [Perozzi *et al.*, 2014], which learns node embeddings by generating random walks over the graph and optimizing node co-occurrence within local neighborhoods. Also, we extend the skip-gram architecture [Grover and Leskovec, 2016; Mikolov *et al.*, 2013] by considering the second-order proximity [Tang *et al.*, 2015] of each node in defect-aware graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ to better capture structural proximity. The objective function of local structure modeling is designed as follows:

$$\mathcal{O}_{local} = -\log Z_u + \sum_{v \in N(u)} \Big(1 + \omega_{v,u}\Big)\Big(\psi(v) \cdot \psi(u)\Big), \quad (4)$$

$$\omega_{v,u} = \begin{cases} J\big(N(v), N(u)\big) & \text{, if } v \neq u \text{ and } v, u \in \mathcal{V}_{task} \\ 0 & \text{, otherwise} \end{cases}, \quad (5)$$

where $Z_u = \sum_{u \in \mathcal{V}} \exp(\psi(v) \cdot \psi(u))$, and $N(u)$ refers to th set of nodes in the random walk starting from u. The function $J(N(v), N(u))$ denotes the Jaccard similarity between $N(v) \cap \mathcal{V}_{test}$ and $N(u) \cap \mathcal{V}_{test}$. $\psi(v) \in \mathbb{R}^d$ and $\psi(u) \in \mathbb{R}^d$ are embedding vectors of nodes v and u, respectively.

Global structure modeling. For global structure modeling, unlike direct neighborhood relationships, we construct metapaths to traverse multiple intermediate nodes, uncovering high-level relational dependencies. A metapath is an ordered sequence of node types and edge types defined on the network schema. These structures are specifically designed to reveal high-level relational dependencies, which we define as:

Definition 3 (Metapath): We define a metapath P as a path in the form of $u(n_0) \xrightarrow{e_1} n_1 \xrightarrow{e_2} ... \xrightarrow{e_l} v(n_l)$, where $e_j \in \mathcal{E}$, and u and v belong to the same node set, i.e., both are from test items ($\mathcal{V}_{test}$), tasks ($\mathcal{V}_{task}$), or defect types ($\mathcal{V}_{type}$). The intermediate nodes $\{n_1, ..., n_{l-1}\}$ belong to the remaining node sets. For example, if $u, v \in \mathcal{V}_{type}$, the intermediate nodes satisfy $\{n_1, \ldots, n_{l-1}\} \subset \mathcal{V}_{task} \cup \mathcal{V}_{test}$. The metapath describes a composite relationship $\pi = e_1 \circ n_1 \circ e_2 \circ ... \circ e_l$ between node u and v, where $\circ$ denote the composition operator that sequentially connects edges and intermediate nodes in $\mathcal{G}$. Therefore, a metapath structure can be represented in a triplet form $P = (u, \pi, v)$.

To strengthen these structured dependencies, we create a triplet-based representation and employ a margin-based objective to grasp the interconnections between the source node u and the target node v within a metapath $P = (u, \pi, v)$. This approach facilitates the learning of representations for various nodes in the defect-aware graph. Let $\mathcal{P}_{u,v}^{\leq l}$ denote all metapaths from u to v with a length of at most $l \in \mathbb{N}$. The objective function of global structure modeling is defined as:

$$\mathcal{O}_{global} = -\frac{1}{|\mathcal{P}_{u,v}^{\leq l}|} \sum_{P=(u,\pi,v) \in \mathcal{P}_{u,v}^{\leq l}} L_P, \quad (6)$$

$$L_P = \sum_{P' \in \{(u',\pi,v') | u' \neq u \text{ or } v' \neq v\}} \Big[E_P + \xi - E_{P'}\Big]_+, \quad (7)$$

where P' is the set of negative samples generated based on metapath P. E_P is a scoring function that evaluates the plausibility of a metapath P, derived from knowledge graph embedding to capture metapath (triplet structure) relationships. We employ the HolE model [Nickel *et al.*, 2016] as the scoring function for its effectiveness in capturing complex relational triplet structures. Here, $[\mu]_+$ denotes $max(0, \mu)$, and $\xi > 0$ is a hyperparameter of margin.

The objective function for integrating both modeling approaches can be easily derived as:

$$\begin{aligned} &\max_{\psi}(\mathcal{O}_{local}) + \lambda(O_{global}) \\ &= \max_{\psi} \sum_{u \in \mathcal{V}} \Bigg\{ \underbrace{-\log Z_u + \sum_{v \in N(u)} \Big[\Big(1 + \omega_{v,u}\Big)\Big(\psi(v) \cdot \psi(u)\Big)}_{\text{local structure}} \\ &\qquad \underbrace{- \lambda \cdot \frac{1}{|\mathcal{P}_{u,v}^{\leq l}|} \sum_{P \in \mathcal{P}_{u,v}^{\leq l}} \sum_{P'} \Big[E_P + \xi - E_{P'}\Big]_+ \Big]}_{\text{global structure}} \Bigg\}, \end{aligned} \quad (8)$$

where $\lambda > 0$ is a hyperparameter balancing both modeling. Finally, the representation of each test item, task and defect type in the defect-aware graph can be obtained.

Based on Eq. 8, we can formulate the representation of a new task t_k ($\notin \mathcal{V}_{task}$) by considering the failed test items in its ATE logs. Let $\mathcal{F}_k \subset \mathcal{V}_{test}$ includes the failed test items in the ATE logs of new task t_k. The probability of task t_k belongs to defect type $p \in \mathcal{V}_{type}$ is estimated as follows:

$$\text{prob}(k, p) = sim\Big(\frac{1}{|\mathcal{F}_k|} \sum_{f \in \mathcal{F}_k} \psi(f),\ \psi(p)\Big), \quad (9)$$

where $\psi(\cdot)$ represents the embedding vector of a node. The function $sim(\cdot, \cdot)$ represents the cosine similarity. Subsequently, the normalized defect-aware representation $\bar{\mathbf{t}}_k \in \mathbb{R}^{|\mathcal{V}_{type}|}$ of new task t_k is designed as follows:

$$\bar{\mathbf{t}}_k = \left[\frac{\text{prob}(k, p)}{\sum_{p'} \text{prob}(k, p')}\right]_{p \in \mathcal{V}_{type}}. \quad (10)$$

3.2 Contrasting-Based Assignment

After obtaining the defect-aware representation $\bar{\mathbf{e}}_i$ (Eq. 2) and $\bar{\mathbf{t}}_k$ (Eq. 10), we aim to assign the most appropriate engineer to handle the new task t_k. Here, we design the **assignment score** $\mu_{i,k}$ to quantify the suitability of each engineer

$e_i \in E$ for new task t_k. Inspired by contrastive decoding approaches [Li *et al.*, 2022; Chuang *et al.*, 2023], the assignment score $\mu_{i,k}$ is computed by contrasting information across representations $\bar{\mathbf{t}}_k$ and $\bar{\mathbf{e}}_i$ to refine the task-to-engineer assignment process, as designed as follows:

$$\mu_{i,k} = \sum_{p \in \mathcal{V}_{type}} \mathcal{S}(e_i, t_k, p), \tag{11}$$

$$\mathcal{S}(e_i, t_k, p) = \begin{cases} \log \frac{\bar{\mathbf{e}}_i(p) + \epsilon}{\bar{\mathbf{t}}_k(p) + \epsilon} & \text{, if } N_{i,p} \neq 0 \text{ and } \frac{w_i^{curr}}{w_i^{avg}} < 1 \\ -\infty & \text{, otherwise} \end{cases}, \tag{12}$$

where $\epsilon > 0$ is small constant. $\bar{\mathbf{e}}_i(p) = e_{i,p}$ represents the ability of engineer e_i for defect type p (Eq. 1, Eq. 2). $\bar{\mathbf{t}}_k(p) = s_{k,p}$ indicates the probability that new task t_k belongs to defect type p (Eq. 9, Eq. 10). $N_{i,p}$ denotes the number of tasks of defect type p handled by engineer e_i.

Specifically, to prevent excessive workload on engineers, we introduce a workload constraint. The values w_i^{curr} and w_i^{avg} represent the current number of tasks handled by engineer e_i and e_i's average historical workload, respectively. The condition $w_i^{curr}/w_i^{avg} < 1$ ensures that engineers with lighter workloads are prioritized.

Finally, according to the assignment score $\mu_{i,k}$ for each engineer $e_i \in E$ and the new task t_k, we assign task t_k to the engineer $\hat{e}$ with the highest score, ensuring that the selected engineer has sufficient capacity and a manageable workload to handle the new task effectively.

4 Experiments

4.1 Experimental Setup

Dataset and Preprocessing. We use a real-world ATE dataset collected from a semiconductor company, as described in Sec. 2.1. To evaluate methods under both scarce and informative defect data conditions, we consider the original dataset as scarce and create an expanded dataset by increasing the number of engineers, defect types, RMA logs and other relevant factors. The statistics of the scarce (original) dataset and the expanded dataset are presented in Table 2. The first 75% of the total RMA data is used for training, while the remaining 25% is reserved for testing.

Baselines. We compare *DeCo* with the following baselines: **LowestWL**, **MostEXP**, **CF**, as well as our variant **CA**. LowestWL and MostEXP are widely used strategies in industries.

- **LowestWL** assigns each new task to the engineer with the lowest current workload.
- **MostEXP** assigns each new task to the engineer who has handled the most tasks with similar failed test items, considering them the most experienced.

Datasets	# Testers	#Test items	#Defect types	# ATE logs	Avg. handling days
ATE$_{\text{scar}}$	4	1,140	8	234	38.3
ATE$_{\text{exp}}$	26	1,703	14	3,190	41.1

Table 2: Dataset statistics.

- **CF** employs collaborative filtering (CF). An engineer-task matrix from past RMA records calculates suitability scores. Similarity between a new task and past tasks uses Word2Vec embeddings of failed test items, averaging these embeddings to represent each task. An engineer's suitability for a new task is a weighted sum of past task scores, with weights from task cosine similarity.
- **CA** is a variant of *DeCo* retaining only contrast-based assignment (remove defect-aware graphs). The new task's representation uses cosine similarity between fail rates of test items in ATE logs and those of each defect type.

We also evaluate our method with local, global, and combined structure modeling, denoted as ***DeCo***$_{\text{local}}$, ***DeCo***$_{\text{global}}$, and ***DeCo***, respectively.

Evaluation Metrics. Our evaluation relies on two metrics:

- **Success**: The *success rate* (%) is the ratio of tasks successfully completed by engineers, estimated as $|\mathcal{N}_{success}|/|\mathcal{N}_{total}|$, where $\mathcal{N}_{success}$ is the set of tasks completed before their close dates, and $\mathcal{N}_{total}$ includes all new tasks that need to be handled.
- **Workload**: The metric reports the average *daily workload* per engineer, with lower values being better. The daily workload of each engineer is estimated as w_i^t/w_i^{avg}, where w_i^t is the number of tasks handled by engineer e_i on day t, and w_i^{avg} is their average daily task count.

Engineer Processing Time Simulation. To simulate the time required for e_i to handle the new task t_k, we define an influence factor x based on the current workload and experience. It is estimated as: $x = (w_i^{curr}/w_i^{avg}) + (N_{i,p}/N_i^{avg})$, where w_i^{curr} and w_i^{avg} are the current and average number of tasks handled by e_i, respectively. Similarly, $N_{i,k}$ is the number of times e_i has handled tasks of the same defect type as t_k, and N_i^{avg} is the average number of times an engineer handles each defect type. Given the influence factor x, the time $\mathcal{T}_{i,k}$ required for e_i to handle the new task t_k is as:

$$\mathcal{T}_{i,k} = \mathcal{T}_{i,k}^{\text{avg}} + \sigma(\kappa \cdot x - 0.5)\Delta\mathcal{T}_i, \tag{13}$$

where $\mathcal{T}_{i,k}^{avg}$ is the average time e_i takes to handle tasks of the same defect type as t_k, and $\Delta\mathcal{T}_i$ is the difference between the longest and shortest handling times for e_i. The parameter $\kappa > 0$ controls the impact of the influence factor x, with a larger κ means a greater influence of workload and experience on the required time for handling t_k.

Implementation details. For local structure modeling, the walk length and number of walks are set to 20 and 10, respectively. For global structure modeling, the metapath length l is set to 5 (Eq. 6), and the margin ξ is set to 1 (Eq. 7). In Eq. 8, λ is set to 1, the embedding dimension to 128, and the training epoch to 50. All experiments are conducted on a system with 12 CPU cores and 64GB RAM, running CUDA 12.1.

4.2 Results and Analysis

Comparison Results. Table 3 presents the comparative results under different engineer processing time settings, with $\kappa = 1, 2, 5$ (Eq. 13). Key observations include:

Effectiveness. Our framework outperforms all baselines in success rate, with the second-highest success rate also

Dataset $\mathbf{ATE}_{\text{scar}}$								
Metric	κ	LowestWL	MostEXP	CF	CA	***Ours***		
						DeCo$_{\text{local}}$	***DeCo***$_{\text{global}}$	***DeCo***
Success(%) (↑)	1	69.33±7.60	48.66±11.69	64.00±2.79	76.67±1.82	82.66±4.34	82.00±4.47	**86.00 ± 2.79**
Workload (↓)		1.38±0.10	2.21±0.51	1.66±0.11	1.60±0.05	1.66±0.06	1.71±0.08	1.69±0.08
Success(%) (↑)	2	68.00±5.58	48.66±14.45	64.66±2.98	74.49±2.61	**84.66 ± 1.82**	80.00±4.71	83.99±3.65
Workload (↓)		1.45±0.17	2.19±0.53	1.63±0.03	1.54±0.15	1.62±0.20	1.79±0.12	1.68±0.10
Success(%) (↑)	5	74.66±5.05	48.66±14.45	61.33±2.97	73.33±1.01	88.00±1.62	82.66±4.34	**88.01 ± 1.82**
Workload (↓)		1.36±0.18	2.18±0.53	1.63±0.07	1.42±0.07	1.50±0.14	1.62± 0.16	1.50±0.14
Dataset $\mathbf{ATE}_{\text{exp}}$								
Metric	κ	LowestWL	MostEXP	CF	CA	***Ours***		
						DeCo$_{\text{local}}$	***DeCo***$_{\text{global}}$	***DeCo***
Success(%) (↑)	1	76.09±1.94	13.74±11.06	48.35±2.12	78.37±2.24	78.63±2.28	78.06±1.69	**79.32 ± 1.19**
Workload (↓)		0.65±0.01	1.42±0.34	0.70±0.01	0.55±0.01	0.70±0.06	0.74±0.06	0.68±0.04
Success(%) (↑)	2	84.81±1.36	13.24±11.10	60.42±6.74	88.27±2.12	**90.08 ± 1.17**	89.18±1.77	89.52±2.13
Workload (↓)		0.58±0.01	1.23±0.63	0.67±0.04	0.48±0.01	0.50±0.04	0.49±0.04	0.52±0.07
Success(%) (↑)	5	89.38±0.58	13.21±11.25	70.36±3.59	95.07±1.12	95.13±1.31	**95.83 ± 0.71**	95.34±0.83
Workload (↓)		0.33±0.01	1.44±0.03	0.62±0.03	0.40±0.34	0.41±0.02	0.39±0.01	0.41±0.01

Table 3: Performance comparison on datasets $\mathbf{ATE}_{\text{scar}}$ and $\mathbf{ATE}_{\text{exp}}$. The results are reported as the mean ± standard deviation over 10 runs. The best success rate (abbreviated as "Success") results are highlighted in **bold**. Cells in gray indicate workload ratios (abbreviated as "Workload") that are better than the average performance across methods.

achieved by *DeCo* variants (***DeCo***$_{\text{local}}$, ***DeCo***$_{\text{global}}$, or ***DeCo***). Meanwhile, it maintains a competitive workload, demonstrating the effectiveness of capturing failure characteristics from the defect-aware graph. Although LowestWL and CA perform similarly to our method on the $\mathbf{ATE}_{\text{exp}}$ dataset, their success rates remain lower across different κ values, while their workload is only slightly lower.

Generalization. *DeCo* achieves the highest success rate while keeping workloads manageable on both datasets. In $\mathbf{ATE}_{\text{scar}}$, it significantly outperforms baselines, showing its applicability with limited data. In $\mathbf{ATE}_{\text{exp}}$, although *DeCo*$_{\text{local}}$, *DeCo*$_{\text{global}}$, and *DeCo* perform similarly, an interesting trend emerges: when more ATE logs are available ($\mathbf{ATE}_{\text{exp}}$) and the engineer processing time is more affected by workload and experience ($\kappa = 5$), global structure modeling alone achieves a higher success rate. This is because it captures broader task relationships and engineer expertise, leading better task assignments when data is abundant and processing time depends more on workload and experience.

Performance Analysis of Local and Global Modeling Integration. To evaluate the impact of integrating local and global structure modeling, we vary the λ value (Eq. 8) to control the degree of integration, as shown in Figure 4. In $\mathbf{ATE}_{\text{scar}}$, setting λ around 10^{-2} and 10^{-1} (where global modeling has less impact than local modeling) achieves the highest success rate and lowest workload. In $\mathbf{ATE}_{\text{exp}}$, the success rate remains stable across different λ values, but setting λ to 10^{1} yields the lowest workload. This suggests that the expanded dataset contains richer information, allowing global modeling to better capture defect relationships.

Performance on Defect Type Detection. We further evaluate *DeCo* for defect type detection using defect-aware representations. In Eq. 9, $\text{prob}(k, p)$ denotes the probability of task t_k belonging to defect type p. We select the top-3 and top-5 defect types with the highest $\text{prob}(k, p)$ values as detection results. Figure 5 presents the results, analyzing the impact of varying walk length and margin in *DeCo*. Results show that performance improves (stronger red) when both walk length and margin are either smallest or largest. Longer walks help when the margin is 10, while a larger margin helps only at maximum walk length. This implies that longer walks capture more informative structures but need a larger margin for effective embedding separation. Balancing these two factors can further enhance performance.

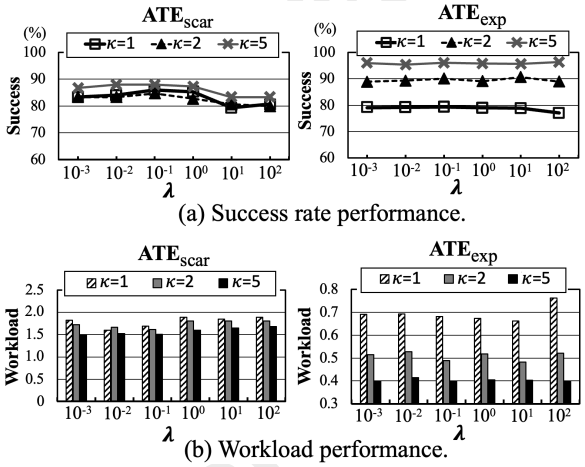

Figure 4: The performance of *DeCo* with different values of λ.

Case study. To examine how *DeCo* assigns different tasks, we present two cases where engineers successfully completed tasks before close dates, as shown in Figure 6.

Case 1. Figure 6a shows a task of defect type "#EIPD". Based

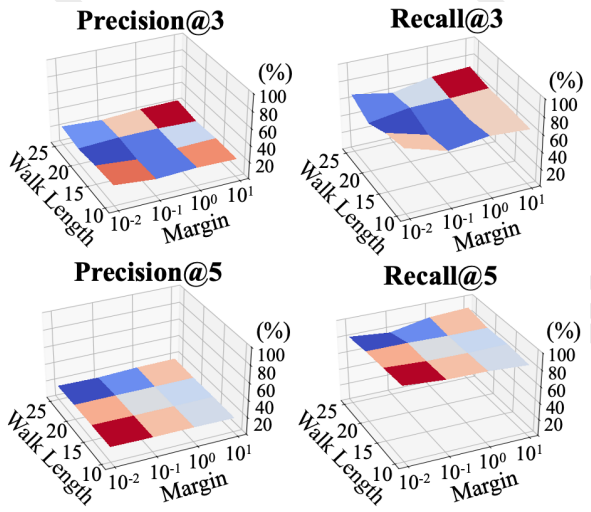

Figure 5: The performance of *DeCo* for detect type detection on dataset $\mathbf{ATE}_{\text{scar}}$ with $\kappa = 1$. Within each metric, higher values are represented in red, while lower values are shown in blue.

on the learned representation $\bar{\mathbf{t}}$ of the task, *DeCo* correctly detects the defect type and assigns the task to Engineer *A*, whose representation $\bar{\mathbf{e}}$ indicates the highest capability for handling "#EIPD". Engineer *A* also has the highest historical handling ratio for this defect type. This case demonstrates that defect-aware representations learned in *DeCo* can effectively capture failure characteristics, enabling accurate defect detection and task assignment to the most experienced engineers.

Case 2. Figure 6b shows a task belonging to type "#BEOL_defect". As observed in Sec. 2.1, "#BEOL_defect" shares similar failure characteristics with "#MIM_capacitor". This similarity causes the defect-aware representation $\bar{\mathbf{t}}$ to predict a higher probability for "#MIM_capacitor" than "#BEOL_defect". However, the contrasting-based assignment in *DeCo* identifies Engineer *D* as a suitable candidate. Even though Engineer *D* has no prior experience with "#BEOL_defect", the engineer's defect-aware representation $\bar{\mathbf{e}}$ is similar to task $\bar{\mathbf{t}}$, indicating potential capability. This suggests that Engineer *D*'s experience with other defect types provides transferable skills for handling this task. As a result, *DeCo* assigns the task to Engineer *D*, who successfully completes it despite having no prior experience with this type.

5 Related Work

Fault Localization. Research on IC testing primarily focuses on optimizing test design in hardware design and manufacturing processes [Lang *et al.*, 2022; Park *et al.*, 2017; Afacan *et al.*, 2021; Lyu *et al.*, 2018]. Machine learning methods have been applied for fault detection and diagnosis during the thin film deposition stage [Fan *et al.*, 2020]. VAE-based generative models have been used to address data imbalance in thin film deposition [Fan *et al.*, 2023]. Power supply noise in ICs has also been detected and analyzed using ATE systems [Liau and Schmitt-Landsiedel, 2003].

Although existing works perform well in various scenarios, certain defects may exhibit similar failure characteristics, making identification challenging. In such cases, accurate diagnosis often relies on experienced engineers. Therefore, our work focuses on identifying IC failure characteristics and assign them to suitable engineers to improve defect handling. This is an area still unexplored in IC test research.

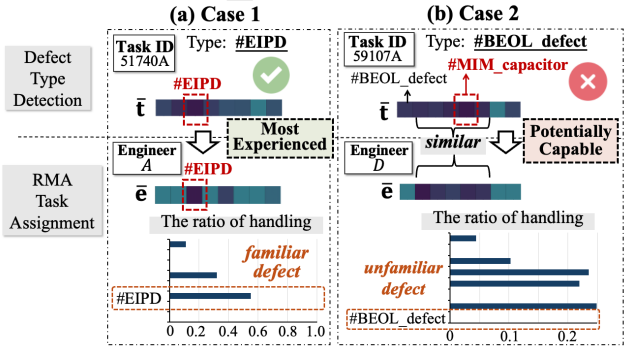

Figure 6: Details of two successfully completed cases, analyzed in two stages: defect type detection and RMA task assignment.

Task Assignment. Recent research focuses on effective assignment while considering constrains [Mo *et al.*, 2013; Hettiachchi *et al.*, 2022; Dickerson *et al.*, 2018; Andersen *et al.*, 2016; Liu *et al.*, 2015; Constantino *et al.*, 2017]. Some research develops the QASCA system to support quality-aware task assignment by incorporating worker skills and quality standards [Zheng *et al.*, 2015]. Various evaluation metrics are also used depending on the task type [Li *et al.*, 2014; Lee *et al.*, 2014; Balakrishnan *et al.*, 2015; De Alfaro and Shavlovsky, 2014]. Some studies have assessed the time required for workers to complete tasks [Mavridis *et al.*, 2016], while others have considered fairness among the population under budget constraints [Goel and Faltings, 2019].

While these works address worker skills, task difficulty, and efficiency, they often overlook worker workloads, a critical factor in practice. Our RMA task assignment addresses this by considering engineer expertise, task failure characteristics, and workload balance while maximizing success rate and ensuring practical applicability.

6 Conclusion

We propose *DeCo*, a framework for optimizing RMA task assignment by constructing a defect-aware graph from ATE logs and employing local and global structure modeling to learn task representations. A contrast-based assignment method refines task-to-engineer allocation. When applied to real-world IC testing data, *DeCo* achieves the highest success rates while keepping engineers' workloads manageable across both scarce and expanded ATE log conditions. We also demonstrate its effectiveness in defect type detection. Two case studies show that *DeCo* can assign potentially capable engineers to previously unfamiliar defects.

Acknowledgments

This paper was supported in part by National Science and Technology Council (NSTC), R.O.C., under Contract 113-2221-E-006-203-MY2, 114-2622-8-006-002-TD1 and 113-2634-F-006-001-MBK.

References

[Afacan *et al.*, 2021] Engin Afacan, Nuno Lourenço, Ricardo Martins, and Günhan Dündar. Machine learning techniques in analog/rf integrated circuit design, synthesis, layout, and test. *Integration*, 77:113–130, 2021.

[Andersen *et al.*, 2016] Per-Arne Andersen, Christian Kråkevik, Morten Goodwin, and Anis Yazidi. Adaptive task assignment in online learning environments. In *Proceedings of the 6th international conference on web intelligence, mining and semantics*, pages 1–10, 2016.

[Balakrishnan *et al.*, 2015] Ramji Balakrishnan, Haijin Lin, and Shiva Sivaramakrishnan. Task assignment, relative and absolute performance evaluation. AAA, 2015.

[Chuang *et al.*, 2023] Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*, 2023.

[Constantino *et al.*, 2017] Ademir Aparecido Constantino, Candido Ferreira Xavier de Mendonca Neto, Silvio Alexandre de Araujo, Dario Landa-Silva, Rogério Calvi, and Allainclair Flausino dos Santos. Solving a large real-world bus driver scheduling problem with a multi-assignment based heuristic algorithm. 2017.

[De Alfaro and Shavlovsky, 2014] Luca De Alfaro and Michael Shavlovsky. Crowdgrader: A tool for crowdsourcing the evaluation of homework assignments. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 415–420, 2014.

[Dickerson *et al.*, 2018] John P Dickerson, Karthik A Sankararaman, Aravind Srinivasan, and Pan Xu. Assigning tasks to workers based on historical data: Online task assignment with two-sided arrivals. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2018.

[Fan *et al.*, 2020] Shu-Kai S. Fan, Chia-Yu Hsu, Du-Ming Tsai, Fei He, and Chun-Chung Cheng. Data-driven approach for fault detection and diagnostic in semiconductor manufacturing. *IEEE Transactions on Automation Science and Engineering*, 17(4):1925–1936, 2020.

[Fan *et al.*, 2023] Shu-Kai S. Fan, Du-Ming Tsai, and Pei-Chi Yeh. Effective variational-autoencoder-based generative models for highly imbalanced fault detection data in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 36(2):205–214, 2023.

[Goel and Faltings, 2019] Naman Goel and Boi Faltings. Crowdsourcing with fairness, diversity and budget constraints. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 297–304, New York, NY, USA, 2019. Association for Computing Machinery.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proc. of ACM SIGKDD*, 2016.

[He and Wang, 2007] Q. Peter He and Jin Wang. Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, 20(4):345–354, 2007.

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

[K *et al.*, 2023] Sudeendra Kumar K, S. S. Rekha, Akshay Koushik, Ayas Kanta Swain, and K. K. Mahapatra. Lightweight secured split test technique with rma capability to prevent ic counterfeiting. In *2023 IEEE International Symposium on Smart Electronic Systems (iSES)*, pages 191–196, 2023.

[Lang *et al.*, 2022] Christopher I. Lang, Fan-Keng Sun, Bruce Lawler, Jack Dillon, Ash Al Dujaili, John Ruth, Peter Cardillo, Perry Alfred, Alan Bowers, Adrian Mckiernan, and Duane S. Boning. One class process anomaly detection using kernel density estimation methods. *IEEE Transactions on Semiconductor Manufacturing*, 35(3):457–469, 2022.

[Lee *et al.*, 2003] Y.J. Lee, T. Kane, J.-J. Lim, L. Schiano, Y.-B. Kim, F.J. Meyer, F. Lombardi, and S. Max. Analysis and measurement of timing jitter induced by radiated emi noise in automatic test equipment. *IEEE Transactions on Instrumentation and Measurement*, 52(6):1749–1755, 2003.

[Lee *et al.*, 2014] Sooyoung Lee, Sehwa Park, and Seog Park. A quality enhancement of crowdsourcing based on quality evaluation and user-level task assignment framework. In *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, pages 60–65. IEEE, 2014.

[Li *et al.*, 2014] Hongwei Li, Bo Zhao, and Ariel Fuxman. The wisdom of minority: discovering and targeting the right group of workers for crowdsourcing. *Proceedings of the 23rd international conference on World wide web*, 2014.

[Li *et al.*, 2022] Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*, 2022.

[Liau and Schmitt-Landsiedel, 2003] E. Liau and D. Schmitt-Landsiedel. Evolution of automatic semiconductor test equipment: automatic test pattern learning, classification, optimisation and generation for power supply noise. In *IEEE International Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2003. VECIMS '03. 2003*, pages 39–44, 2003.

[Liu *et al.*, 2015] Hong Liu, Peng Zhang, Bin Hu, and Philip Moore. A novel approach to task assignment in a cooperative multi-agent design system. *Applied Intelligence*, 43:162–175, 2015.

[Lyu *et al.*, 2018] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *International conference on machine learning*, pages 3306–3314. PMLR, 2018.

[Mavridis *et al.*, 2016] Panagiotis Mavridis, David Gross-Amblard, and Zoltán Miklós. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. *Proceedings of the 25th International Conference on World Wide Web*, 2016.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proc. of Workshop at ICLR*, 2013.

[Mo *et al.*, 2013] Luyi Mo, Reynold Cheng, Ben Kao, Xuan S. Yang, Chenghui Ren, Siyu Lei, David Wai-Lok Cheung, and Eric Lo. Optimizing plurality for human intelligence tasks. *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013.

[Nickel *et al.*, 2016] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[Park *et al.*, 2017] Sung Joo Park, Bumhee Bae, Joungho Kim, and Madhavan Swaminathan. Application of machine learning for optimization of 3-d integrated circuits and systems. *IEEE transactions on very large scale integration (VLSI) systems*, 25(6):1856–1865, 2017.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. *Proceedings of the 24th International Conference on World Wide Web*, 2015.

[Zheng *et al.*, 2015] Yudian Zheng, Jiannan Wang, Guoliang Li, Reynold Cheng, and Jianhua Feng. Qasca: A quality-aware task assignment system for crowdsourcing applications. *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015.