SemDistill: Bootstrapping a Low-Latency, Low-Cost Semantic Table Annotator from Noisy LLM

Abstract

Relational tables widely exist in enterprise data lakes and repositories, yet they often lack explicit semantic context such as column types and relationships, creating a bottleneck for automated data management. *Semantic Table Annotation*, comprising tasks like *Column Type Annotation* (CTA) and *Column Property Annotation* (CPA), is essential to restoring missing semantic context. While Large Language Models (LLMs) excel at these tasks, their scalable deployment is hindered by prohibitive costs, high latency, and privacy information leakage. Distilling LLM capabilities into compact local models offers a viable solution; however, these models inevitably overfit to the structured noise in the teacher's predictions. In this paper, we formalize the setting of label-efficient distillation from noisy LLMs. Our empirical analysis reveals two error patterns: (1) *Systematic Ontological Drift*, a global class-conditional bias where LLMs tend to over-generalize specific concepts; and (2) *High-Confident Hallucinations*, stubborn high-confidence errors on ambiguous data. To address these, we propose SemDistill, a framework that bootstraps high-performance local annotators from noisy LLM outputs using only scarce trusted anchors. SemDistill employs a divide-and-conquer strategy: it rectifies global drift via anchor-guided statistical correction and identifies residual hallucinations through trajectory-aware learning dynamics. Extensive experiments demonstrate that SemDistill effectively addresses the cost-quality trade-off. It outperforms the LLM teacher (GPT-5-mini) and the distillation baseline while reducing inference costs by **3800×** and latency by **2600×**. The source code is available at an anonymous repository[1].

ACM Reference Format:

. 2018. SemDistill: Bootstrapping a Low-Latency, Low-Cost Semantic Table Annotator from Noisy LLM. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Relational tables serve as the universal carrier for structured knowledge, widely existing across large-scale enterprise data lakes and repositories [4, 43]. However, raw tables often lack clear semantic meaning; their headers are frequently ambiguous, and the relationships between columns are difficult to interpret. This semantic gap severely impedes automated data management: without standardized concepts, schema discovery [1, 8] fails to identify meaningful table relationships, data integration struggles to align heterogeneous vocabularies [23, 36]. To bridge this gap, *Semantic Table Annotation* is indispensable. In this work, we focus on two fundamental tasks: *Column Type Annotation* (CTA) [14, 24], which infers the semantic type of a column (e.g., classifying a list of names as *"Director"*), and *Column Property Annotation* (CPA) [33, 47], which identifies the semantic relationship between column pairs (e.g., *"directed_by"*).

Traditionally, the state-of-the-art approach relies on *Small Language Models* (SLMs), such as BERT-based encoders [4, 33], which are fine-tuned on massive labeled datasets. While effective, these supervised methods face a severe cold-start bottleneck in real-world deployments: enterprise data lakes contain distinct, private schemas that publicly available datasets fail to cover. Consequently, deploying SLMs requires expensive, expert-driven annotation of thousands of data for every new domain, making the process unscalable. Large Language Models (LLMs), such as GPT-5 [25] and Gemini [3], offer an appealing alternative. With their remarkable zero-shot reasoning capabilities, LLMs can annotate tables without task-specific training data [6, 9, 10, 21, 48]. However, deploying LLMs directly for high-throughput data lake management faces three prohibitive barriers: **(1) Prohibitive Cost:** Running full-scale inference on millions of columns incurs unsustainable expenses (e.g., $>$ \$500 per million columns). **(2) High Latency:** The multi-second API latency (e.g., $\sim$ 4.15s/col) fails to meet the real-time interactivity requirements of modern data catalogs. **(3) Privacy Constraints:** Sending sensitive enterprise data to external APIs often violates strict compliance regulations. Thus, while LLMs solve the *label scarcity* problem, they introduce critical *deployment barriers*.

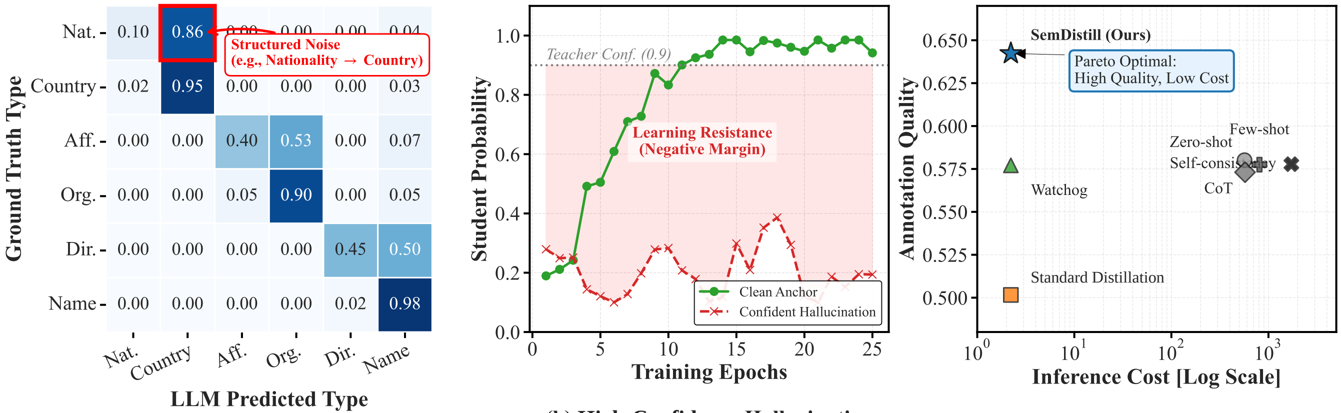
To resolve this dilemma, *Knowledge Distillation* [11] emerges as the natural convergence point: using the LLM as a teacher to generate pseudo-labels for training a compact, local student (SLM). This paradigm promises to combine the zero-shot generalization of LLMs with the inference efficiency and privacy of SLMs. Yet, this approach introduces a new trouble: the noise propagation trap. LLMs are not perfect oracles; they inevitably produce errors and hallucinations. In a naive distillation setup, the student model blindly mimics the teacher's noisy predictions, leading to poor performance. Crucially, we cannot rely on massive human intervention to filter this noise, as that would revert us to the original supervision bottleneck. We operate under a realistic label-efficient setting, assuming access to only a scarce set of trusted anchors (e.g., $\sim$ 20 verified examples per type). The core challenge, therefore, is how to bootstrap a robust student from noisy LLM signals using minimal human guidance.

To address this, we must first understand the nature of LLM errors in tabular domains. We conducted an empirical analysis on real-world datasets (e.g., VizNet [12]), which reveals that LLM noise is not uniformly distributed—as typically assumed in robust learning benchmarks—but exhibits structured, systematic patterns:

[1] https://anonymous.4open.science/r/SemTabAnn-8382

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX



(a) Asymmetric Ontological Drift

(b) High-Confidence Hallucinations

(c) Cost-Quality Pareto Frontier

Figure 1: Analysis of LLM Noise Patterns and Performance Trade-offs. (a) Asymmetric Drift: Confusion matrix showing structural bias where specific types are generalized to generic parents. (b) Learning Resistance: Training dynamics reveal that students struggle to fit confident hallucinations (red) compared to clean anchors (green). (c) Pareto Frontier: SemDistill (top-left) uniquely achieves teacher-level quality at student-level cost, outperforming existing baselines.

- **Finding 1: Systematic Ontological Drift (Global Bias)** First, we observe a dominant class-conditional error pattern. The confusion matrix (Figure 1a) is highly sparse and asymmetric: rather than making random uniform errors, the probability of error is strictly dependent on the semantic category. Specifically, LLMs systematically misclassify fine-grained concepts (e.g., *Nationality*) into generic ancestors (e.g., *Country*) with high probability. This class-conditional bias fundamentally distorts the label distribution, which cannot be corrected by simple instance filtering.
- **Finding 2: High-Confidence Hallucinations (Local Miscalibration).** Second, beyond global bias, we observe stubborn instance-specific errors. Specifically, on ambiguous samples, the teacher exhibits high certainty (assigning > 0.9 verbalized confidence[2]) to incorrect labels. However, the student explicitly resists learning them. As visualized in Figure 1b, clean anchors (green) converge rapidly, whereas these confident hallucinations (red) exhibit continuous low-margin fluctuations. This distinct training dynamic enables the detection of errors overlooked by global correction.

These observations collectively reveal that LLM-generated supervision constitutes a unique compound noise regime, fundamentally different from the random noise typically assumed in literature. Building a robust student model in this context is non-trivial: simply applying existing Learning with Noisy Labels (LNL) frameworks is insufficient because the structured bias and confident hallucinations of LLMs directly violate their core assumptions (e.g., instance independence and reliability of confidence). Consequently, we must address three specific challenges arising from this mismatch:

Challenge I: *How to rectify global ontological drift when the noise distribution is structurally biased?* The first hurdle lies in the limitations of the *small-loss hypothesis* adopted by representative LNL methods [20]. These approaches typically employ Gaussian Mixture Models (GMM) to separate clean and noisy samples, assuming that incorrect labels produce higher losses during early training. However, this assumption collapses under the systematic nature of Finding 1. Since the LLM exhibits a consistent structural bias (e.g., systematically mapping *Specific* → *Generic*), the student model readily fits this regular pattern just as it fits the ground truth, resulting in low loss even for incorrect samples. Consequently, distinguishing this systematic drift from the true underlying distribution becomes mathematically ill-posed without a mechanism to explicitly decouple bias from semantics.

Challenge II: *How to identify confident hallucinations precisely for targeted correction?* Traditional selection strategies rely on static uncertainty (e.g., Entropy in Active Learning). However, as noted in Finding 2, the teacher is misleadingly confident about its hallucinations, making them indistinguishable from correct labels in a static snapshot. To break this dilemma, we leverage the temporal behavior observed in our analysis, where the student model consistently struggles to fit these hallucinations compared to clean data. This aligns with the memorization effect in deep learning [2], which states that neural networks prioritize learning generalizable patterns before memorizing irregular noise. The crucial task is to operationalize this observation and quantify the training dynamics to isolate high-risk instances specifically for efficient correction.

Challenge III: *How to efficiently integrate corrections without signal dilution?* Our high-quality supervision comes from two scarce sources: the initial trusted anchors and the surgical corrections. However, these gold samples are vastly outnumbered by the massive noisy pseudo-labels (silver). Naively mixing these heterogeneous sources creates a dominance problem: standard optimization objectives (e.g., cross-entropy) average gradients over the entire batch, causing the abundant noisy data to inevitably overwhelm the sparse supervision signals. The challenge is to design a unified objective that allows the student to absorb general knowledge from the noisy corpus while strictly prioritizing the guidance from these sparse trusted data, ensuring their high-quality signal is not diluted.

[2] We obtain confidence scores from a black-box LLM by explicitly prompting the model to output a self-evaluated probability (from 0.0 to 1.0) alongside its prediction.

To address these challenges, we propose SemDistill, a cost-effective framework for bootstrapping high-quality semantic table annotators from noisy LLMs. SemDistill adopts a divide-and-conquer strategy to handle the two noise patterns identified above. First, to tackle *Asymmetric Drift* (C1), it employs a global statistical correction module. We utilize a class-conditional transition matrix as a robust approximation to rectify the dominant systematic bias, ensuring the student captures the correct schema topology. Second, to handle the residual *High-Confidence Hallucinations* (C2) that escape global correction, it leverages trajectory-aware profiling by tracking the learning trajectory of each sample over epochs. Instances that exhibit persistent learning resistance are automatically isolated as potential hallucinations. Third, to optimize efficiency (C3), it introduces a surgical correction strategy that selectively triggers oracle intervention only for high-risk samples. Finally, a bi-quality distillation objective seamlessly integrates the trusted data both initial anchors and surgical corrections with the globally calibrated noisy data.

Once trained, the student model is fully decoupled from the LLM, enabling 100% local, privacy-preserving inference at deployment. This strategic decoupling effectively retains the teacher's semantic reasoning capabilities while eliminating its deployment overheads. As visualized in Figure 1c, this holistic design allows SemDistill to occupy the unique pareto optimal position (top-left). While existing students (e.g., Standard Knowledge Distillation, Watchog [24]) cluster in the low-cost/low-quality region and teachers (e.g., GPT-5-mini) remain prohibitively expensive, our approach successfully resolves this efficiency-quality trade-off, matching teacher-level quality at student-level cost (≈ $0.13). Our contributions are summarized as follows:

- **Empirical Insights:** We characterize LLM supervision as a unique compound noise regime, distinguishing systematic ontological drift and stubborn high-confidence hallucinations.
- **Problem Formulation:** We formalize the setting of label-efficient distillation from noisy LLMs, aiming to bootstrap robust local annotators using only scarce trusted anchors.
- **Methodology:** We introduce a dual-granularity profiling mechanism that rectifies systematic drift via global transition matrices and identifies instance-level hallucinations through trajectory-aware learning dynamics.
- **Optimization:** We design a bi-quality distillation objective that efficiently synthesizes heterogeneous signals, trusted anchors, surgical corrections, and calibrated noisy labels, into a unified framework.
- **Performance:** Extensive experiments demonstrate that SemDistill outperforms both the LLM teacher (GPT-5-mini) and weak-supervised distillation baselines while reducing inference costs by **3800**× and latency by **2600**×.

2 Preliminary

Semantic table annotation is a fundamental capability for transforming raw data lakes into actionable assets. As discussed in previous sections, it serves as the cornerstone for downstream tasks like data integration and governance. Formally, we define a relational table T as a tuple comprising metadata and content. A table consists of n rows and m columns, denoted as $C = \{c_1, c_2, \ldots, c_m\}$. Each column c_i is associated with two components: (i) the header h_i, which provides a linguistic description of the column content; and (ii) a sequence of cell values $\mathcal{V}_i = \{v_{1,i}, v_{2,i}, \ldots, v_{n,i}\}$. Without loss of generality, we treat headers and cell values as textual tokens. In the following, we provide the formal definitions for the two primary tasks: Column Type Annotation (CTA) and Column Property Annotation (CPA).

Definition 2.1. (Column Type Annotation (CTA)). Given a column $c \in C$ and a predefined ontology of semantic types $\mathcal{Y}_{\text{type}}$ (e.g., $\mathcal{Y}_{\text{type}} = \{\textit{Person}, \textit{Location}, \ldots\}$), CTA aims to identify a semantic label $y \in \mathcal{Y}_{\text{type}}$ for c. Formally, the goal is to learn a mapping function $f_{\text{cta}} : C \to \mathcal{Y}_{\text{type}}$ such that the predicted type correctly describes the semantic category of the cell values in c.

Definition 2.2. (Column Property Annotation (CPA)). Given a pair of columns (c_i, c_j) from the same table T $(i \neq j)$ and a set of semantic relations $\mathcal{Y}_{\text{prop}}$ (e.g., $\mathcal{Y}_{\text{prop}} = \{\textit{directed\_by}, \textit{born\_in}, \ldots\}$), CPA aims to identify a relation label $r \in \mathcal{Y}_{\text{prop}}$ that holds between them. Formally, the goal is to learn a mapping function $f_{\text{cpa}} : C \times C \to \mathcal{Y}_{\text{prop}}$ such that r accurately describes the relationship between the entities in c_i and c_j.

Unified Notation. To establish a unified framework, we denote an input instance as $x \in \mathcal{X}$, representing either a single column or a column pair. The label space is denoted as $\mathcal{Y} = \{1, \ldots, K\}$. The goal is to learn a student model $f_\theta : \mathcal{X} \to \Delta^{K-1}$ that minimizes prediction error against the ground truth y^*.

Problem Statement. We formulate the problem of label-efficient distillation from noisy LLM supervision.

Input. We operate under a realistic semi-supervised setting with three key resources:

(1) Unlabeled Data $\mathcal{U} = \{x_i\}_{i=1}^N$: A massive collection of raw table instances from the data lake, where the ground truth labels y_i^* are latent and unknown.
(2) Trusted Anchors $\mathcal{D}_{clean} = \{(x_j, y_j)\}_{j=1}^{N_{clean}}$: A scarce set of human-verified samples drawn from the true distribution, where $N_{clean} \ll N$ (e.g., $N_{clean} \approx 20$ per class).

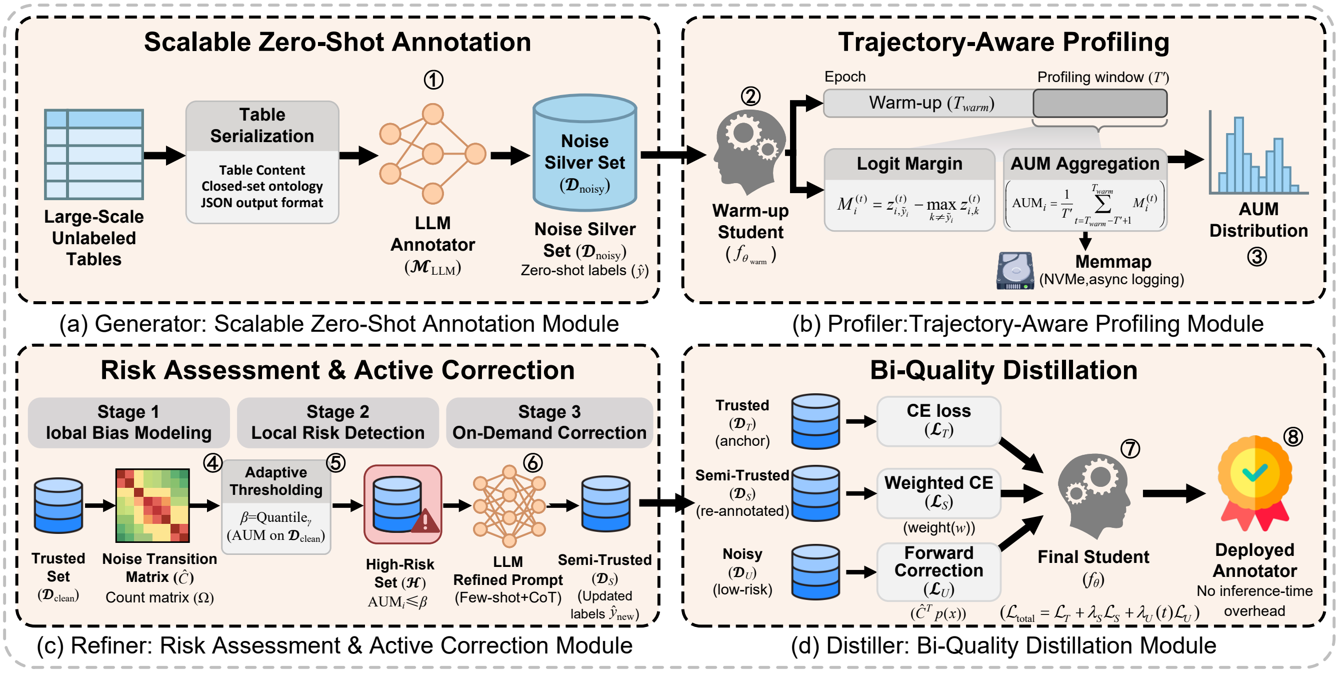
Weak Supervision Generation. Since $\mathcal{U}$ is unlabeled, we employ $\mathcal{M}_{\text{LLM}}$ as a teacher to generate pseudo-labels for all instances in $\mathcal{U}$. For each x_i, we obtain a weak label $\tilde{y}_i = \mathcal{M}_{\text{LLM}}(x_i)$. This process yields a noisy dataset $\mathcal{D}_{noisy} = \{(x_i, \tilde{y}_i)\}_{i=1}^N$, where the label distribution $P(\tilde{y}|x)$ inherently contains systematic biases and hallucinations compared to the true distribution $P(y^*|x)$.

Objective. Directly training on $\mathcal{D}_{noisy}$ leads to a biased student that overfits the teacher's errors. Our goal is to leverage the scarce $\mathcal{D}_{clean}$ to rectify the noisy signals in $\mathcal{D}_{noisy}$ and learn a student model f_θ that minimizes the expected risk on the true distribution $\mathcal{D}_{true}$:

$$\theta^* = \arg\min_\theta \mathbb{E}_{(x,y^*)\sim\mathcal{D}_{true}} [\ell(f_\theta(x), y^*)]. \tag{1}$$

where $\ell(\cdot, \cdot)$ denotes a standard loss function (e.g., cross-entropy). The fundamental challenge lies in decoupling the systematic bias $P(\tilde{y}|y^*)$ and local hallucinations within $\mathcal{D}_{noisy}$ to recover the underlying true distribution using only minimal trusted guidance.

Student Model. While SemDistill is agnostic to the specific backbone architecture, we require a student model f_θ capable of encoding the 2D topological structure of relational tables. We formulate


Scalable Zero-Shot Annotation
Large-Scale Unlabeled Tables
Table Serialization
Table Content
Closed-set ontology
JSON output format
①
LLM Annotator ($\mathcal{M}_{\text{LLM}}$)
Noise Silver Set ($\mathcal{D}_{\text{noisy}}$)
Noise Silver Set ($\mathcal{D}_{\text{noisy}}$)
Zero-shot labels ($\hat{y}$)
Trajectory-Aware Profiling
②
Warm-up Student ($f_{\theta_{\text{warm}}}$)
Epoch
Warm-up (T_{warm})
Profiling window (T')
Logit Margin
$M_i^{(t)} = z_{i,\tilde{y}_i}^{(t)} - \max_{k \neq \tilde{y}_i} z_{i,k}^{(t)}$
AUM Aggregation
$\text{AUM}_i = \frac{1}{T'} \sum_{t=T_{warm}-T'+1}^{T_{warm}} M_i^{(t)}$
Memmap
(NVMe,async logging)
AUM Distribution
③
Risk Assessment & Active Correction
Stage 1
lobal Bias Modeling
Stage 2
Local Risk Detection
Stage 3
On-Demand Correction
Trusted Set ($\mathcal{D}_{\text{clean}}$)
Noise Transition Matrix ($\hat{C}$)
Count matrix (Ω)
④
Adaptive Thresholding
β=Quantile$_\gamma$
(AUM on $\mathcal{D}_{\text{clean}}$)
⑤
High-Risk Set ($\mathcal{H}$)
$\text{AUM}_i \leqslant \beta$
⑥
LLM Refined Prompt
(Few-shot+CoT)
Semi-Trusted ($\mathcal{D}_S$)
(Updated labels $\hat{y}_{\text{new}}$)
Bi-Quality Distillation
Trusted ($\mathcal{D}_T$) (anchor)
Semi-Trusted ($\mathcal{D}_S$) (re-annotated)
Noisy ($\mathcal{D}_U$) (low-risk)
CE loss ($\mathcal{L}_T$)
Weighted CE ($\mathcal{L}_S$)
(weight(w))
Forward Correction ($\mathcal{L}_U$)
($\hat{C}^T p(x)$)
⑦
Final Student (f_θ)
($\mathcal{L}_{\text{total}} = \mathcal{L}_T + \lambda_S \mathcal{L}_S + \lambda_U(t)\mathcal{L}_U$)
⑧
Deployed Annotator
No inference-time overhead

(a) Generator: Scalable Zero-Shot Annotation Module

(b) Profiler:Trajectory-Aware Profiling Module

(c) Refiner: Risk Assessment & Active Correction Module

(d) Distiller: Bi-Quality Distillation Module

Figure 2: The SemDistill Framework.

the input instance x (representing a column or column-pair) as a serialized sequence of tokens. To capture schematic dependencies, we adopt a structure-aware serialization protocol similar to DoDUO [33]. Let $\mathcal{S}(x)$ be the linearized sequence enriched with column-type and segment embeddings. The student encoder maps this sequence to a dense representation $\mathbf{e}(x) \in \mathbb{R}^d$:

$$\mathbf{p}(x) = \text{Softmax}(W_h \cdot f_\theta(\mathcal{S}(x))), \tag{2}$$

where $\mathbf{p}(x) \in \Delta^{K-1}$ is the predicted probability distribution over K semantic classes. Note that our primary contribution lies not in the network architecture, but in the noise-resilient learning paradigm that enables standard encoders to reach state-of-the-art performance using imperfect supervision.

3 Framework Overview

As illustrated in Figure 2, the SemDistill pipeline operates on two input sources: a massive unlabeled dataset $\mathcal{U}$ and a scarce set of trusted anchors $\mathcal{D}_{clean}$. The framework comprises four integrated modules, transforming raw data into an optimized model f_θ.

Architecture. ❶ *Generator: Scalable Zero-shot Annotation (Sec. 4.1).* To overcome the latency and cost barriers of online LLM inference, this module utilizes the LLM as an offline labeling function. It projects the unlabeled instances in $\mathcal{U}$ into a silver-standard dataset $\mathcal{D}_{noisy}$ via zero-shot prompting. ❷ *Profiler: Trajectory-Aware Noise Profiling (Sec. 4.2).* This is the diagnostic engine of our framework. It trains a warm-up student model on the noisy data to log instance-level training dynamics. ❸ *Refiner: Risk Assessment and Correction (Sec. 4.3).* This decision engine analyzes the training dynamics to characterize noise at two levels: *(i) Global Bias Modeling:* It estimates a noise transition matrix that quantifies the systematic confusion patterns of the Teacher LLM (e.g., consistently mislabeling *City* as *Location*). *(ii) Local Risk Assessment:* It utilizes the Area Under the Margin (AUM) statistic to identify high-risk samples that resist the model's inductive bias. These high-risk samples undergo on-demand correction, splitting the data into trusted, semi-trusted, and noisy partitions. ❹ *Distiller: Bi-Quality Optimization (Sec. 4.4).* The final module synthesizes the heterogeneous supervision signals. It minimizes a unified objective that applies standard supervision to trusted data while employing forward correction to robustly learn from the vast noisy partition.

System Workflow. When a raw tabular corpus is ingested, the *Generator* first produces preliminary noisy labels and a calibration set. The *Profiler* then initiates a warm-up phase, streaming training dynamics to a memory-mapped buffer. Subsequently, the *Refiner* analyzes these dynamics to distinguish between reliable supervision and structured noise, selectively triggering re-annotation for high-risk instances. Finally, the *Distiller* synthesizes the refined signals to train the final student model. During deployment, the optimized student operates as a standalone encoder, ensuring low-latency inference with zero dependency on external LLMs.

4 The SemDistill Framework

In this section, we present SemDistill, a holistic framework designed to distill high-quality semantic table annotators from noisy LLM supervision with minimal human effort.

4.1 Scalable Zero-Shot Annotation

In the initial phase, we leverage the LLM as an offline, noisy annotator. Our primary objective is not immediate perfection, but rather the efficient transfer of the LLM's annotation capabilities to the student model, establishing a warm start for subsequent refinement.

LLM as an Offline Noisy Teacher. Directly deploying LLMs for inference on gigabyte-scale data lakes is often infeasible due to latency (seconds per query) and cost constraints. Therefore, SemDistill utilizes the LLM strictly as an offline annotator to create a silver standard training corpus. Given the unlabeled corpus $\mathcal{U}$, we perform zero-shot annotation on every instance $x \in \mathcal{U}$ to generate a pseudo-label $\tilde{y}$. This results in a massive but noisy dataset $\mathcal{D}_{noisy} = \{(x_i, \tilde{y}_i)\}_{i=1}^{N}$. Crucially, at this stage, we prioritize coverage over perfection. We acknowledge that the LLM may generate incorrect annotation, particularly with domain-specific acronyms or ambiguous values. However, we rely on the subsequent distillation process to correct these errors, rather than attempting to create a perfect teacher from the start.

To achieve this, we use a zero-shot prompting strategy to construct the initial dataset $\mathcal{D}_{noisy}$. Unlike few-shot prompting, which requires inserting multiple examples into the context window and significantly increases token consumption per API call, zero-shot prompting leverages the LLM's internalized world knowledge. Specifically, the prompt encapsulates four distinct components: (1) *Task Definition:* Instructs the model to assume a specific role (e.g., data quality expert) and outlines the annotation logic; (2) *Domain Constraints:* Explicitly provides the closed-set label ontology ("Class List") to prevent open-ended generation; (3) *Output Formatting:* Enforces a strict JSON structure to ensure parsability; (4) *Data Context:* Injects the serialized table content (Only header index and cell values) as the target for annotation. It is worth noting that the prompt design for CPA follows a symmetric structure, differing primarily in input serialization (column pairs versus single columns). Due to space limitations, we provide the full prompt in appendix.

Despite explicit constraints, LLMs may still produce unstructured responses, such as additional punctuation, conversational fillers, or semantically equivalent synonyms. To reliably convert the raw textual output r into a discrete label, we implement a strict parsing function ϕ : String $\rightarrow \mathcal{Y} \cup \{\text{OTHER}\}$. Specifically, we first check whether r exactly matches any canonical label name in the closed set $\mathcal{Y}$. If this check fails, we apply an alias-resolution step that maps common synonyms (e.g., "Nation" → "Country") using a predefined dictionary. Any response that cannot be resolved by either rule is mapped to "OTHER" and excluded it. After parsing and filtering, we obtain the noisy training set $\mathcal{D}_{noisy} = \{(x_i, \tilde{y}_i)\}_{i=1}^{N}$, which serves as the silver-standard supervision for the subsequent profiling phase.

4.2 Trajectory-Aware Profiling

Unlike standard distillation methods that treat pseudo-labels as static supervision targets, SemDistill explicitly models the compatibility between a noisy label and the student model's inductive bias. We posit that the correctness of a label is not solely reflected in the final prediction confidence, but more fundamentally in how consistently the model aligns with this label throughout the training dynamics.

Modeling Label-Model Compatibility. Formally, for each pseudo-labeled instance $(x_i, \tilde{y}_i) \in \mathcal{D}_{noisy}$, we aim to estimate a latent variable $r_i \in \{\text{Compatible}, \text{Conflicting}\}$. A *Compatible* state implies the assigned label $\tilde{y}_i$ is semantically consistent with the generalizable patterns learned by the model, whereas a Conflicting state suggests a mismatch between the label and the underlying features.

To implement this, we exploit the memorization effect of deep networks [46]: models preferentially learn simple, clean patterns early in training before overfitting to noise. Consequently, clean samples typically exhibit stable, monotonically increasing confidence, whereas mislabeled samples manifest as persistent gradient conflicts, the model's inductive bias repeatedly rejects the assigned label. We capture this temporal divergence through a two-step profiling process.

Noise-Agnostic Warm-up. We first initiate a warm-up training phase to expose the student model f_θ to the raw noisy dataset $\mathcal{D}_{noisy}$. Crucially, this phase employs standard supervised learning without any correction logic. This strategy serves a dual purpose: it enables the student to capture generalizable semantic structures (e.g., date formats, numerical patterns) while establishing a stable feature representation for subsequent risk analysis. We train for a fixed number of epochs T_{warm} using the standard Cross-Entropy loss:

$$\mathcal{L}_{\text{init}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{I}(\tilde{y}_i = k)\log p_k(x_i). \tag{3}$$

Margin Trajectory Analysis. To quantify the resistance of the model to the assigned labels, we record the training dynamics during the late stages of the warm-up. We focus on the *logit margin*, defined as the difference between the assigned class logit and the largest other logit. For a sample x_i at epoch t, the margin is:

$$M_i^{(t)} = z_{i,\tilde{y}_i}^{(t)} - \max_{k \neq \tilde{y}_i} z_{i,k}^{(t)}. \tag{4}$$

where $\mathbf{z}_i^{(t)}$ is the logits, a positive margin ($M_i^{(t)} > 0$) indicates that the model currently favors the assigned label, while a negative margin reflects a conflict. Instead of relying on a single snapshot, which may be unstable, we aggregate this signal over a profiling window T' to compute the Area Under the Margin (AUM) [28]:

$$\text{AUM}_i = \frac{1}{T'}\sum_{t=T_{warm}-T'+1}^{T_{warm}} M_i^{(t)}. \tag{5}$$

The resulting AUM$_i$ serves as a robust statistic for label quality. As visualized in Figure 3, we observe a distinct distributional shift: high AUM values indicate learned compatibility (reliable labels), while low or negative AUM values signal training dynamics that contradict the noisy supervision (potential errors).

Implementation Detail: Storing full trajectories for millions of instances in GPU memory is impossible. We implement an asynchronous streaming buffer: logits are detached from the computation graph and flushed to a memory-mapped file (NumPy memmap) on the NVMe SSD at the end of each epoch. This allows us to scale to large-scale datasets with minimal memory overhead, enabling the offline computation of AUM in the next component.

4.3 Risk Assessment and Correction

This phase acts as the decision-making engine of SemDistill. It analyzes the profiling statistics derived in Section 4.2 to systematically identify high-risk instances and trigger targeted corrections, followed by a reliability-based data partitioning.

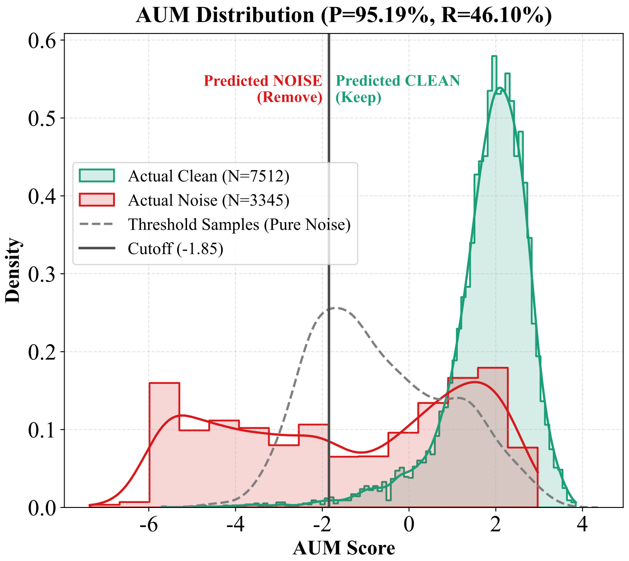


Figure 3: Analysis of Training Dynamics on VizNet.

Global Modeling via Transition Matrix Estimation. First, we aim to capture the macroscopic error patterns of the teacher LLM. In semantic annotation tasks, label noise is rarely uniform; it is heavily structured by semantic proximity (e.g., an LLM might consistently predict *Artist* for *Person*). We formalize this structure via a global noise transition matrix $C \in \mathbb{R}^{K\times K}$, where an entry $C_{jk} = P(\tilde{Y} = k \mid Y = j)$ represents the probability that the noisy teacher predicts class k when the true class is j.

Estimating C typically requires extensive paired data. However, given only the scarce trusted set $\mathcal{D}_{clean}$, a naive maximum likelihood estimation would result in a sparse matrix with high variance. To mitigate this data sparsity problem, we employ a regularized estimation approach using Laplacian Smoothing. Let $\Omega \in \mathbb{R}^{K\times K}$ be the raw count matrix where Ω_{jk} denotes the number of samples in $\mathcal{D}_{clean}$ with true label j and LLM prediction k. The smoothed estimator is given by:

$$\hat{C}_{jk} = \frac{\Omega_{jk} + \alpha}{\sum_{k'=1}^{K} (\Omega_{jk'} + \alpha)}. \tag{6}$$

Here, the hyperparameter $\alpha > 0$ (empirically set to 0.1) acts as a Dirichlet prior, ensuring that the student model remains robust to rare confusion patterns that may not be fully manifested in the small $\mathcal{D}_{clean}$ during the subsequent forward correction phase.

Local Risk Detection via Adaptive Thresholding. While $\hat{C}$ captures the average bias, pinpointing specific errors requires instance-level resolution. We utilize the AUM statistics derived in Section 4.2 as the primary risk indicator. As illustrated in Figure 3, the AUM distribution exhibits a clear bimodal structure, where the valley between peaks suggests a natural decision boundary between compatible (clean) and conflicting (noisy) samples.

To strictly separate these high-risk outliers without manual tuning, we implement a data-driven calibration strategy. Instead of a fixed scalar threshold, we treat the trusted set $\mathcal{D}_{clean}$ as a compatibility anchor. Since $\mathcal{D}_{clean}$ contains verified labels, its AUM distribution represents the model's behavior on valid data. We determine the cutoff threshold β corresponding to the γ-th percentile of this reference distribution:

$$\beta = \text{Quantile}_{\gamma}\left(\{\text{AUM}_j \mid (x_j, y_j) \in \mathcal{D}_{clean}\}\right). \tag{7}$$

This adaptive approach ensures that the selection criteria dynamically adjust to the learning difficulty of the specific dataset. As demonstrated by the vertical cutoff line in Figure 3, this strategy effectively isolates the high-risk mode (left tail). Notably, as indicated by the metrics in the figure (P=95.19%), this conservative thresholding prioritizes high precision in noise identification. This ensures that our subsequent on-demand corrections effectively address actual errors, preventing the incorrect relabeling of correct labels. The high-risk set is then identified as $\mathcal{H} = \{i \in \mathcal{D}_{noisy} \mid \text{AUM}_i \leq \beta\}$, effectively capturing samples that contradict the model's inductive bias.

Active Correction and Partitioning. Once the high-risk candidates $\mathcal{H}$ are identified, we proceed to the correction phase. To ensure cost-efficiency, we perform on-demand active correction exclusively on $\mathcal{H}$. Since simply repeating the zero-shot prompt is often ineffective, we employ two enhancement strategies for the re-annotation prompt $\mathcal{P}_{refined}$: (1) Chain-of-Thought (CoT) Reasoning, instructing the LLM to analyze semantic consistency before labeling; and (2) Few-shot In-Context Learning, where we inject a small set of manually corrected examples into the context to guide the model's reasoning logic.

Following re-annotation, we incorporate these refined signals into the training process. Since these labels are generated by a stronger inference process (Few-shot + CoT) but lack human verification, we treat them as semi-trusted. Accordingly, we stratify the corpus into three reliability partitions:

- **Trusted Partition ($\mathcal{D}_T$):** Consists strictly of the clean seed data $\mathcal{D}_{clean}$. These samples serve as the immutable semantic anchors.
- **Semi-Trusted Partition ($\mathcal{D}_S$):** Contains the high-risk samples that underwent re-annotation ($i \in \mathcal{H}$). We utilize the updated labels $\tilde{y}_{new}$ as improved but soft supervision targets.
- **Noisy Partition ($\mathcal{D}_U$):** Contains the vast majority of samples deemed low-risk ($i \notin \mathcal{H}$). These retain their original zero-shot labels $\tilde{y}_{old}$ and are handled via forward noise correction.

4.4 Bi-Quality Distillation

In the final phase, we re-initialize the student model to train a robust annotator. To effectively assimilate the heterogeneous supervision signals derived from the previous partitioning, we formulate a unified framework that applies distinct loss functions tailored to the reliability of each data tier.

Supervision from Trusted and Semi-Trusted Data. For the trusted partition $\mathcal{D}_T$, which combines the clean seed data $\mathcal{D}_{clean}$ and high-confidence corrections, we assume the labels are ground truth. This subset serves as the semantic "anchor", preventing the model from drifting due to noise correction artifacts. We minimize the standard Cross-Entropy (CE) loss:

$$\mathcal{L}_T(\theta) = \frac{1}{|\mathcal{D}_T|} \sum_{(x,y)\in\mathcal{D}_T} H(\mathbf{y}, \mathbf{p}(x)), \tag{8}$$

where $H(\mathbf{p}, \mathbf{q}) = -\sum_k p_k \log q_k$ is the cross-entropy function and $\mathbf{y}$ is the one-hot encoding of the true label.

However, for the semi-trusted partition $\mathcal{D}_S$, treating labels as absolute ground truth risks overfitting to hallucinations. Instead, we adopt a *Soft-Target Re-weighting* strategy. Each sample $(x, \tilde{y}, w)$ carries a confidence score $w \in [0, 1]$ from the Re-annotation phase. We use this score to modulate the gradient magnitude:

$$\mathcal{L}_S(\theta) = \frac{1}{|\mathcal{D}_S|} \sum_{(x,\tilde{y},w)\in\mathcal{D}_S} w \cdot H(\tilde{\mathbf{y}}, \mathbf{p}(x)). \tag{9}$$

Intuitively, this implies the loss contribution when the LLM is uncertain ($w \approx \tau_{mid}$), allowing the model's inductive bias (learned from $\mathcal{D}_T$) to potentially override weak or incorrect labels.

Noisy Supervision via Forward Correction. The noisy partition $\mathcal{D}_U$ typically constitutes the largest portion of the data (e.g., >80%). Standard training on $\mathcal{D}_U$ would force the student to mimic the teacher's systematic errors (e.g., learning that "New York" is of type *Location* instead of *City*). To utilize $\mathcal{D}_U$ without inheriting its bias, we employ the Forward Correction method [26]. The core idea is to model the noise generation process itself. We posit that the student model predicts the *latent clean distribution* $\mathbf{p}(z)$. Before computing the loss, we project this clean distribution through our estimated transition matrix $\hat{C}$ to simulate the teacher's noise:

$$\mathbf{p}_{noisy}(x) = \hat{C}^\top \mathbf{p}(x). \tag{10}$$

Here, the k-th element of $\mathbf{p}_{noisy}(x)$ represents the probability that the noisy teacher *would* predict class k, given the student's belief about the true class probabilities. The loss is then computed against the observed noisy label $\tilde{y}$:

$$\mathcal{L}_U(\theta) = \frac{1}{|\mathcal{D}_U|} \sum_{(x,\tilde{y})\in\mathcal{D}_U} H(\tilde{\mathbf{y}}, \hat{C}^\top \mathbf{p}(x)). \tag{11}$$

Theoretical Justification: Forward vs. Backward. A natural alternative is *Backward Correction*, which attempts to "clean" the noisy label by multiplying the loss with $\hat{C}^{-1}$. However, we explicitly choose Forward Correction for two reasons: (1) Numerical Stability: The estimated matrix $\hat{C}$ is often ill-conditioned or singular (e.g., when two classes are perfectly confused). Inverting it leads to exploding gradients. Forward correction involves only matrix multiplication, which is numerically stable. (2) Gradient Demixing: The gradient of $\mathcal{L}_U$ encourages the student to increase the probability of all "ancestor" classes j that could transition to the observed noisy class k (i.e., high $\hat{C}_{jk}$). Combined with the anchor data $\mathcal{D}_T$, the model effectively disambiguates which ancestor is the true label.

Finally, we combine these components into a composite multi-task objective function:

$$\mathcal{L}_{total}(\theta) = \mathcal{L}_T(\theta) + \lambda_S \mathcal{L}_S(\theta) + \lambda_U(t)\mathcal{L}_U(\theta). \tag{12}$$

Algorithm 1 SemDistill: LLM-Error-Aware Distillation

```
Require: Unlabeled corpus $\mathcal{U}$; Trusted dataset $\mathcal{D}_{clean}$; LLM Annotator $\mathcal{M}_{LLM}$; Risk percentile threshold $\gamma$ (e.g., 5%); Warm-up epochs $T_{warm}$.
Ensure: Optimized Student Model parameters $\theta^*$.
    Phase 1: Scalable Zero-Shot Annotation
    Generate noisy labels $\tilde{y}$ for $\mathcal{U}$ using $\mathcal{M}_{LLM} \rightarrow \mathcal{D}_{noisy}$.
    Phase 2: Trajectory-Aware Profiling
    Initialize warm-up student $\theta_{warm}$.
    for epoch $t = 1$ to $T_{warm}$ do
        Update $\theta_{warm}$ using $\mathcal{L}_{\text{init}}$ on $\mathcal{D}_{noisy}$.
        if $t > T_{warm} - T'$ then                ▷ Log logits in profiling window
            Record logits $\mathbf{z}_i^{(t)}$ to form trajectory $\mathcal{T}_i$.
        end if
    end for
    Phase 3: Risk Assessment & On-Demand Correction
    Estimate transition matrix $\hat{C}$ using $\mathcal{D}_{clean}$ with smoothing (Eq. 6).
    Compute AUM$_i$ for all instances via Eq. (8).
    Determine threshold $\beta \leftarrow$ Quantile($\{\text{AUM}_j \mid j \in \mathcal{D}_{clean}\}, \gamma$).
    Identify high-risk set $\mathcal{H} \leftarrow \{i \in \mathcal{D}_{noisy} \mid \text{AUM}_i \leq \beta\}$.    ▷ Adaptive selection
    Query $\mathcal{M}_{LLM}$ for $x \in \mathcal{H}$ with CoT prompts to get $(\tilde{y}^{(1)}, w)$.
    Partition data into $\mathcal{D}_T, \mathcal{D}_S, \mathcal{D}_U$ based on conf. thresholds $\tau_{high}, \tau_{mid}$.
    Phase 4: Bi-Quality Distillation
    Initialize final student $\theta$ (optionally warm-started).
    while not converged do
        Update curriculum weight $\lambda_U(t)$.
        Sample stratified batch $B_T \sim \mathcal{D}_T, B_S \sim \mathcal{D}_S, B_U \sim \mathcal{D}_U$.
        Compute losses $\mathcal{L}_T$, $\mathcal{L}_S$, $\mathcal{L}_U(\hat{C})$ (Eq. 15-18).
        Update $\theta$ to minimize $\mathcal{L}_{total} = \mathcal{L}_T + \lambda_S \mathcal{L}_S + \lambda_U \mathcal{L}_U$.
    end while
    return $\theta$
```

Training with Forward Correction relies heavily on the accuracy of $\hat{C}$ and the initial stability of the student. In the early epochs, the student's predictions are random, making the projected $\mathbf{p}_{noisy}$ meaningless. To stabilize training, we implement a Linear Warm-up Curriculum for the noisy loss weight $\lambda_U(t)$. Let t be the current epoch and T_{ramp} be the ramp-up period:

$$\lambda_U(t) = \lambda_{max} \cdot \min\left(1, \frac{t}{T_{ramp}}\right). \tag{13}$$

We start with $\lambda_U(0) = 0$, training solely on trusted and semi-trusted data to establish a robust feature representation. As the epoch count approaches T_{ramp}, we gradually introduce the noisy data, allowing the student to leverage the massive scale of $\mathcal{D}_U$ to refine its decision boundaries without being derailed by noise early on.

4.5 Complexity and Scalability Analysis

We analyze the computational efficiency of SemDistill. In terms of time complexity, although the framework involves two training rounds, the warm-up phase is intentionally short (typically 30% of standard training epochs), and the overhead from matrix estimation and AUM calculation is linear $O(N)$ and negligible compared to gradient backpropagation. The primary computational cost lies in LLM re-annotation; however, by design, our risk-aware strategy restricts this to the small high-risk subset $\mathcal{H}$ (typically $< 15\%$ of the corpus), and these queries are parallelizable via asynchronous API calls. Consequently, the total training time is approximately 1.3× to 1.5× that of standard fine-tuning. Regarding space complexity, storing trajectory logs requires $O(N \cdot K \cdot T')$ memory. For million-scale datasets, this amounts to a few gigabytes, which we handle efficiently using memory-mapped files to avoid RAM bottlenecks.

Crucially, SemDistill introduces zero latency overhead during inference, as the final deployed model architecture is identical to a standard Transformer encoder.

5 Experimental Evaluation

In this section, we conduct extensive experiments to evaluate the effectiveness of SemDistill. We aim to demonstrate that our student model, trained via error-aware distillation, can surpass its teacher and state-of-the-art baselines under strict label noise conditions.

Table 1: Statistics of CTA and CPA Benchmarks. Classes refers to types for CTA and relations for CPA. Labeled denotes labeled columns for CTA and labeled pairs for CPA.

Benchmark	# Tables	# Classes	Total Cols	Labeled	Avg. Density
VizNet	78,733	78	119,360	95,629	2.3
SOTABv2-CTA	45,834	82	120,507	101,855	8.1
GitTablesDB	3,737	101	45,304	5,433	12.1
GitTablesSC	2,853	53	34,148	3,863	12.0
SOTABv2-CPA	48,379	176	462,815	174,998	3.6
REDTab	9,146	23	44,664	22,236	2.4

5.1 Experimental Setup

Datasets. We conduct experiments on six widely used benchmarks covering both Column Type Annotation (CTA) and Column Property Annotation (CPA). For the CTA task, we employ four datasets: (1) VizNet [12], a large-scale corpus of web tables containing 78 semantic types; (2) SOTABv2-CTA [18], a high-quality benchmark annotated with the Schema.org ontology; and (3-4) GitTablesDB and GitTablesSC [13], two subsets derived from GitHub CSVs that present unique challenges with dirty data and non-standard headers. For CPA and Relation Extraction, we use: (5) SOTABv2-CPA [18], which focuses on identifying semantic relationships between column pairs to capture table structure; and (6) REDTab [32], a dataset derived from Wikipedia tables for diverse binary relation extraction. Detailed statistics for all datasets are summarized in Table 1.

Baselines. We compare SemDistill against two categories of methods to rigorously evaluate its effectiveness under label noise and data scarcity. *(1) LLM Inference Baselines:* We report the performance of the Teacher itself (Zero-shot) and advanced prompting strategies, including Few-shot (5-shot), Chain-of-Thought (CoT) [39], and Self-Consistency [37]. Additionally, we compare against recent table reasoning frameworks: Archetype [9], which performs schema matching via re-mapping, and Chorus [15], which decomposes complex reasoning tasks. These baselines rely on the frozen parametric knowledge of the LLM. *(2) Student Distillation Baselines:* Using Doduo [33] as the unified student backbone, we evaluate various training paradigms, ranging from naive baselines to state-of-the-art weak-supervised frameworks. We first include Trusted-Only (training exclusively on the scarce $\mathcal{D}_{clean}$) to assess the impact of data sparsity, and Standard Distillation (naive fine-tuning on the noisy $\mathcal{D}_{LLM}$) to measure the impact of label noise. We also report Mixture, which simply trains on the union of both datasets. Watchog [24] is a SOTA semi-supervised framework specifically designed for tabular tasks, utilizing contrastive learning and self-training techniques. In our configuration, we consider the trusted anchors $\mathcal{D}_{clean}$ as the labeled dataset and the extensive noisy corpus $\mathcal{D}_{LLM}$ as the unlabeled dataset. FreeAL [44], a weak supervision method utilizing LLM-SLM collaboration for label refinement. Finally, we report Supervised GT (trained on full ground truth) as the theoretical upper bound.

Evaluation Metrics. Following standard practice, we report Micro-F1 and Macro-F1 scores. Micro-F1 measures the overall instance-level accuracy, heavily influenced by dominant classes, while Macro-F1 averages performance across all semantic types, highlighting the model's robustness on long-tail and fine-grained categories.

Implementation Details. We employ GPT-5-mini as the default oracle teacher to generate the initial noisy training corpus and as the backbone for all LLM-based inference baselines (e.g., Zero-shot CoT, Chorus) to ensure a fair comparison; meanwhile, we utilize Qwen3-Max specifically for the selective re-annotation phase to leverage its advanced reasoning capabilities for resolving high-risk samples. The student model adopts the pre-trained Doduo [33] (12-layer Transformer, 768 hidden units), where input tables are serialized by allocating the 512-token budget equally across columns and truncating row-wise to preserve the top-most rows. The model is optimized using AdamW [16] with a learning rate of 2e-5 and a batch size of 64 for 50 epochs. We utilize PyTorch 2.8 and HuggingFace Transformers 4.56.2 on a server equipped with an Intel Xeon Silver 4316 CPU and an NVIDIA RTX 4090 GPU. Key hyperparameters include the loss balancing weights $\lambda_S = 1.0$ and $\lambda_U = 0.5$, and the noise transition matrix smoothing factor $\alpha = 0.01$. For risk detection, the AUM threshold parameter γ is set to 0.3 based on the distribution of trusted anchors (i.e., retaining samples with confidence margins superior to the bottom 30% of the trusted reference). All prompts we used are detailed in appendix.

5.2 Main Results

We present the comprehensive performance results in Table 2. The results show that SemDistill consistently outperforms both the teacher LLM and distilled baselines across diverse benchmarks.

Superiority of Student over Teacher. A significant finding is that our distilled student surpasses its teacher on structural understanding. On VizNet, SemDistill achieves a Macro-F1 of 53.98%, outperforming the Zero-shot Teacher (41.82%) by +12.16%. Similarly, on the high-quality SOTABv2-CTA, our model reaches 78.64%, surpassing the Teacher's 71.09%. While large language models possess vast world knowledge, they suffer from systematic confusion on fine-grained types. By explicitly modeling this confusion matrix and performing heterogeneous verification, SemDistill effectively "cleans" the teacher's knowledge, proving that a compact student can be more reliable than a generic LLM for domain-specific tasks.

Robustness against Noise. Comparing SemDistill with *Standard Distillation* (Standard KD) highlights the failure of noise-agnostic methods. On VizNet, Standard KD suffers a catastrophic performance drop to 25.63% Macro-F1, as it blindly fits the errors in $\mathcal{D}_{LLM}$. In contrast, SemDistill achieves a +28.35% gain over Standard KD. Furthermore, while *Trusted-Only* performs decently on clean datasets (38.06% on VizNet), it collapses on dirty datasets like GitTablesDB (17.25% Macro-F1). This underscores that relying

Table 2: Main Performance Comparison (Micro-F1 / Macro-F1 %) under the Realistic Mode. All models are evaluated using the Fixed Anchor protocol ($K = 20$). (Bold: Best Student; Underline: Second Best Student).

Method	VizNet		SOTABv2-CTA		GitTablesDB		GitTablesSC		SOTABv2-CPA		REDTab	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Teacher & LLM Baselines												
Zero-shot (Teacher)	59.91	41.82	76.97	71.09	43.98	28.73	51.17	31.11	65.06	59.91	65.40	41.04
Few-shot (5-shot)	59.16	41.33	77.35	70.72	44.12	28.08	50.35	32.30	65.94	61.15	77.06	51.64
CoT [39]	59.04	41.01	76.79	70.18	42.00	26.17	51.42	33.74	65.47	60.51	67.25	38.87
Self-Consistency [37]	59.83	42.96	76.83	70.82	44.09	27.20	50.35	31.74	62.07	55.84	52.23	36.87
Archetype [9]	45.53	41.15	65.87	64.88	31.97	23.89	34.81	24.68	48.32	45.32	68.25	37.94
Chorus [15]	56.46	38.45	72.85	64.58	43.28	27.21	47.93	28.86	60.67	54.50	64.98	36.95
Students & Weak Supervised Distillation Baselines (Model: Doduo)												
Distillation [11]	57.73	25.63	70.13	48.36	35.29	10.37	37.50	11.16	54.47	43.19	61.52	34.56
Trusted-Only	50.08	38.06	75.87	50.58	20.67	17.25	33.09	30.71	35.38	34.85	54.50	37.02
Mixture	60.36	42.95	75.21	65.70	46.72	30.61	47.30	20.02	54.54	43.64	62.42	25.71
Watchog [24]	61.23	41.32	75.22	65.90	**47.12**	30.12	47.23	20.13	56.31	56.90	61.66	42.09
FreeAL [44]	57.46	38.27	72.54	63.68	42.69	22.54	36.52	21.27	46.87	**57.93**	58.15	29.98
SemDistill (Ours)	**69.69**	**53.98**	**81.42**	**78.64**	43.19	**30.88**	**54.17**	**37.70**	**61.20**	56.23	**70.90**	**44.12**
vs. Distillation	*+11.96*	*+28.35*	*+11.29*	*+30.28*	*+7.9*	*+20.51*	*+16.67*	*+26.54*	*+6.73*	*+13.04*	*+9.38*	*+9.56*
Reference: Fully Supervised Upper Bound												
Supervised GT (Doduo)	*92.35*	*82.58*	*89.74*	*88.39*	*49.41*	*35.06*	*60.05*	*32.66*	*69.88*	*62.99*	*73.93*	*45.86*

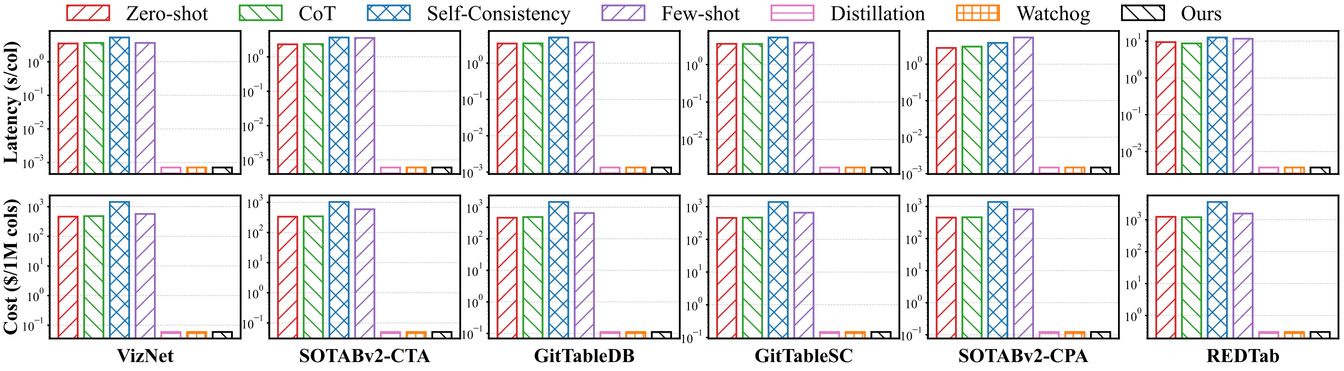


Figure 4: Label Efficiency Trends vs. Trusted Data Size (K).

solely on scarce human annotations is insufficient for covering the vast feature space of wild tables, whereas SemDistill successfully leverages the noisy corpus to boost performance to 30.88%.

Limitations of Frozen LLM Reasoning. Notably, advanced frameworks like Archetype and Chorus often underperform our distilled student. For instance, on GitTablesDB, Archetype achieves only 23.89% Macro-F1 compared to our 30.88%. While *Archetype* improves alignment via re-mapping and *Chorus* mitigates error propagation, both methods rely on the *frozen* parametric knowledge of the LLM. They are strictly inference-time optimizations and cannot overcome fundamental domain gaps. In contrast, SemDistill updates the student's internal weights through closed-loop distillation, effectively internalizing the domain structure and correcting the teacher's inherent biases.

5.3 Efficiency Analysis

We conclude our evaluation by analyzing SemDistill from two critical dimensions for real-world adoption: label efficiency and deployment economics.

Label Efficiency and Scaling Laws. To simulate the data scarcity issue prevalent in data lakes, we evaluate performance under varying budgets of trusted anchors ($K \in \{20, 50, 100\}$). Figure 5 illustrates the Micro-F1 and Macro-F1 trends across six benchmarks, revealing distinct behavioral patterns among the methods. First, we observe a critical saturation effect in *Standard Distillation* (gray dotted line), which exhibits a static, flat trend regardless of K. For instance, on GitTablesDB (Macro-F1), it stagnates at a negligible ≈10%. This confirms that without explicit error correction, the student is fundamentally bounded by the systematic bias of the teacher. Second, the *Trusted-Only* baseline (green dashed line) demonstrates significant instability. While effective on clean data, it struggles with distribution shifts in dirty tables; notably, on GitTablesSC (Macro-F1), its performance paradoxically degrades as K increases from 50 to 100. This suggests that relying solely on sparse supervision risks overfitting to unrepresentative anchors. In sharp contrast, SemDistill (red solid line) consistently achieves the best or near-best performance, particularly in the extreme scarcity setting ($K = 20$). By acting as a knowledge augmentor, it successfully leverages the massive $\mathcal{D}_{noisy}$ for feature coverage while using the scarce $\mathcal{D}_{clean}$ to refine the transition matrix. It follows a positive scaling law: as the trusted budget increases, the noise estimation becomes more precise, allowing the student to continuously improve and widen the gap against the frozen teacher.

Cost and Latency Analysis. To rigorously quantify the operational feasibility for high-throughput enterprise systems, we detail our calculation methodology. For teacher baselines, we utilize GPT-5-mini, priced at \$0.25 and \$2.00 per million input and output tokens, respectively. To normalize expenses across varying table dimensions, we extrapolate the cost to a standardized metric of 1 million columns (1M$_{cols}$). The estimated cost is calculated as:

$$C_{teacher} = \frac{10^6}{N_{cols}} \times \left(\frac{T_{in}}{10^6} \times 0.25 + \frac{T_{out}}{10^6} \times 2.00 \right) \tag{14}$$

where N_{cols} is the number of processed columns, and T_{in}/T_{out} represent the aggregated input/output token counts. Latency is recorded as the cumulative sum of API delays. For student models, including SemDistill, we conduct inference on an NVIDIA RTX 4090 GPU with a batch size of 128. Costs are derived from the commercial GPU rental rate (P_{gpu} = \$0.298/h):

$$C_{student} = \frac{t_{total}}{3600} \times 0.298 \tag{15}$$

where t_{total} denotes the total inference time for one million columns in seconds. Based on this framework, the advantages of SemDistill are substantial. While the teacher model incurs a latency of ~4.17s per column, SemDistill operates at 1.59ms/col via batched inference, reducing latency by 2600× and enabling real-time schema discovery. Financially, relying on external LLM endpoints costs over \$500 per run for 1M$_{cols}$, whereas SemDistill reduces this to a minimal operational cost of approximately \$0.13 (excluding the one-time annotation expense of ~\$5). This represents a staggering cost reduction of 3800×, confirming that our distillation framework effectively democratizes LLM-level intelligence within low-cost, low-latency infrastructures.

5.4 Experimental Analysis

Component-wise Ablation. To verify the contribution of each component in SemDistill, we conduct a component-wise ablation study across four benchmarks. Table 3 summarizes the Micro-F1 results. (1) Structure Modeling: Removing the transition matrix modeling (w/o Fwd Corr.) leads to severe degradation across all datasets. Most notably, on the dirty GitTablesSC, performance drops by 22.55% (54.17 → 31.62), confirming that explicitly modeling $\hat{C}$ is foundational for unmixing systematic bias in complex web tables. (2) Dynamics-based Selection: Replacing AUM with Random Sampling reduces performance significantly, with a 10.14% drop on SOTABv2 (81.42 → 71.28). This indicates that random sampling fails to distinguish between useful hard samples and detrimental noise, whereas AUM effectively targets the latter. (3) CoT Necessity: The role of Chain-of-Thought (CoT) is dataset-dependent. On the cleaner VizNet, removing CoT has a negligible impact (69.69 → 69.26). However, on the dirty GitTablesDB, removing CoT causes a catastrophic collapse to 31.93% (-11.26%), performing even worse than Standard KD. This implies that without the reasoning capabilities of the heterogeneous agent (Qwen3-Max), simple zero-shot re-annotation generates "toxic corrections" that mislead the student.

Table 3: Component-wise Ablation (Micro-F1). Impact of removing Fwd Correction, AUM Selection, and CoT Re-annotation.

Variant	VizNet	SOTAB	Git-DB	Git-SC
SemDistill (Full)	**69.69**	**81.42**	43.19	**54.17**
w/o Fwd Correction	63.20	31.71	26.89	31.62
w/o AUM Selection	66.99	71.28	43.70	44.61
w/o CoT (Zero-shot)	69.26	64.47	31.93	47.06
w/o Re-annotation	69.72	77.71	**45.89**	50.25

Table 4: Selection Strategy Comparison (Micro-F1, Budget=10%). Comparing SemDistill against Random, Entropy, and High-Loss strategies.

Strategy	VizNet	SOTAB	Git-DB	Git-SC
Standard KD	57.73	70.13	35.29	37.50
Random	68.73	78.54	**43.87**	49.75
Entropy	69.03	78.54	32.02	52.06
High-Loss	69.29	78.85	42.35	52.21
SemDistill (AUM)	**69.69**	**81.42**	43.19	**54.17**

Impact of Selection Strategy. Table 4 compares SemDistill against established active learning baselines (Random, Entropy, High-Loss), with the re-annotation budget fixed at 10%. The results reveal nuanced insights into noise handling. First, Re-annotation is Essential: Even simple Random selection yields massive gains over Standard KD (e.g., +11.0% on VizNet), confirming that the Teacher's raw labels contain significant correctable noise. Second, Dynamics vs. Heuristics: On high-quality datasets like SOTAB, SemDistill (81.42) clearly outperforms High-Loss (78.85). This suggests that loss-based methods struggle to distinguish "hard-but-correct" samples from mislabeled ones, whereas AUM successfully isolates systematic hallucinations. On GitTablesSC, SemDistill achieves the highest score of 54.17, surpassing Entropy (52.06). We note that on GitTablesDB, Random selection (43.87) performs competitively with SemDistill (43.19). We attribute this to the extreme noise density in this specific dataset, where the clean anchor assumption for AUM calibration is weakly violated. Nevertheless, SemDistill remains robust and comparable to the best heuristic even in this edge case.

Long-Tail Effect and Bias Correction. A key observation from our main results (Table 2) is that SemDistill achieves a disproportionately higher gain in Macro-F1 compared to Micro-F1. On VizNet, the improvement over the Teacher is +12.16% in Macro-F1 (53.98 vs 41.82), compared to +9.78% in Micro-F1. This implies that our framework is particularly effective for semantic types that are under-represented in the training data.

To verify this, we categorize the semantic types in VizNet into three groups based on frequency: Head (>500), Torso (100-500), and Tail (<100). Figure 6 illustrates the breakdown: (1) Collapse

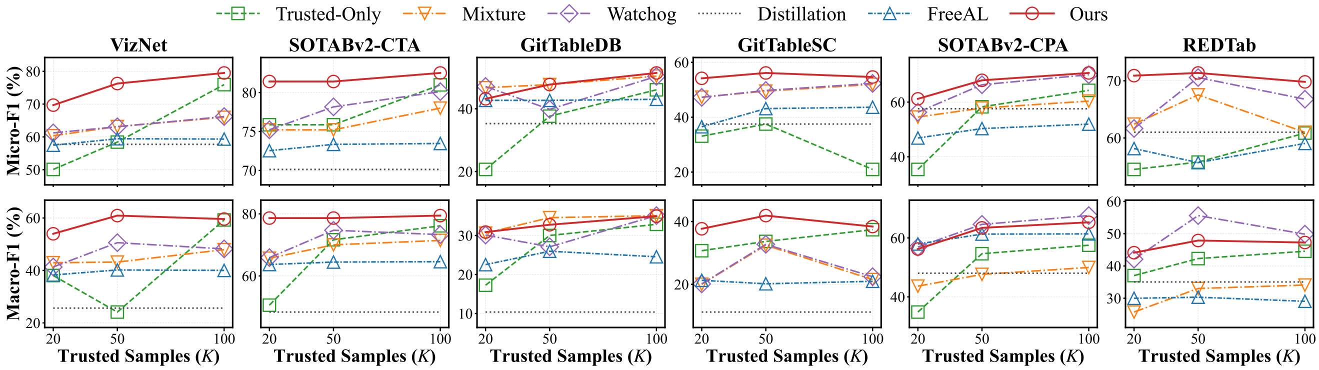


Figure 5: Label Efficiency Trends vs. Trusted Data Size (K). Comparison of Micro-F1/Macro-F1 trends for Gold-Only, Mix, Watchog, Standard KD, FreeAL, and SemDistill across benchmarks.

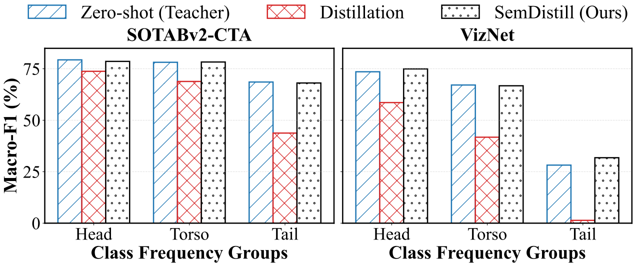


Figure 6: Performance Breakdown by Class Frequency. Comparison of Macro-F1 on Head, Torso, and Tail classes.

of Standard Distillation: A critical finding is that naive distillation (red hatched bar) suffers a catastrophic collapse on tail classes, dropping to near-zero F1 on VizNet. Without explicit correction, the student overfits to the Teacher's frequency bias, completely ignoring rare concepts. (2) Resilience of SemDistill: In contrast, SemDistill (black dotted bar) maintains robust performance across all groups. Notably, on the Tail classes of VizNet, it achieves ≈30% Macro-F1, significantly outperforming both the Teacher (≈25%) and Standard Distillation (≈0%). This confirms that the estimated transition matrix $\hat{C}$ effectively mines rare examples from the noisy stream, solving the supervision bottleneck for long-tail concepts where the Teacher is prone to hallucination.

Qualitative Analysis: Visualizing Correction. To analyze how SemDistill handles systematic hallucinations, we visualize the error transition matrices of the Teacher and Student on VizNet for the top-24 most confusing semantic types (Figure 7). The Teacher's Systematic Bias (Fig. 7a): The teacher exhibits a strong "semantic sink" phenomenon, where fine-grained concepts are systematically absorbed into generic super-classes. This is most visibly evidenced by the dark blue block at the intersection of Ground Truth *operator* and Predicted *name*. It indicates that the LLM almost universally misclassifies *operator* columns as the generic *name* type, failing to capture the specific functional role. Similarly, distinct confusion clusters are observed where *person* is mislabeled as *name*, and *company* leaks into *brand*. The Student's Disambiguation (Fig. 7b):

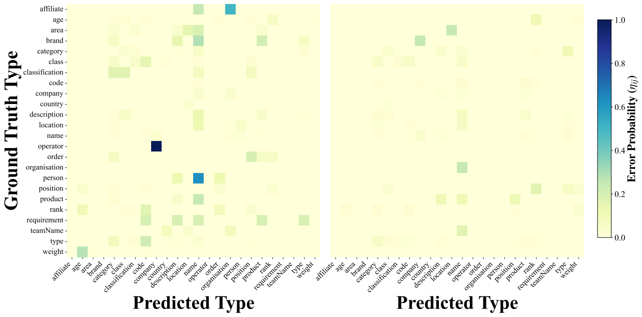


Figure 7: Visualization of Noise Correction. (a) The Teacher shows systematic errors (e.g., *operator* → *name*). (b) The Student effectively suppresses these errors.

In sharp contrast, the Student model effectively "unlearns" these specific noise patterns. The dark off-diagonal blocks are nearly eliminated, resulting in a significantly sparser and diagonal-dominant matrix. Crucially, the Student successfully disambiguates *operator* from *name*, recovering the correct semantic boundary. This confirms that our Forward Correction mechanism ($\hat{C}^T p(x)$) successfully modeled the Teacher's tendency to over-generalize, allowing the Student to leverage the trusted anchors to sharpen its discrimination capability even when the majority of the training data $\mathcal{D}_{noisy}$ pointed to the incorrect generic label.

Parameter Sensitivity. We investigate the impact of the noisy loss weight λ_U and trusted loss weight λ_S on performance, as shown in Figure 8. (1) Noisy Weight λ_U (Risk of Hallucination): Performance generally follows an inverted-U shape. At $\lambda_U = 0$, disregarding the noisy corpus hurts generalization (e.g., dropping to ≈30% on GitTableDB). Performance peaks around $\lambda_U \in [0.1, 0.5]$, confirming the value of noisy supervision. However, excessive weights ($\lambda_U \geq 5.0$) cause overfitting to hallucinations, degrading F1 scores by over 10% on GitTableSC. (2) Trusted Weight λ_S (Correction Strength): A distinct ascending trend is observed. Low λ_S (< 0.1) fails to override noisy signals, while increasing λ_S significantly boosts performance by prioritizing high-quality anchors. Gains saturate around $\lambda_S \geq$

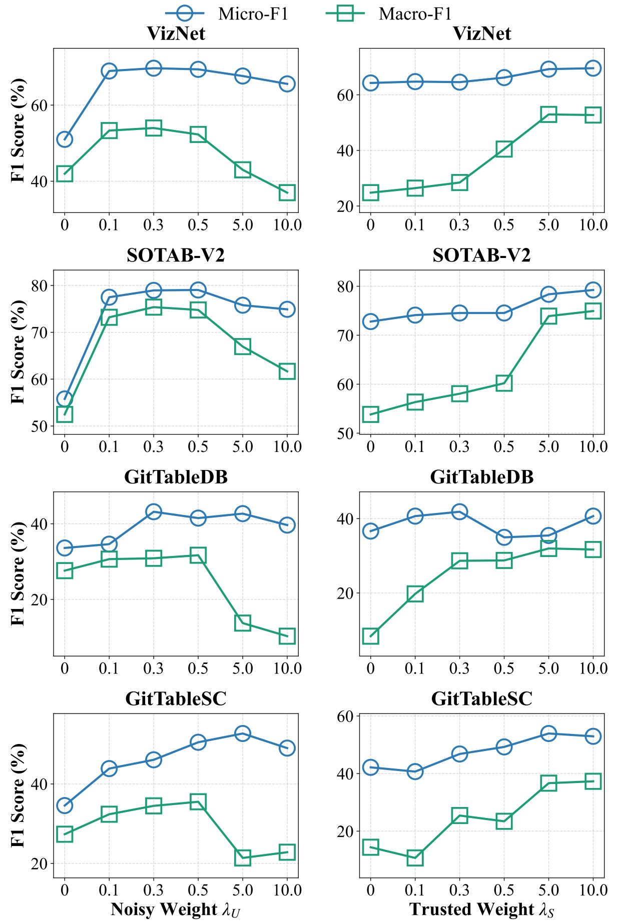


Figure 8: Parameter Sensitivity. Impact of Noisy Weight λ_U (Left) and Trusted Weight λ_S (Right) on Micro/Macro-F1.

5.0, indicating that once the correction signal is sufficiently strong, further weighting yields diminishing returns.

6 Related Work

Traditional Semantic Table Annotation. Traditional approaches for Column Type Annotation (CTA) and Column Property Annotation (CPA) relied on hand-crafted features and statistical matching [14], such as SemanticTyper [29] which utilizes TF-IDF and Kolmogorov-Smirnov tests, and probabilistic models combining multiple statistical features (e.g., Mann-Whitney test) [27]. Inspired by the success of BERT [5] in NLP, recent deep learning methods such as TURL [4], Doduo [33], and TaBERT [45] have adapted Transformer architectures for tabular data. These models typically learn structural representations by pre-training on massive corpora like VizNet [12]. Sato [47] further incorporates topic modeling to enhance semantic detection. To further improve representation quality, recent works have introduced contrastive learning frameworks. Starmie [7] and Watchog [24] leverage contrastive pre-training to capture semantic similarities and address class imbalance. Additionally, RECA [34] addresses the complexity of wide tables by aligning inter-table contexts. While effective, these supervised models require large-scale, high-quality labeled datasets.

LLM-based Semantic Table Annotation. Recently, LLMs have demonstrated advanced zero-shot capabilities, significantly enhancing STA through various methods. One direction involves fine-tuning generalist models like TableLlama [49] and Table-GPT [22] on comprehensive table corpora. Alternatively, unified frameworks harness foundation models without parameter updates. Specifically, Chorus [15] targets data discovery tasks including CTA, while ArcheType [9] focuses on zero-shot inference optimization through strategies such as context serialization and label remapping. Addressing the noise in wide tables, Ding et al. [6] propose a retrieve-and-verify framework to select informative column contexts. To enhance reasoning capabilities, Chain-of-Table [38] guides LLMs through iterative steps, while Korini et al. [17] improve generalization in Column Property Annotation via joint fine-tuning strategies. Addressing specific limitations, RACOON [40] employs retrieval-augmented generation to mitigate hallucinations, and Li et al. [19] generate labeling functions to enable low-cost weak supervision. However, since direct deployment faces latency and privacy constraints, we adopt a label-efficient distillation paradigm inspired by West et al. [41] to train a compact student model, explicitly mitigating the risk of overfitting to the teacher’s structural flaws.

Weak Supervision and Distillation. Weak supervision (WS) has emerged as a standard approach to address the labeling bottleneck in data labeling. Classic frameworks like Snorkel [30] and HoloClean [31] generate training data from heuristic rules. However, relying on hand-crafted rules is brittle and often lacks the semantic knowledge required for fine-grained entity typing. Recently, leveraging LLMs as noisy annotators has emerged as a promising alternative [35]. General distillation frameworks, such as FreeAL [44] and CanDist [42], employ collaborative filtering or self-training to refine LLM pseudo-labels. While effective for unstructured text, applying these generic distillation methods to STA is suboptimal. First, they typically treat data as independent text segments, ignoring the vertical schema constraints inherent in relational tables. Second, they often rely on uncertainty-based filtering (similar to DivideMix [20]), which fails to detect systematic hallucinations in tabular tasks where LLMs are confident but consistently wrong (e.g., frequency bias). SemDistill addresses these domain-specific challenges by utilizing Training Dynamics (AUM) to explicitly calibrate structural errors using a small set of trusted data.

7 Conclusion

In this work, we proposed SemDistill, a label-efficient framework that bootstraps high-performance local annotators from noisy LLM supervision to address the cost, latency, and privacy bottlenecks of semantic table annotation. By identifying and rectifying the unique compound noise of LLMs, specifically systematic ontological drift and confident hallucinations, our approach effectively decouples the student from the teacher’s errors. Extensive experiments confirm that SemDistill achieves a Pareto-optimal solution, significantly outperforming both LLM teacher and SLM distillation methods while reducing inference costs by 3,800 times. This advancement democratizes LLM-level intelligence for large-scale enterprise data governance.

References

[1] Qi An, Chihua Ying, Yuqing Zhu, Yihao Xu, Manwei Zhang, and Jianmin Wang. 2025. LEDD: large language model-empowered data discovery in data lakes. *arXiv preprint arXiv:2502.15182* (2025).

[2] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *ICML*.

[3] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261* (2025).

[4] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record* 51, 1 (2022), 33–40.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

[6] Zhihao Ding, Yongkang Sun, and Jieming Shi. 2025. Retrieve-and-verify: A table context selection framework for accurate column annotations. *Proceedings of the ACM on Management of Data* 3, 6 (2025), 1–27.

[7] Grace Fan, Jin Wang, Yuliang Li, Dan Zhang, and Renée Miller. 2022. Semantics-aware dataset discovery from data lakes with contextualized column-based representation learning. *arXiv preprint arXiv:2210.01922* (2022).

[8] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *ICDE*.

[9] Benjamin Feuer, Yurong Liu, Chinmay Hegde, and Juliana Freire. 2023. Archetype: A novel framework for open-source column type annotation using large language models. *arXiv preprint arXiv:2310.18208* (2023).

[10] Juliana Freire, Grace Fan, Benjamin Feuer, Christos Koutras, Yurong Liu, Eduardo Peña, Aécio SR Santos, Cláudio T Silva, and Eden Wu. 2025. Large Language Models for Data Discovery and Integration: Challenges and Opportunities. *IEEE Data Eng. Bull.* 49, 1 (2025), 3–31.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[12] Kevin Hu, Snehalkumar'Neil'S Gaikwad, Madelon Hulsebos, Michiel A Bakker, Emanuel Zgraggen, César Hidalgo, Tim Kraska, Guoliang Li, Arvind Satyanarayan, and Çağatay Demiralp. 2019. VizNet: Towards a large-scale visualization learning and benchmarking repository. In *CHI*.

[13] Madelon Hulsebos, Çagatay Demiralp, and Paul Groth. 2023. Gittables: A large-scale corpus of relational tables. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–17.

[14] Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. 2019. Sherlock: A deep learning approach to semantic data type detection. In *SIGKDD*.

[15] Moe Kayali, Anton Lykov, Ilias Fountalis, Nikolaos Vasiloglou, Dan Olteanu, and Dan Suciu. 2023. CHORUS: foundation models for unified data discovery and exploration. *arXiv preprint arXiv:2306.09610* (2023).

[16] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Keti Korini and Christian Bizer. 2024. Column property annotation using large language models. In *ESWC*.

[18] Keti Korini, Ralph Peeters, and Christian Bizer. 2022. SOTAB: The WDC Schema.org table annotation benchmark. In *CEUR Workshop Proceedings*, Vol. 3320. RWTH Aachen, 14–19.

[19] Chenjie Li, Dan Zhang, and Jin Wang. 2024. Llm-assisted labeling function generation for semantic type detection. *arXiv preprint arXiv:2408.16173* (2024).

[20] Junnan Li, Richard Socher, and Steven CH Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394* (2020).

[21] Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2024. Table-gpt: Table fine-tuned gpt for diverse table tasks. *Proceedings of the ACM on Management of Data* 2, 3 (2024), 1–28.

[22] Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2024. Table-gpt: Table fine-tuned gpt for diverse table tasks. *Proceedings of the ACM on Management of Data* 2, 3 (2024), 1–28.

[23] Chuangtao Ma, Sriom Chakrabarti, Arijit Khan, and Bálint Molnár. 2025. Knowledge graph-based retrieval-augmented generation for schema matching. *arXiv preprint arXiv:2501.08686* (2025).

[24] Zhengjie Miao and Jin Wang. 2023. Watchog: A light-weight contrastive learning based framework for column annotation. *Proceedings of the ACM on Management of Data* 1, 4 (2023), 1–24.

[25] OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).

[26] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*.

[27] Minh Pham, Suresh Alse, Craig A Knoblock, and Pedro Szekely. 2016. Semantic labeling: a domain-independent approach. In *ISWC*.

[28] Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q. Weinberger. 2020. Identifying Mislabeled Data using the Area Under the Margin Ranking. In *NeurIPS*.

[29] S Krishnamurthy Ramnandan, Amol Mittal, Craig A Knoblock, and Pedro Szekely. 2015. Assigning semantic labels to data sources. In *ESWC*.

[30] Alexander J. Ratner, Stephen H. Bach, Henry Ehrenberg, Jason A. Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid Training Data Creation with Weak Supervision. *Proc. VLDB Endow.* 11, 3 (2017), 269–282.

[31] Theodoros Rekatsinas, Xu Chu, Ihab F Ilyas, and Christopher Ré. 2017. Holoclean: Holistic data repairs with probabilistic inference. *arXiv preprint arXiv:1702.00820* (2017).

[32] Siffi Singh, Alham Fikri Aji, Gaurav Singh Tomar, and Christos Christodoulopoulos. 2022. A relation extraction dataset for knowledge extraction from web tables. In *COLING*.

[33] Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating columns with pre-trained language models. In *SIGMOD*.

[34] Yushi Sun, Hao Xin, and Lei Chen. 2023. Reca: Related tables enhanced column semantic type annotation framework. *Proceedings of the VLDB Endowment* 16, 6 (2023), 1319–1331.

[35] Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. *arXiv preprint arXiv:2402.13446* (2024).

[36] Jianhong Tu, Ju Fan, Nan Tang, Peng Wang, Guoliang Li, Xiaoyong Du, Xiaofeng Jia, and Song Gao. 2023. Unicorn: A unified multi-tasking model for supporting matching tasks in data integration. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–26.

[37] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* (2022).

[38] Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. 2024. Chain-of-table: Evolving tables in the reasoning chain for table understanding. *arXiv preprint arXiv:2401.04398* (2024).

[39] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[40] Lindsey Linxi Wei, Guorui Xiao, and Magdalena Balazinska. 2024. RACOON: An LLM-based Framework for Retrieval-Augmented Column Type Annotation with a Knowledge Graph. In *NeurIPS*.

[41] Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *NAACL-HLT*.

[42] Mingxuan Xia, Haobo Wang, Yixuan Li, Zewei Yu, Jindong Wang, Junbo Zhao, and Runze Wu. 2025. Prompt Candidates, then Distill: A Teacher-Student Framework for LLM-driven Data Annotation. *arXiv preprint arXiv:2506.03857* (2025).

[43] Guorui Xiao, Dong He, Jin Wang, and Magdalena Balazinska. 2025. CENTS: A Flexible and Cost-Effective Framework for LLM-Based Table Understanding. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4574–4587.

[44] Ruixuan Xiao, Yiwen Dong, Junbo Zhao, Runze Wu, Minmin Lin, Gang Chen, and Haobo Wang. 2023. Freeal: Towards human-free active learning in the era of large language models. *arXiv preprint arXiv:2311.15614* (2023).

[45] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314* (2020).

[46] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* (2016).

[47] Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çağatay Demiralp, and Wang-Chiew Tan. 2019. Sato: Contextual semantic type detection in tables. *arXiv preprint arXiv:1911.06311* (2019).

[48] Haochen Zhang, Yuyang Dong, Chuan Xiao, and Masafumi Oyamada. 2024. Jellyfish: Instruction-tuning local large language models for data preprocessing. In *EMNLP*.

[49] Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024. Tablellama: Towards open large generalist models for tables. In *NAACL HLT*.