LLMs as Stratification Signals for KG Accuracy Evaluation

Stefano Marchesin
University of Padua
Padua, Italy
stefano.marchesin@unipd.it

Matteo Ceccarello
University of Padua
Padua, Italy
matteo.ceccarello@unipd.it

Gianmaria Silvello
University of Padua
Padua, Italy
gianmaria.silvello@unipd.it

ABSTRACT

Knowledge Graph (KG) accuracy assessment is essential for ensuring data quality in downstream applications, yet remains prohibitively expensive due to annotation costs and scale. Large Language Models (LLMs), trained on vast corpora, offer cheap fact validation but remain unreliable as direct accuracy estimators due to hallucinations and knowledge gaps. We propose a novel approach that exploits LLM capabilities without relying on their correctness: using aggregated LLM predictions as stratification signals for sampling-based accuracy estimation. By partitioning KGs into internally homogeneous strata guided by aggregated LLM outputs, we achieve statistically significant cost reductions ranging from 11% to 54% over unstratified and topology-based baselines on real-world KGs. To scale beyond LLM computational constraints, we introduce a knowledge distillation strategy that transfers stratification signals to efficient student models, requiring annotation of only 0.25% of facts while maintaining signal quality. Experiments on six KGs spanning 20M+ triples demonstrate consistent improvements over SotA methods, with statistical guarantees on accuracy estimates.

PVLDB Reference Format:
Stefano Marchesin, Matteo Ceccarello, and Gianmaria Silvello. LLMs as Stratification Signals for KG Accuracy Evaluation. PVLDB, 19(1): XXX-XXX, 2026.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/kg-quality/llm4kg-accuracy.

1 INTRODUCTION

Large-scale Knowledge Graphs (KGs), such as Wikidata [46], DBpedia [2], YAGO [20], and NELL [33], have emerged as core infrastructural components underpinning a wide range of academic and industrial systems [25]. These systems span diverse application domains, including digital assistants and web search [24, 34], as well as recommender systems and question answering platforms [40, 41], all of which depend on the accuracy and integrity of the underlying KGs. Nonetheless, the systematic assessment and assurance of KG quality remain comparatively underexplored in both research and operational settings. This gap is particularly critical because inaccuracies in KGs directly degrade user experience and compromise system reliability, as evidenced by systems such as Saga [25], which explicitly emphasize on-demand KG accuracy evaluation as a prerequisite for delivering trustworthy results.

KGs accuracy and reliability are undermined by incorrect assertions, logical inconsistencies, and sparse data [14, 39], which increase misinformation risk [31]. These problems also degrade KG embeddings – the core of modern recommendation and search systems [24, 34] – which are highly noise-sensitive and perform poorly when trained on noisy data [39]. Because these quality issues undermine data integrity and application performance, systematic KG accuracy assessment is vital for informed deployment decisions in direct and downstream use cases.

Assessing KG accuracy requires assigning correctness labels to individual assertions, a task constrained by two fundamental challenges: annotation cost and scale. Manual labeling is prohibitively expensive [35], making exhaustive evaluation infeasible for real-world KGs containing hundreds of millions to billions of facts [17]. Recent work addresses these constraints by reformulating accuracy assessment as a constrained optimization problem [17, 28, 29]: minimize annotation costs while maintaining statistical guarantees on estimated accuracy. State of the Art (SotA) methods employ iterative procedures combining sampling strategies, point estimators, and $(1-\alpha)$ intervals to quantify uncertainty. Sampling design is the critical component determining both efficiency and reliability of accuracy estimation. Current approaches primarily leverage simple random sampling and *cluster sampling* with promising results [17].

While a third strategy based on stratified sampling has been identified as theoretically superior for reducing annotation costs, existing methods do not employ it due to the challenge of constructing effective strata [17, 28]. Stratified sampling divides the KG into strata using informative signals to ensure consistency within each partition. When stratification aligns with the feature of interest, it yields reliable estimates from smaller samples than unstratified sampling [10], reducing annotation costs. However, simple topology-based signals produce mixed, inconclusive results [17, 28], showing the need for more informative signals that approach theoretical cost bounds. The key challenge is finding stratification signals that capture the latent feature of interest – i.e., KG accuracy – which is unknown a priori, thus making this task inherently difficult.

This work addresses this gap by proposing the use of Large Language Models (LLMs) to generate stratification signals for KG accuracy evaluation. LLMs have achieved near-human performance across several tasks and offer practical advantages for fact validation: they can process textual evidence, identify logical inconsistencies between claims and supporting documents, and access factual knowledge encoded in model parameters [37, 48, 49]. Hence, by automating fact validation at scale, LLMs could enable cost-effective evaluation of large-scale KGs. Yet LLMs suffer from well-documented limitations: they generate hallucinations and unfaithful responses [22], and exhibit systematic biases and knowledge

Proceedings of the VLDB Endowment, Vol. 19, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

gaps that undermine their reliability for fact validation [43]. In this paper, we substantiate these limitations, showing that suboptimal performance in fact verification leads to unreliable KG accuracy estimates when derived directly from LLMs. To mitigate this issue, we introduce a novel methodology: instead of tasking LLMs with directly determining KG accuracy, we employ their outputs as stratification signals within a sampling-based accuracy estimation framework. This shift in role is central to our approach. To the best of our knowledge, LLMs have not previously been employed to support accuracy estimation in this specific way. By assigning LLMs to the task of signal generation, while reserving ground-truth annotation exclusively for human annotators, we exploit the capacity of LLMs to identify patterns across assertions without propagating their systematic errors into the final estimates. This functional separation enables us to leverage the advantages of LLMs while maintaining the unbiasedness and statistical validity of the resulting accuracy estimators.

Contributions. The main contributions of this work are:

- An empirical analysis of the use of LLMs to assess KG accuracy. We show that LLM-derived KG accuracy estimates can deviate by up to 45% from true accuracy values. This substantial error confirms that direct use of LLMs as annotators is inadequate for reliable KG evaluation, motivating a novel integration of LLMs into the evaluation pipeline.
- We propose using aggregated LLM predictions as stratification signals, achieving up to 50% cost reduction over topology-based and unstratified baselines on real-world KGs. By combining multiple LLM outputs through soft majority voting, our approach mitigates individual model hallucinations while producing internally homogeneous strata that enable efficient sampling-based accuracy estimation.
- We introduce a signal distillation approach that enables stratified evaluation of million- to billion-scale KGs by training efficient student models on only 0.25% of triples annotated by LLMs. To date, this is the first use of distilled models to approximate LLM-based signals for large-scale statistical estimation. Experiments on 20M-triple KGs demonstrate that distilled models achieve performance on par with their LLM teachers at a fraction of the cost.

Outline. Section 2 introduces the evaluation framework. Section 3 describes sampling and estimation techniques. Section 4 introduces the LLMs as stratification signals for KG accuracy evaluation. Section 5 presents the signal distillation strategy. Sections 6 and 7 detail the experimental setup and results. Section 8 reviews related work, and Section 9 concludes and outlines future work.

2 BACKGROUND

Preliminaries. Following Bonifati et al. [4], we model KGs as grounded RDF graphs $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \eta)$, where $\mathcal{V} = \{\mathcal{E} \cup \mathcal{L}\}$ is the set of nodes (entities $\mathcal{E}$ and literals $\mathcal{L}$), $\mathcal{R}$ the set of relationships, and $\eta : \mathcal{R} \to \mathcal{E} \times \{\mathcal{E} \cup \mathcal{L}\}$ maps each relationship to an ordered node pair. This induces the ternary relation $\mathcal{T}$ of triples (s, p, o) with $s \in \mathcal{E}$, $p \in \mathcal{R}$, and $o \in \{\mathcal{E} \cup \mathcal{L}\}$; denoting the total number of triples as $M = |\mathcal{T}|$. Given $\mathcal{T}$, the **entity cluster** of $e \in \mathcal{E}$ is $\mathcal{G}[e] = \{(s, p, o) \in \mathcal{T} \mid s = e\}$. The set of all entity clusters forms the KG cluster family $\mathcal{C}$.

Treating triples as first-class citizens of KGs, we redefine $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{T}, \eta)$ and focus solely on A-Box triples (extensional), ignoring T-Box triples (intensional). Thus, each triple represents a factual assertion, and we use the terms triple and fact interchangeably.

KG Accuracy Evaluation. To evaluate KG accuracy, we define the correctness of a triple $t \in \mathcal{T}$ via an indicator function $\mathbb{1}(t)$, where $\mathbb{1}(t) = 1$ if t is correct and 0 otherwise. The KG accuracy is then the proportion of correct triples, τ, in $\mathcal{G}$:

$$\mu = \frac{\sum_{t \in \mathcal{T}} \mathbb{1}(t)}{M} = \frac{\tau}{M} \tag{1}$$

where $\mathbb{1}(t)$ is obtained through manual or automatic annotation.

Given its importance, KG accuracy is mainly evaluated via manual annotation, which provides reliable ground truth but is costly and time-consuming. Therefore, accuracy is usually estimated from a smaller, representative sample drawn with an unbiased sampling strategy $\mathcal{S}$. This enables computing an unbiased estimator $\hat{\mu}$ with $E[\hat{\mu}] = \mu$ over the sample $\mathcal{T}_\mathcal{S} \subset \mathcal{T}$, along with its $(1 - \alpha)$ interval to quantify sampling uncertainty.

Building on this, prior work [17, 29] formulates KG accuracy evaluation as a constrained minimization problem. Let $\text{cost}(\mathcal{T}_\mathcal{S})$ denote the manual validation cost of the sample, and let Margin of Error (MoE) be half the width of the $1 - \alpha$ interval.

Problem. KG accuracy evaluation:

$$\begin{aligned} \underset{\mathcal{S}}{\text{minimize}} \quad & \text{cost}(\mathcal{T}_\mathcal{S}) \\ \text{subject to} \quad & E[\hat{\mu}] = \mu, \text{MoE} \leq \varepsilon \end{aligned}$$

Here, ε sets the upper bound on the MoE and acts as the stopping criterion. As the sample size $n_\mathcal{S} = |\mathcal{T}_\mathcal{S}|$ increases, the interval width (MoE) decreases, reaching zero when $n_\mathcal{S} \approx M$. However, larger samples increase $\text{cost}(\mathcal{T}_\mathcal{S})$ due to manual validation. The objective is thus to identify the smallest sample such that $\text{MoE} \leq \varepsilon$.

Evaluation Framework. The minimization problem is solved with an iterative four-step framework. In the **first step**, a small batch of triples is sampled from the KG using a strategy $\mathcal{S}$. In the **second step**, these triples are manually annotated and merged with previous annotations. In the **third step**, the accumulated sample is used to compute the unbiased estimate $\hat{\mu}$ and its $1 - \alpha$ interval. In the **fourth step**, the stopping condition $\text{MoE} \leq \varepsilon$ is checked. If satisfied, the process returns $\hat{\mu}$ and its interval; otherwise, it restarts from the first step. This iterative design limits oversampling and annotation cost while ensuring accurate estimates.

Within this framework, the sampling strategy is central, as the sample $\mathcal{T}_\mathcal{S}$ directly affects the annotation cost. Our work therefore focuses on improving sampling effectiveness. We review current SotA sampling and estimation methods [17, 29] and show how stratified sampling can further reduce costs.

Gao et al. [17] showed that ideal (oracle) stratification can drastically reduce costs over existing SotA strategies. However, they found that simple topology-based stratification yields mixed, inconclusive results, highlighting the need for more informative signals to approach ideal costs.

3 SAMPLING AND ESTIMATION: SOTA

Current KG accuracy evaluation methods move beyond simple random sampling to more cost-effective cluster-based strategies [17, 28]. Recently, confidence intervals have also been replaced with credible intervals to provide a more efficient and interpretable quantification of sampling uncertainty [29]. We first outline cluster-based sampling – the current SotA baseline – and then introduce credible intervals, which we use to ensure convergence.

Cluster Sampling. Among cluster-based methods [10], Two-stage Weighted Cluster Sampling (TWCS) is the most cost-effective unbiased approach [17, 28]. It works in two stages: (i) sample n_C clusters with probability proportional to size, $\pi_i = M_i/M$, where $M_i = |\mathcal{G}[e_i]|$; (ii) from each selected cluster, sample $\min\{M_i, m\}$ triples uniformly without replacement. Thus, TWCS combines weighted cluster sampling with second-stage SRS.

Point Estimation. Let $\hat{\mu}_i$ denote the estimated accuracy of the i-th entity cluster, computed as the sample proportion:

$$\hat{\mu}_i = \frac{\sum_{j=1}^{\min\{M_i,m\}} \mathbb{1}(t_j)}{\min\{M_i, m\}} \tag{2}$$

which is known to be an unbiased estimator [10]. When TWCS is applied with a second-stage sample of approximate size m, the unbiased estimator of the overall KG accuracy μ is given by [10]:

$$\hat{\mu}_{\text{TWCS}} = \frac{\sum_{i=1}^{n_C} \hat{\mu}_i}{n_C} \tag{3}$$

Interval Estimation. To quantify uncertainty, we adopt Bayesian $1-\alpha$ Credible Intervals (CrIs) [16]. Modeling annotation as $\text{Bin}(n_{\mathcal{S}}, \mu)$ and assuming a conjugate $\text{Beta}(a, b)$ prior [3], observing $\tau_{\mathcal{S}}$ correct triples yields the posterior distribution $\mu|\mathcal{T}_{\mathcal{S}} \sim \text{Beta}(a+\tau_{\mathcal{S}}, b+n_{\mathcal{S}}-\tau_{\mathcal{S}})$. In this framework, a $1-\alpha$ CrI is any interval (l, u) satisfying:

$$\Pr(l \le \mu \le u|\mathcal{T}_{\mathcal{S}}) = F(u|\mathcal{T}_{\mathcal{S}}) - F(l|\mathcal{T}_{\mathcal{S}}) = 1-\alpha \tag{4}$$

where $F(\cdot|\mathcal{T}_{\mathcal{S}})$ is the posterior Cumulative Distribution Function (CDF). Among such intervals, the Highest Posterior Density (HPD) interval is shortest and contains the highest posterior mass [32, 38]. HPD CrIs thus provide direct probabilistic interpretation, express precision via width, and capture the most credible parameter range, making them well suited for KG accuracy evaluation [29].

4 STRATIFIED KG ACCURACY EVALUATION

Stratified sampling is a variance reduction technique [8, 10] that can further improve the efficiency of KG accuracy evaluation. It is effective when the stratification signal groups items into subpopulations with relatively homogeneous elements, enabling variance reduction. This reduction directly controls how quickly $1-\alpha$ intervals shrink – since interval width depends on variance – thereby speeding convergence of the optimization problem.

We next define an effective stratification model and explain the limitations of existing approaches. We then present our solution, detailing the LLM-driven stratification procedure, stratified sampling strategy, and estimator, followed by an end-to-end overview of the complete LLM-based stratified KG accuracy evaluation framework.

4.1 Stratification Model

Stratified sampling partitions the population into non-overlapping strata based on features of interest, ensuring their representation. When strata are internally homogeneous, precise estimates can be derived from small samples within each stratum [10]. Poorly aligned stratification signals, however, increase variance. Thus, the effectiveness of stratification depends on how well the stratification signal aligns with the target feature.

In KG accuracy evaluation, entity clusters are the natural analysis unit, since auditing triples within the same cluster $\mathcal{G}[e]$ is cheaper than auditing across entities [17, 30]. We therefore stratify at the cluster level, but cluster-level accuracy is unknown a priori, so we need a proxy correlated with this latent feature. An ideal stratification model yields strata with distinct expected accuracies and low within-stratum variance, yet this is hard to design because factual accuracy is complex and context-dependent [51]. To this end, Gao et al. [17] proposed using topological properties (e.g., cluster size) as stratification signals. In our experiments, we show the limitations of this approach, attributing weak correlations to the limited link between KG topology and semantic correctness [21, 52]. To address this gap, we introduce a semantics-aware approach that leverages LLM-based correctness estimates for KG triples to derive proxy measures of cluster-level accuracy.

4.2 LLMs as Stratification Signals

To use LLMs as stratification signals, we obtain LLM predictions for triples and aggregate them to estimate cluster accuracies as proxies for true accuracy, defining a prompt as a function of the triple.

Definition 1 (LLM prompt). Given a triple t, an LLM prompt is a function $\text{prompt}(t)$ that maps t to a natural language template suitable for LLM input to assess fact correctness.

We employ *few-shot prompting* with six KG-independent examples – three per label class – as this approach more effectively preserves LLM factuality and mitigates hallucinations compared to zero-shot or in-domain methods [43]. By KG-independent, we mean examples that neither derive from nor overlap with the target KG's specific facts, but still fit its general theme. The same example content can be reused across KGs, with its representation adapted each time to the relevant schema and predicate conventions. We design prompts to use only LLM parametric knowledge, without retrieval. This (1) ensures consistent triple evaluation despite variable retrieval quality across contents and KG domains, and (2) isolates LLM knowledge limits, clarifying whether poor stratification comes from knowledge gaps or retrieval failures. For stratification, this suffices: clusters only need consistent accuracy rankings, not perfect validation. Parametric-only responses yield more reliable consistency than retrieval-based methods, which can introduce biases from retrieval success rather than true accuracy. Since triple correctness is framed as binary classification (cf. Section 2), we design prompts that elicit binary LLM responses, effectively treating the LLM as an automated indicator function.[1]

Definition 2 (LLM indicator function). Given a triple t and its corresponding $\text{prompt}(t)$, the LLM is defined as an indicator

[1]The template prompts and corresponding few-shot examples are provided in the online repository for space reasons.

function LLM(prompt(t)) that outputs 1 if the triple is deemed correct and 0 otherwise.

This lets us redefine cluster-level accuracy using LLM instead of human annotations by replacing the indicator $\mathbb{1}(\cdot)$ with LLM$(\cdot)$.

Definition 3 (LLM-derived cluster accuracy). Given an entity cluster $\mathcal{G}[e]$ with M_e triples, the LLM-derived cluster accuracy is:

$$\hat{\mu}_{e|\text{LLM}} = \frac{\sum_{t \in \mathcal{G}[e]} \text{LLM}(\text{prompt}(t))}{M_e} \tag{5}$$

This provides a stratification signal for partitioning the KG. The closer LLM-derived predictions match true triple correctness, the better this signal approximates cluster-level accuracy, yielding more homogeneous strata and more cost-effective evaluation.

Individual LLMs differ in factual accuracy, reasoning, and hallucination rates, adding noise to their predictions. To reduce these model-specific biases, we use an ensemble aggregation strategy instead of single-model judgments. Among crowdsourcing and ensemble-learning methods, soft majority voting offers a principled balance between noise reduction and uncertainty preservation [1, 23]. Unlike hard voting schemes that discard minority opinions, soft voting retains information about inter-model disagreement – a signal itself informative for stratification. Values near 0.5 reflect genuine disagreement and indicate triples that lack consensus, while values near 0 or 1 reflect strong agreement.

Definition 4 (LLM-aggregated triple score). For a triple t and a set of models $\mathcal{A} = \{\text{LLM}_1, \ldots, \text{LLM}_k\}$, the aggregated score is:

$$\text{agg}(t) = \frac{\sum_{i=1}^{k} \text{LLM}_i(\text{prompt}(t))}{k} \tag{6}$$

where $\text{LLM}_i(\text{prompt}(t)) \in \{0, 1\}$ is the prediction of model LLM_i.

We see that for each triple, we aggregate binary predictions from all LLMs using soft majority voting: each model contributes a binary prediction (1 for correct, 0 for incorrect), and the aggregated score is the mean of these votes. Note that this ensemble aggregation strategy is orthogonal to the specific choice of LLMs. That is, individual models in the ensemble can be replaced or upgraded as better LLMs become available, without requiring changes to the underlying methodology. As a result, advances in LLMs quality directly translate into more reliable aggregate signals.

Definition 5 (LLM-aggregated cluster score). For an entity cluster $\mathcal{G}[e]$ with M_e triples, the LLM-aggregated cluster score is the mean of its triple-level LLM-aggregated scores:

$$\hat{\mu}_{e|\text{agg}} = \frac{\sum_{t \in \mathcal{G}[e]} \text{agg}(t)}{M_e} \tag{7}$$

This formulation preserves signals from inter-model disagreement, improving robustness on boundary cases and providing more informative guidance under prediction uncertainty.

4.3 Stratification and Sampling Procedures

Stratification partitions the KG into non-overlapping subsets of entity clusters. Specifically, it divides the cluster population C into Q strata based on informative signals, aiming to reduce estimator variance by decreasing intra-stratum heterogeneity while preserving inter-stratum differences [10]. In our setting, where stratification signal is a proxy for true cluster accuracy, variance reduction increases with the strength of its alignment with the true accuracy.

To implement this, we use the Cumulative Square Root of Frequency (CSRF) method [12], adopted in prior work for its strong theoretical foundation [17, 28], leveraging LLM-aggregated cluster scores as the guiding stratification signal. CSRF partitions the proxy scores according to cumulative square-root frequencies, forming strata that adapt to the empirical distribution of the signal: dense score regions are subdivided more finely, while sparse regions are grouped more coarsely. This process groups clusters with similar proxy scores and separates those with different ones. When the proxy correlates well with true cluster accuracy, this alignment reduces intra-stratum variance in the target quantity, thereby decreasing the variance of the stratified estimator – and, in turn, the MoE. Given the cluster scores, CSRF attempts to create exactly Q strata, returning fewer if required by the data distribution.

Proxy signals can create degenerate strata with (near) zero variance when multiple clusters receive almost identical scores. This over-stratification reflects misalignment between the proxy and the latent target (true cluster accuracy): separating such strata does not reduce intra-stratum variance, as they mostly capture noise and yield small, unstable partitions. We address this with an adaptive merging step that folds zero-variance strata into their nearest neighbors (by signal proximity), preserving meaningful signal structure while avoiding unnecessary fragmentation and producing strata that better match the underlying target distribution.

We empirically validated the LLM-based stratification and merging through an ablation study and variance analysis, confirming that both reduce variance and annotation costs. Detailed results are available in the online repository.

We use **stratified cluster sampling** to draw annotation batches from the Q strata. We adopt proportional allocation [10], drawing clusters from each stratum in proportion to its share of KG triples (the sampling units). Within each stratum q, we then apply a Stratified TWCS (STWCS) design: TWCS with second-stage sample size m. Proportional allocation gives all KG triples equal selection probability, so STWCS is an unbiased Equal Probability of Selection Method (EPSEM) design [10]. In other words, STWCS ensures that the collected samples are representative of the underlying population, thereby preserving the KG label distribution.

Finally, let $\mathcal{E}_q$ denote the set of N_q subject entities in stratum q, $C_q = \{\mathcal{G}[e] \mid e \in \mathcal{E}_q\}$ the corresponding family of entity clusters, and $M_q = \sum_{i=1}^{N_q} M_i$ the total number of triples contained in the stratum. Defining the stratum weight as $W_q = M_q/M$, the unbiased estimator of the KG accuracy μ under STWCS is given by [10]:

$$\hat{\mu}_{\text{STWCS}} = \sum_{q=1}^{Q} W_q \hat{\mu}_q \tag{8}$$

where $\hat{\mu}_q$ is the unbiased accuracy estimate for stratum q, obtained under TWCS (cf. Equation (3)).

4.4 Stratified Evaluation Framework

Figure 1 illustrates how the evaluation framework from Section 2 is instantiated through LLM-guided stratified sampling. The pipeline

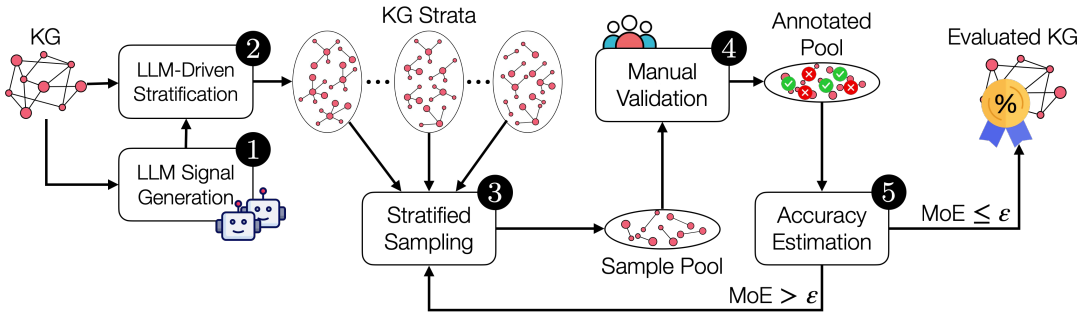

Figure 1: LLM-based stratified KG accuracy evaluation framework.

comprises five phases driven by LLM-derived signals. In phase ❶, LLM-aggregated triple scores are computed and aggregated at the cluster level, yielding proxy signals for true cluster accuracy. In phase ❷, these signals guide the stratification step, partitioning KG clusters into strata that reflect the structure induced by the LLM scores. In phase ❸, a batch of triples is sampled from the resulting strata via STWCS. In phase ❹, manual annotations are obtained for sampled triples and merged with existing annotations. Finally, in phase ❺, an unbiased point estimator computes $\hat{\mu}$, the KG accuracy estimate, and an interval estimator builds a $1-\alpha$ HPD CrI to quantify sampling uncertainty. If MoE $\leq \varepsilon$, the process terminates and reports the accuracy estimate with its $1-\alpha$ CrI; otherwise, it loops back to ❸, progressively increasing the estimation precision until the target condition (MoE $\leq \varepsilon$) is satisfied.

The framework can be expressed as a unified algorithm that combines all components needed for efficient, reliable KG accuracy estimation. Algorithm 1 takes as input the KG triples $\mathcal{T}$, a set of LLM models $\mathcal{A}$, the number of strata Q, the significance level α, the MoE threshold ε, and the second-stage sample size m. It outputs the estimated KG accuracy $\hat{\mu}$ and its $1-\alpha$ HPD CrI (l, u). The algorithm is organized around the five phases of the LLM-based stratified evaluation framework (cf. Figure 1). The first two phases focus on stratification, while the remaining three implement the iterative stratified sampling and estimation procedure.

Phase ❶ (lines 1–11): This phase builds the stratification model by generating LLM-derived signals at both triple and cluster levels. For each triple $t \in \mathcal{T}$, binary correctness predictions are obtained from all LLM models in $\mathcal{A}$ (lines 3–5), and aggregated via soft majority voting (line 6) as specified in Equation (6). KG entity clusters C are then derived by grouping triples according to their subject entity (line 8). For each cluster $\mathcal{G}[e] \in C$, the LLM-aggregated cluster score $\hat{\mu}_{e|\text{agg}}$ is computed as the mean of its constituent triple-level scores (line 10), following Equation (7).

Phase ❷ (lines 12–16): This phase conducts KG stratification. Using the CSRF method, the KG cluster family C is partitioned into up to Q strata based on the LLM-aggregated cluster scores (line 13), yielding the KG partition $\mathcal{P} = \{C_1, \ldots, C_Q\}$. Any zero-variance strata are merged with their nearest neighbors (line 14) to prevent over-stratification and ensure variance reduction. This merging procedure applies only to LLM-derived signals, not oracle signals. This asymmetry reflects a key difference: zero-variance strata in LLM signals can arise from systematic bias (overfitting) or hallucinations, whereas zero-variance strata in oracle signals mark semantically homogeneous clusters. Merging LLM zero-variance strata therefore helps recover coherent groupings lost to model distortions. The final number of strata Q is updated (line 15), and the stratum weights W_q are determined as the fraction of triples within each stratum (line 16).

Algorithm 1 LLM-based Stratified KG Accuracy Evaluation

Input:
KG triples $\mathcal{T}$;
LLM models $\mathcal{A} = \{\text{LLM}_1, \ldots, \text{LLM}_k\}$;
Desired number of strata Q;
Significance level α;
MoE threshold ε;
Second-stage sample size m.
Output: Estimated KG accuracy $\hat{\mu}$ with $1-\alpha$ HPD interval (l, u).

1: // **Phase ❶: LLM signal generation (stratification model)**
2: **for** each $t \in \mathcal{T}$ **do** ▷ LLM-aggregated triple score
3: **for** each $\text{LLM}_i \in \mathcal{A}$ **do**
4: $\text{pred}_i(t) \leftarrow \text{LLM}_i(\text{prompt}(t))$
5: **end for**
6: $\text{agg}(t) \leftarrow \frac{1}{k}\sum_{i=1}^{k} \text{pred}_i(t)$ ▷ Triple score (Eq. 6)
7: **end for**
8: $C \leftarrow \{\mathcal{G}[e] \mid \exists (s, p, o) \in \mathcal{T} : s = e\}$ ▷ Derive KG cluster family
9: **for** each $\mathcal{G}[e] \in C$ **do** ▷ LLM-aggregated cluster score
10: $\hat{\mu}_{e|\text{agg}} \leftarrow \frac{1}{M_e}\sum_{t \in \mathcal{G}[e]} \text{agg}(t)$ ▷ Cluster score (Eq. 7)
11: **end for**
12: // **Phase ❷: LLM-driven KG stratification**
13: $\mathcal{P} = \{C_1, \ldots, C_Q\} \leftarrow \text{CSRF}(\{(\mathcal{G}[e], \hat{\mu}_{e|\text{agg}})\}, Q)$ ▷ Stratify C
14: $\mathcal{P} \leftarrow$ merge any zero-variance strata with nearest neighbors
15: $Q \leftarrow |\mathcal{P}|$ ▷ Final number of strata
16: $W_q \leftarrow M_q / M \; \forall q \in \{1, \ldots, Q\}$ ▷ Compute strata weights
17: // **Phases ❸–❺: Iterative stratified sampling and estimation**
18: $\Omega_q \leftarrow \emptyset \; \forall q \in \{1, \ldots, Q\}$ ▷ Strata sample initialization
19: **for** $q = 1$ to Q **do** ▷ Warmup: sample batch from all strata
20: $\mathcal{B} \leftarrow \text{TWCS}(C_q, m)$ ▷ Batch of facts from stratum q
21: $\Omega_q \leftarrow \Omega_q \cup \mathbb{1}(\mathcal{B})$ ▷ Update sample w/ new annotations
22: **end for**
23: MoE $\leftarrow 1$ ▷ Constraint initialization
24: **while** MoE $> \varepsilon$ **do** ▷ Iterative procedure
25: Choose $q \sim \text{Cat}(W_1, \ldots, W_Q)$ ▷ Proportional allocation
26: $\mathcal{B} \leftarrow \text{TWCS}(C_q, m)$ ▷ Batch of facts from stratum q
27: $\Omega_q \leftarrow \Omega_q \cup \mathbb{1}(\mathcal{B})$ ▷ Update sample w/ new annotations
28: **for** $q = 1$ to Q **do**
29: $\hat{\mu}_q \leftarrow \hat{\mu}_{\text{TWCS}}(\Omega_q)$ ▷ Estimate stratum q accuracy (Eq. 3)
30: **end for**
31: $\hat{\mu} \leftarrow \hat{\mu}_{\text{STWCS}}(\{(\hat{\mu}_q, W_q)\}_{q=1}^{Q})$ ▷ Estimate KG accuracy (Eq. 8)
32: $(l, u) \leftarrow \text{HPD}(\bigcup_{q=1}^{Q} \Omega_q, \alpha)$ ▷ Compute $1-\alpha$ HPD CrI
33: MoE $\leftarrow (u - l)/2$ ▷ Update margin of error
34: **end while**
35: **return** $\hat{\mu}$, (l, u)

Phases ❸–❺ (lines 17–34): These phases perform iterative stratified sampling and estimation. Stratum-specific sample sets Ω_q are initialized (line 18), then a warmup step ensures coverage of all strata (lines 19–22): an initial batch of triples is sampled from each stratum via TWCS (line 20) and annotated (line 21). The MoE is set to 1 (line 23), and sampling and estimation repeat until it drops below ε (lines 24–34). In each iteration, a stratum q is drawn from a categorical distribution over stratum weights (line 25), enforcing proportional allocation. A batch of m triples is sampled from that stratum via TWCS (line 26), annotated, and added to Ω_q (line 27). Accuracy is then estimated per stratum (lines 28–30) using Equation (3), and the overall KG accuracy $\hat{\mu}$ is computed as the weighted sum of stratum estimates (line 31) via Equation (8). Finally, the $1-\alpha$ HPD CrI (l, u) is obtained from pooled samples across strata (line 32), and the MoE is updated (line 33).

Upon convergence, the algorithm returns the final KG accuracy estimate $\hat{\mu}$ and the corresponding $1-\alpha$ interval (l, u) (line 35).

5 SCALING LLMS: SIGNAL DISTILLATION

The presented stratified KG accuracy evaluation framework relies on LLMs to generate correctness signals for the entire KG. However, typical KGs contain hundreds of millions to billions of triples, making comprehensive LLM-based annotation computationally prohibitive and economically infeasible. To address scalability, in this section we propose a signal distillation strategy that transfers LLM capabilities to smaller, efficient models capable of annotating entire KGs at a fraction of the cost. Section 5.1 introduces knowledge distillation and formalizes our distillation procedure, while Section 5.2 discusses alternative scaling strategies.

5.1 Distilling LLM Signals

Knowledge distillation is a model compression technique that transfers knowledge from a large, complex *teacher* model to a smaller, efficient *student* model [6, 19]. The student model is trained not to match ground-truth labels, but to replicate the predictions of the more capable teacher, thereby preserving learned representations in a compact form suitable for large-scale deployment.

Knowledge distillation transfers LLM-derived stratification signals to smaller, locally deployable models [50]. Since LLMs incur high API costs, entail vendor dependency, and restrict offline deployment, distillation enables cost-effective and privacy-preserving signal generation for million- to billion-scale KGs.

Two primary distillation approaches exist: hard-target and soft-target distillation [6, 19]. Hard-target distillation [6] uses only the final predicted labels from the teacher, while soft-target distillation [19] transfers intermediate representations (probability distributions) containing richer information about class relationships. Although soft-target distillation generally yields superior performance with full model access, hard-target distillation is more practical for cloud-based LLM APIs where intermediate representations are often inaccessible. We adopt hard-target distillation to ensure our framework remains agnostic to underlying LLM architectures and can seamlessly integrate with any LLM provider.

Based on this, we now formalize the distillation procedure. The process begins by constructing a representative training dataset using LLM annotations on a sampled subset of KG triples.

Definition 6 (Distillation training set). For a KG with ternary relation $\mathcal{T}$, a distillation training set $\mathcal{D} = \{(t_i, y_i)\}_{i=1}^{n_\mathcal{D}}$ is a collection of $n_\mathcal{D}$ sampled triples paired with their corresponding LLM predictions, where $t_i \in \mathcal{T}$ and $y_i = \text{LLM}(\text{prompt}(t_i)) \in \{0, 1\}$ represents the binary correctness assessment.

The representativeness of $\mathcal{D}$ is essential for training distilled models that can effectively generalize across the full KG. Because our goal is to obtain a representative sample of the KG at the triple level, we apply simple random sampling to select $n_\mathcal{D}$ triples uniformly at random from $\mathcal{T}$, so that every triple has the same chance of being included [10]. After $\mathcal{D}$ has been constructed and annotated, we train a student model via standard supervised learning to minimize a loss function over $\mathcal{D}$, resulting in a computationally efficient surrogate for the LLM teacher.

Definition 7 (Distilled indicator function). Given a triple t, the distilled model is an indicator function $\text{distill-LLM}(t) : \mathcal{T} \to \{0, 1\}$ that outputs 1 if t is predicted correct and 0 otherwise.

The distilled model can directly replace the LLM within the stratified evaluation framework to predict triple-level correctness and derive cluster accuracy scores (cf. Definition 3).

To enhance robustness, we mirror multi-LLM aggregation by extending distillation to multiple teachers. Given a set of k LLM models $\mathcal{A} = \{\text{LLM}_1, \ldots, \text{LLM}_k\}$, we construct k distillation training sets $\{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$, where each $\mathcal{D}_i$ is annotated by LLM_i. Training separate student models on each $\mathcal{D}_i$ yields an ensemble $\{\text{distill-LLM}_1, \ldots, \text{distill-LLM}_k\}$. Distilled predictions are then aggregated using soft majority voting to compute triple-level scores (Equation (6)), which are averaged within entity clusters to obtain distill-LLM-aggregated cluster scores (Equation (7)).

Crucially, distilled models incur negligible inference costs, run locally without API dependencies, and enable offline workflows, thus making stratified evaluation of large-scale KGs practical where LLMs alone are infeasible.

5.2 Alternative Scaling Strategies

We investigated clustering-based stratification as an alternative: the intuition was that if clusters exhibited homogeneous accuracy within domains (e.g., biology, politics), annotating representative samples per cluster could yield signals generalizable to entire clusters. However, this approach failed to improve cost over distillation, and we omit technicalities in favor of reporting high-level insights.

First, we tested clustering strategies working directly on the graph structure, with the rationale that clusters encompassing tightly connected subgraphs would exhibit homogeneous accuracy. We tested both k-center clustering [9] and spectral clustering. Neither approach resulted in significant benefits, as KG topology is often quite sparse and thus does not result in meaningful clusters.

Alternatively, we tested whether subjects with high Jaccard similarity – computed over related objects or related predicates – would cluster into semantic domains with homogeneous accuracy. However, object-based similarity produced near-zero Jaccard scores due to KG sparsity, while predicate-based similarity formed coherent clusters but provided no accuracy signal. This suggests that structural features cannot be used effectively unless KG data are dense and free of topology-agnostic noise – i.e., random errors that do

Table 1: Statistics for NELL, DBPEDIA, FACTBENCH, and YAGO-{HQ, MQ, LQ} datasets.

Dataset	Triples	Clusters	Avg. cluster size	Accuracy (μ)	Skewness
NELL	1,860	817	2.28±1.49	0.91	−2.94
DBPEDIA	9,344	2,936	3.18±1.74	0.85	−1.97
FACTBENCH	4,219	1,157	3.65±4.24	0.28	+0.96
YAGO-HQ	20,433,311	4,391,886	4.65±3.96	0.75	−1.15
YAGO-MQ	20,433,311	4,293,627	4.76±3.86	0.50	0.00
YAGO-LQ	20,433,311	4,219,391	4.84±3.78	0.25	+1.15

not follow underlying KG patterns – a condition rarely met, especially in large-scale benchmarks that rely on random error injection in the absence of real ground-truth. As a result, assumptions of domain-specific accuracy homogeneity often fail. These findings reinforce that semantic signals (e.g., LLMs), rather than structural similarity, are essential for effective stratification, as they remain robust to sparsity and noise injection and can generalize to any KG.

6 EXPERIMENTAL SETUP

We report the datasets, stratification and distillation settings, sampling strategies, interval estimators, evaluation metrics, and computational resources used in our experiments.

Datasets. We evaluate on three real-world and three synthetic datasets, summarized in Table 1.

Real-world benchmarks: **NELL** [35] contains sports facts with crowdsourced annotations ($\mu = 0.91$). **DBPEDIA** [30] samples diverse topics from DBpedia [2], annotated by layman workers with expert quality control ($\mu = 0.85$). **FACTBENCH** [18] is a controlled benchmark with synthetically generated errors via multiple corruption strategies ($\mu = 0.28$).

Synthetic large-scale benchmarks: we corrupted YAGO 4.5 triples [42] by entity substitution (same type, semantically incorrect). This yields three 20M-triple datasets with controlled accuracy: **YAGO-HQ** ($\mu = 0.75$), **YAGO-MQ** ($\mu = 0.50$), **YAGO-LQ** ($\mu = 0.25$).

In realistic KG accuracy evaluation settings, label distributions are often highly imbalanced, with correct triples vastly outnumbering incorrect ones. We therefore report in Table 1 the skewness of each dataset label distribution. Following Bulmer [7], distributions with skewness less than -1 or greater than $+1$ are considered highly skewed; values between $[-1, -0.5)$ or $(0.5, 1]$ indicate moderate skewness; and values in $[-0.5, 0.5]$ indicate approximate symmetry. As shown in Table 1, three of the six datasets exhibit highly skewed label distributions toward the positive class (i.e., left skewness < -1), with NELL and DBPEDIA – both real-world benchmarks – exceeding the threshold by factors of three and two, respectively. On the other hand, FACTBENCH and YAGO-LQ display moderate (+0.96) and high (+1.15) right skewness, respectively, reflecting scenarios with significant instrumental or systematic noise in KG construction. Finally, YAGO-MQ is approximately symmetric (skewness = 0.00). Overall, the considered datasets span all skewness regimes, with particular emphasis on realistic, highly positive-skewed settings.

Stratification. We employ multiple stratification signals. First, we include a topological baseline using the **size** of entity clusters, as proposed by Gao et al. [17]. Second, we incorporate four open-source LLMs from different families and of varying sizes: **cohere** [11] (Cohere-command-r-08-2024, 32B parameters), **deepseek-v3** [13] (DeepSeek-V3-0324, 685B parameters), **llama3.1-8b-it** [44] (Llama-3.1-8B, 8B parameters), and **mistral-nemo** [45] (Mistral-Nemo, 12B parameters). These LLMs also serve as inputs to our aggregation strategy, **llm-agg**. Leveraging open-source LLMs enables reproducibility and deployment in privacy-sensitive settings, while their heterogeneity – both family- and size-wise – enhances aggregation robustness. We also consider a **random** baseline in which triples are labeled correct/incorrect with 0.5 probability. These random predictions are used to compute cluster accuracy scores as with LLMs. Comparing stratified solutions against the random baseline reveals the effectiveness of each stratification signal for KG accuracy evaluation. Finally, we use ground-truth annotations to compute true cluster accuracies, which serve as an **oracle** stratification signal. Although unattainable in practice, the oracle provides a reference lower bound that helps quantify how well each proxy signal approximates ideal stratification.

Prior work suggests partitioning the KG into a small number of strata $Q \in \{2, \ldots, 10\}$ to ensure sufficiently large stratum sizes [17, 28, 30]. Hence, we set 10 as the upper bound on the number of strata and use it as input parameter for CSRF.[2]

Distillation. To scale to large KGs, we draw a simple random sample of 50K triples from each of YAGO-{HQ, MQ, LQ}, yielding representative subsets of the KGs. We then derive the distillation training sets $\{\mathcal{D}_i\}_{k=1}^{4}$ by pairing the sampled triples with the annotations made by each of the four considered LLMs. This way, each LLM annotates no more than 0.25% of the triples in a given KG.

To distill stratification signals, we train one BERT model (bert-base-uncased) per training set following standard settings [15]: 2 epochs, batch size 16, and learning rate 5e−5. We denote distilled models as **distill-{LLM}** and their aggregation as **distill-llm-agg**.

Sampling Strategies. We use **TWCS** as the reference SotA baseline and apply **STWCS** to the strata produced by each stratification signal. Following the recommendation of Gao et al. [17], we set the second-stage size of TWCS and STWCS to $m = 3$ for all KGs, as they exhibit small cluster sizes (see Table 1).

Interval Estimation. We build $1 - \alpha$ CrIs using the *adaptive* HPD (*a*HPD) algorithm [29], the current SotA. *a*HPD simultaneously evaluates multiple priors to produce competing $1 - \alpha$ HPD intervals and selects the most efficient solution without requiring a prior to be fixed in advance. Consistently with the literature, we use three uninformative priors – Kerman Beta$(\frac{1}{3}, \frac{1}{3})$, Jeffreys Beta$(\frac{1}{2}, \frac{1}{2})$, and Uniform Beta$(1, 1)$ – and apply design effect adjustments for complex sampling designs [26].

Evaluation Metrics. To measure how effectively LLMs assess facts, we use Sensitivity (Sens), the proportion of true positives correctly predicted; Specificity (Spec), the proportion of true negatives

[2]When using the oracle stratification signal, we do not merge zero-variance strata with their nearest neighbors.

Table 2: Microsoft Azure model pricing. Costs are divided between input and output tokens, and reported per 1K tokens.

	LLM Pricing (1K tokens)			
	cohere	deepseek-v3	llama3.1-8b-it	mistral-nemo
Input	\$0.00015	\$0.00114	\$0.00030	\$0.00015
Output	\$0.00060	\$0.00456	\$0.00061	\$0.00015

correctly predicted; and Macro F1 (MaF1), an aggregate performance metric robust to class imbalance. Ground truth labels come from the manual (and synthetic) annotations for each dataset.

To evaluate whether a stratification signal reliably approximates cluster accuracy (the latent feature), we use Spearman's rank correlation (ρ) and the coefficient of determination (R^2). Spearman's ρ quantifies the monotonic relationship between the proxy and true cluster accuracies. R^2 measures the proportion of variance in cluster accuracies explained by the stratification signal. An R^2 of 1 indicates perfect separation (i.e., clusters within a stratum have identical accuracy), while 0 indicates no explanatory power. Together, Spearman's ρ captures how well the signal tracks accuracy, whereas R^2 captures how well it partitions clusters – jointly characterizing the cost-effectiveness of the stratification.

To assess the effectiveness of the stratified evaluation framework, we set the significance level to the standard value $\alpha = 0.05$ and the upper bound for the MoE to $\varepsilon = 0.05$, as done in [17, 28, 29]. Performance is measured using two metrics: the number of annotated triples and the annotation cost (in hours). The latter follows the cost model by Gao et al. [17], which distinguishes entity identification from fact verification: $\text{cost}(\mathcal{T}_\mathcal{S}) = |\mathcal{E}_\mathcal{S}| \cdot c_1 + |\mathcal{T}_\mathcal{S}| \cdot c_2$. Here, c_1 and c_2 are the average time (in hours) for entity identification and fact verification, respectively. As in prior work [17, 28, 29], we set $c_1 = 0.0125$ and $c_2 = 0.0069$. Note that this cost model captures only human annotation time, which constitutes the online and time-critical component of the evaluation pipeline, whereas LLM annotations for stratification are executed offline in batch mode and thus do not affect the effective turnaround time of KG accuracy estimation. Results are reported once MoE ≤ 0.05, so interval widths are omitted. Accuracy estimates are reported only for the proposed llm-agg strategy, as it serves as the reference point for comparison with LLMs used as direct accuracy estimators. We omit the estimates for the other sampling-based strategies, since all methods yield unbiased estimates with minimal deviation from ground truth upon convergence. Interest readers can find the full set of accuracy estimates for all sampling-based strategies in the online repository. To account for sampling variability, we repeat the evaluation procedure 1,000 times and report mean and standard deviation for both annotation (cost) metrics.

Computational Resources. All methods were implemented in Python 3. For LLMs, we used Microsoft Azure services;[3] input and output costs per 1K tokens for each model are reported in Table 2 (in \$). Distilled models and evaluation were trained and run on a shared in-house Linux machine with an Intel Xeon Gold 6140M CPU @ 2.30GHz, 1.5TB RAM, and 2 NVIDIA A40 GPUs.

[3]https://ai.azure.com/

Table 3: LLM performance on fact validation, measured by Sensitivity (Sens), Specificity (Spec), and Macro F1 (MaF1) across NELL, DBPEDIA, and FACTBENCH datasets.

	NELL			DBPEDIA			FACTBENCH		
	$\mu = 0.91$			$\mu = 0.85$			$\mu = 0.28$		
LLM	Sens	Spec	MaF1	Sens	Spec	MaF1	Sens	Spec	MaF1
cohere	0.88	0.57	0.66	0.62	0.71	0.55	0.78	0.82	0.77
deepseek-v3	0.82	0.65	0.62	0.52	0.82	0.51	0.69	0.94	0.83
llama3.1-8b-it	0.91	0.43	0.64	0.75	0.43	0.55	0.70	0.77	0.71
mistral-nemo	0.72	0.78	0.58	0.52	0.72	0.49	0.46	0.93	0.72

7 EXPERIMENTAL RESULTS

Our experimental objectives are: (i) assess LLMs as direct accuracy estimators; (ii) measure how well LLM-derived signals approximate ground-truth stratification; (iii) compare the cost-effectiveness of LLM stratification against SotA baselines; and (iv) validate that signal distillation preserves stratification quality on large-scale KGs while reducing inference costs.

7.1 Summary of the Experimental Findings

(F1) **LLMs are unreliable direct accuracy estimators.** LLM-derived accuracy estimates deviate up to 45% from ground truth (Sec. 7.2), justifying our stratification approach.

(F2) **LLM-derived signals outperform topology-based stratification.** LLM signals correlate more strongly with true cluster accuracy and explain more variance than topology-based alternatives (Sec. 7.3).

(F3) **Stratified evaluation with LLM signals achieves up to 50% cost reduction.** LLM-stratified sampling reduces annotation costs compared to unstratified and topology-based baselines across all datasets, with gains ranging from 11% (challenging) to 54% (high-accuracy KGs) (Sec. 7.4).

(F4) **Distilled models preserve stratification quality at scale.** Distilled models retain LLM performance gains while reducing inference costs to near-zero, enabling large-scale evaluation (Sec. 7.5).

7.2 LLMs as Direct Accuracy Estimators

The performance of LLMs on fact validation are reported in Table 3. We can see that LLM errors are strongly dataset-dependent. On FACTBENCH, where incorrect triples are synthetically generated, all models achieve high specificity, indicating effectiveness in detecting systematic corruption. Conversely, on NELL – where most triples are correct ($\mu = 0.91$) – all models exhibit high sensitivity but markedly lower specificity, revealing a tendency to over-accept facts and miss genuinely incorrect triples. The situation further degrades on DBPEDIA, the most heterogeneous dataset, where sensitivity and specificity vary widely across models and Macro F1 can be as low as 0.49 (mistral-nemo).

These asymmetric error patterns make LLMs unreliable as direct KG accuracy estimators. Indeed, the estimated accuracy derived from LLM predictions can be expressed as $\hat{\mu}_{\text{LLM}} = \mu \cdot \text{Sens}_{\text{LLM}} + (1-\mu) \cdot (1 - \text{Spec}_{\text{LLM}})$, which implies that different combinations of sensitivity and specificity may yield similar accuracy estimates. Consequently, false positives and false negatives may compensate

Table 4: Comparison between KG accuracy estimates derived from LLM predictions and ground-truth KG accuracy obtained from human (and synthetic) annotations. For each LLM-derived estimate, we report the absolute deviation ($|\Delta|$) from μ. We also report llm-agg (ours) with accuracy averaged over 1,000 repetitions, serving as reference for sampling-based solutions involving minimal human annotation. Estimates closest to μ are shown in bold.

	NELL	DBPEDIA	FACTBENCH
	$\mu = 0.91$	$\mu = 0.85$	$\mu = 0.28$
LLM	$\hat{\mu}_{\text{LLM}}(\lvert\Delta\rvert)$	$\hat{\mu}_{\text{LLM}}(\lvert\Delta\rvert)$	$\hat{\mu}_{\text{LLM}}(\lvert\Delta\rvert)$
cohere	0.84 (0.07)	0.57 (0.28)	0.35 (0.07)
deepseek-v3	0.78 (0.13)	0.47 (0.38)	0.24 (0.04)
llama3.1-8b-it	0.88 (0.03)	0.72 (0.13)	0.36 (0.08)
mistral-nemo	0.68 (0.23)	0.48 (0.37)	0.18 (0.10)
llm-agg	**0.92 (0.01)**	**0.87 (0.02)**	**0.28 (0.00)**

each other, producing numerically close accuracy estimates despite significant instance-level errors. Small deviations from KG accuracy thus provide no guarantee of reliable fact validation.

This effect is confirmed in Table 4, where we compare the KG accuracy estimates derived from LLM predictions with ground-truth values. The comparison highlights significant and systematic deviations between LLM-derived and ground-truth KG accuracies across all datasets, as further evidenced by the reported absolute deviations $|\Delta|$. The issue is most severe on DBPEDIA, where deepseek-v3 estimates an accuracy of 0.47 instead of 0.85, corresponding to a 45% relative error (i.e., $|\Delta| = 0.38$). While some LLMs produce estimates close to true values on specific datasets, none is consistently robust. Besides, even small deviations stem from error compensation rather than fact-level reliability, underscoring LLMs inherent variability and insufficient robustness as direct KG accuracy estimators.

To test whether larger proprietary models overcome this limitation, we evaluated GPT-4o as a direct estimator; for brevity, these results are omitted from Table 3. Despite its higher cost, improvements are minor: GPT-4o matches open models on NELL (llama3.1-8b-it) and DBPEDIA (deepseek-v3) and is only slightly better on FACTBENCH. This confirms that model scale alone does not ensure reliable KG accuracy estimation. Rather, all models share a systematic failure mode: strong performance on semi-synthetic benchmarks but poor generalization to real-world facts.

Overall, LLMs are suboptimal fact validators and thus cannot directly estimate KG accuracy. Moreover, a finer-grained comparison of misclassified triples, reported in the online repository for space reasons, reveals limited overlap between LLM error sets, indicating heterogeneous failure modes across models. This suggests that individual LLMs capture partially complementary signals about fact validity. Together, these findings motivate the use of LLMs as stratification signals and their aggregation to mitigate individual biases and errors. This way, LLMs are only used to guide sampling, while KG accuracy estimates rely exclusively on human annotations. Such separation prevents error-compensation effects, as the estimator does not depend on LLM predictions. In other words, LLM judgments affect sampling efficiency, but not estimator correctness. Our proposed llm-agg method – shown in Table 4 as the reference for LLM-based stratified sampling – produces accuracy estimates that are consistently closest to the ground truth across all datasets (with $|\Delta| \leq 0.02$). This minimal deviation is expected, as llm-agg is grounded in sound statistical principles, ensuring stable and accurate estimates even in challenging settings.

Table 5: Correlation analysis of size and LLM-based stratification signals with true cluster accuracy, measured via Spearman's ρ and coefficient of determination R^2.

	NELL		DBPEDIA		FACTBENCH	
	$\mu = 0.91$		$\mu = 0.85$		$\mu = 0.28$	
Signal	ρ	R^2	ρ	R^2	ρ	R^2
size	-0.19	0.00	**-0.30**	0.01	-0.20	0.10
cohere	**0.50**	0.24	0.21	0.03	0.45	0.22
deepseek-v3	**0.50**	0.24	0.16	0.03	**0.55**	0.38
llama3.1-8b-it	0.41	0.15	0.20	0.01	0.39	0.15
mistral-nemo	0.44	0.21	0.14	0.02	0.29	0.17
llm-agg	0.48	**0.35**	0.24	**0.06**	**0.55**	**0.39**

7.3 LLMs as Stratification Signals

The effectiveness of stratification signals as proxies for cluster accuracy is reported in Table 5, which provides the following insights. Cluster size, the topological baseline, shows weak correlation with cluster accuracy. Moreover, this correlation is negative, indicating that smaller clusters tend to have higher accuracy – in contrast with prior findings [17]. At the same time, when cluster size is used as a stratification signal, it also fails to generate strata that explain variance in (cluster) accuracy, yielding near-zero R^2 values on real-world datasets, NELL and DBPEDIA, and 0.10 on FACTBENCH, the lowest score among the considered signals. Conversely, LLM-derived stratification signals exhibit higher predictive and explanatory power, reaching correlations with true cluster accuracy of up to 0.55 on FACTBENCH. Among these, our aggregation strategy (llm-agg) achieves one of the strongest Spearman's ρ values and produces the best R^2 scores across datasets, confirming the robustness of the aggregated signal. All signals yield low R^2 values on DBPEDIA, highlighting its difficulty as a verification benchmark. We omit random and oracle baselines, which by definition have Spearman's ρ and R^2 of 0 and 1, respectively.

7.4 Stratified KG Accuracy Evaluation via LLMs

Table 6 reports the performance of stratified KG evaluation using our LLM-aggregated strategy (llm-agg), compared with single-LLM strategies and size-based stratification. To better understand the effectiveness of stratification signals for KG accuracy evaluation, we also include results for random and oracle stratification. Finally, we report results for TWCS, the current unstratified SotA method. To assess whether llm-agg significantly outperforms alternative strategies, we perform independent t-tests against each baseline, applying Bonferroni correction to account for multiple comparisons. We report statistical significance when $p < 0.01$. The oracle setting is excluded from this analysis, as it serves only as a reference point for ideal cost-effectiveness rather than as a practical baseline.

From Table 6 (left side) we can see that across all datasets, llm-agg delivers statistically significant cost reductions relative to SotA

Table 6: Performance of KG evaluation methods on NELL, DBPEDIA, and FACTBENCH. Best scores are in bold. $^{\triangle}$ marks methods significantly more expensive than llm-agg (ours), based on independent t-tests with Bonferroni correction ($p < 0.01$). We also report expected monetary costs, split into human (Hum), machine (LLM), and total (Tot). Human costs assume three expert annotators per sampled triple at $5.71 each [36]; machine costs follow LLM pricing for prompts and outputs (Table 2). Lowest total costs are in bold. Oracle stratification is shown only as an ideal lower bound.

NELL ($\mu = 0.91$)						
Sampling	Signal	Triples	Cost (hours)	Hum ($)	LLM ($)	Tot ($)
TWCS	–	116 ± 66	1.45 ± 0.83$^{\triangle}$	1,987	0	1,987
STWCS	random	109 ± 71	1.36 ± 0.89$^{\triangle}$	1,867	0	1,867
STWCS	size	106 ± 75	1.31 ± 0.94$^{\triangle}$	1,816	0	1,816
STWCS	cohere	85 ± 56	1.06 ± 0.71$^{\triangle}$	1,456	< 1	1,456
STWCS	deepseek-v3	82 ± 53	1.02 ± 0.67$^{\triangle}$	1,405	< 1	1,405
STWCS	llama3.1-8b-it	78 ± 57	0.97 ± 0.72$^{\triangle}$	1,336	< 1	1,336
STWCS	mistral-nemo	92 ± 58	1.15 ± 0.73$^{\triangle}$	1,576	< 1	1,576
STWCS	llm-agg	**54 ± 40**	**0.67 ± 0.50**	925	1	**926**
STWCS	oracle	42 ± 30	0.51 ± 0.38	719	0	719
DBPEDIA ($\mu = 0.85$)						
Sampling	Signal	Triples	Cost (hours)	Hum ($)	LLM ($)	Tot ($)
TWCS	–	224 ± 85	2.57 ± 0.97$^{\triangle}$	3,837	0	3,837
STWCS	random	222 ± 95	2.56 ± 1.10$^{\triangle}$	3,803	0	3,803
STWCS	size	218 ± 93	2.52 ± 1.07$^{\triangle}$	3,734	0	3,734
STWCS	cohere	203 ± 94	2.33 ± 1.09	3,477	< 1	3,478
STWCS	deepseek-v3	203 ± 100	2.33 ± 1.15	3,477	4	3,481
STWCS	llama3.1-8b-it	209 ± 99	2.41 ± 1.14$^{\triangle}$	3,580	1	3,581
STWCS	mistral-nemo	203 ± 102	2.33 ± 1.17	3,477	< 1	3,478
STWCS	llm-agg	**195 ± 100**	**2.24 ± 1.14**	3,340	7	**3,347**
STWCS	oracle	46 ± 4	0.52 ± 0.05	788	0	788
FACTBENCH ($\mu = 0.28$)						
Sampling	Signal	Triples	Cost (hours)	Hum ($)	LLM ($)	Tot ($)
TWCS	–	252 ± 52	2.89 ± 0.60$^{\triangle}$	4,317	0	4,317
STWCS	random	252 ± 64	2.89 ± 0.74$^{\triangle}$	4,317	0	4,317
STWCS	size	231 ± 70	2.65 ± 0.81$^{\triangle}$	3,957	0	3,957
STWCS	cohere	235 ± 60	2.70 ± 0.69$^{\triangle}$	4,026	< 1	4,026
STWCS	deepseek-v3	205 ± 59	2.37 ± 0.68	3,512	2	3,514
STWCS	llama3.1-8b-it	240 ± 58	2.76 ± 0.68$^{\triangle}$	4,111	< 1	4,112
STWCS	mistral-nemo	229 ± 71	2.63 ± 0.83$^{\triangle}$	3,923	< 1	3,923
STWCS	llm-agg	**200 ± 63**	**2.30 ± 0.72**	3,426	3	**3,429**
STWCS	oracle	102 ± 47	1.18 ± 0.54	1,747	0	1,747

baselines, with up to 49% improvement over size stratification and 54% over TWCS. llm-agg statistically outperforms most single-LLM methods, confirming that aggregation helps reduce model-specific biases and outlier predictions. The only exception is DBPEDIA, where its advantage over most single-LLM methods is not statistically significant, again highlighting DBPEDIA's complexity. Size stratification performs almost identically to random and TWCS on NELL and DBPEDIA, confirming that size is a negligible stratification signal and supporting [17, 28]. These performance differences match our analysis of stratification signal quality (Section 7.3): llm-agg's superiority stems from its highest explanatory power (R^2) and strong correlation with true cluster accuracy (Spearman's ρ).

Table 7: LLM performance on fact validation, measured by Sensitivity (Sens), Specificity (Spec), and Macro F1 (MaF1) on the 50K sampled triples from the YAGO-{HQ, MQ, LQ} KGs used to construct the distillation training sets.

	YAGO-HQ			YAGO-MQ			YAGO-LQ		
	$\mu = 0.75$			$\mu = 0.50$			$\mu = 0.25$		
LLM	Sens	Spec	MaF1	Sens	Spec	MaF1	Sens	Spec	MaF1
cohere	0.61	0.93	0.67	0.61	0.93	0.77	0.56	0.92	0.75
deepseek-v3	0.37	0.99	0.53	0.38	0.99	0.65	0.37	0.99	0.72
llama3.1-8b-it	0.68	0.67	0.64	0.53	0.81	0.67	0.67	0.67	0.63
mistral-nemo	0.47	0.93	0.58	0.47	0.93	0.69	0.34	0.95	0.67

While the above analysis focuses on annotation volume and human effort, it does not account for the cost of using LLMs. Since LLMs are typically accessed through pay-per-request cloud services (e.g., Microsoft Azure), their pricing may offset the gains achieved through stratification. Using the model prices in Table 2, we compute the monetary cost of processing all triples in the KG with LLMs, including both prompt tokens (input) and model responses (output). For human cost, we assume each sampled triple is judged by at least three expert annotators – a standard setup for annotation tasks [1] – at a rate of $5.71 per triple per annotator, based on the expert pricing reported by Paulheim [36]. Table 6 (right side) presents the resulting monetary costs, separating human and LLM components for all methods. For llm-agg, machine costs reflect the combined usage of all four LLMs.

The monetary analysis in Table 6 (right side) shows that LLM costs are negligible compared to human expenses across all datasets, confirming the cost-effectiveness of LLM-based stratification. In practice, LLM costs are fully offset by the reduced number of triples needing expert annotation, with llm-agg consistently being the most economical non-oracle method. While LLM costs are negligible for small datasets (thousands of triples), they can dominate total costs for large KGs with millions of triples (e.g., YAGO-{HQ, MQ, LQ}). This underscores the need for more computationally and economically efficient methods to scale LLM-based KG evaluation, motivating signal distillation.

7.5 Scaling KG Evaluation via Signal Distillation

To determine whether distilling LLM signals enables cost-effective large-scale KG accuracy evaluation, we extend our analysis to the YAGO-{HQ, MQ, LQ} datasets. Table 7 reports the fact validation performance of the four considered LLMs on the 50K sampled triples per KG used to construct the distillation training sets, providing the reference point for the subsequent distillation experiments. Performance trends are consistent with those observed on smaller datasets: all LLMs underperform, errors remain asymmetric with very high specificity but markedly lower sensitivity – mirroring the behavior on FACTBENCH, as LLMs effectively detect synthetically corrupted triples but fail to reliably identify correct ones.

The distilled models – trained on LLM annotations over the 50K sampled triples per KG – exhibit performance consistent with their

Table 8: Distilled model performance on fact validation, measured by Sensitivity (Sens), Specificity (Spec), and Macro F1 (MaF1) across YAGO-{HQ, MQ, LQ} datasets.

	YAGO-HQ			YAGO-MQ			YAGO-LQ		
	$\mu = 0.75$			$\mu = 0.50$			$\mu = 0.25$		
LLM (distilled)	Sens	Spec	MaF1	Sens	Spec	MaF1	Sens	Spec	MaF1
distill-cohere	0.61	0.84	0.65	0.58	0.90	0.73	0.49	0.94	0.74
distill-deepseek-v3	0.35	0.94	0.50	0.34	0.95	0.61	0.23	0.98	0.61
distill-llama3.1-8b-it	0.74	0.52	0.61	0.52	0.80	0.65	0.69	0.64	0.62
distill-mistral-nemo	0.49	0.88	0.58	0.47	0.90	0.67	0.42	0.92	0.69

Table 9: Comparison between KG accuracy estimates derived from distilled-model predictions and ground-truth KG accuracy obtained from synthetic annotations. We report the absolute deviation ($|\Delta|$) from μ. We also report distill-llm-agg (ours) averaged over 1,000 repetitions, as a sampling-based reference. Estimates closest to μ are shown in bold.

	YAGO-HQ	YAGO-MQ	YAGO-LQ
	$\mu = 0.75$	$\mu = 0.50$	$\mu = 0.25$
LLM (distilled)	$\hat{\mu}_{\text{distill-LLM}}(\lvert\Delta\rvert)$	$\hat{\mu}_{\text{distill-LLM}}(\lvert\Delta\rvert)$	$\hat{\mu}_{\text{distill-LLM}}(\lvert\Delta\rvert)$
distill-cohere	0.50 (0.25)	0.34 (0.16)	0.17 (0.08)
distill-deepseek-v3	0.27 (0.48)	0.19 (0.31)	0.07 (0.18)
distill-llama3.1-8b-it	0.67 (0.08)	0.36 (0.14)	0.44 (0.19)
distill-mistral-nemo	0.40 (0.35)	0.28 (0.22)	0.17 (0.08)
distill-llm-agg	**0.75 (0.00)**	**0.50 (0.00)**	**0.25 (0.00)**

teacher LLMs, as shown in Table 8, though slightly lower. This is an expected outcome of knowledge distillation, where student models are optimized to mimic rather than outperform their teachers. Notably, the performance of distilled models remains very close to that of their teacher LLMs, confirming their potential as scalable, efficient stratification signals for KG accuracy evaluation.

Similar to the results for real LLMs (Table 4), the KG accuracy estimates obtained from distilled-model predictions (Table 9) show substantial and systematic divergences from the ground truth, as reflected by the reported absolute deviations (up to $|\Delta| = 0.48$). These discrepancies mostly stem from low sensitivity profiles, underscoring that neither LLMs nor their distilled versions can be used as reliable, direct accuracy estimators. Again, a fine-grained analysis of misclassified triples (reported online) confirms heterogeneous and only partially overlapping error patterns across models, indicating complementary failure modes and motivating the aggregation of LLM signals. In this regard, our distill-llm-agg approach – reported as the LLM-based sampling reference – matches ground-truth accuracies across all YAGO datasets ($|\Delta| = 0.00$), further reinforcing the effectiveness of adopting these models as stratification signals rather than as direct estimators.

Despite this, analyses in Table 10 show that distilled models serve as effective stratification signals. Compared to cluster size, LLM-distilled solutions yield higher Spearman's ρ and R^2, while size alone fails to explain any variance in true cluster accuracies (R^2 near 0). Among distilled models, the one trained on cohere's annotations (distill-cohere) emerges as the stratification signal most aligned with true cluster accuracies and explains the largest share

Table 10: Correlation analysis of size and distilled-based stratification signals with true cluster accuracy, measured via Spearman's ρ and coefficient of determination R^2.

	YAGO-HQ		YAGO-MQ		YAGO-LQ	
	$\mu = 0.75$		$\mu = 0.50$		$\mu = 0.25$	
Signal	ρ	R^2	ρ	R^2	ρ	R^2
size	-0.31	0.02	-0.13	0.02	0.10	0.01
distill-cohere	**0.36**	0.12	**0.48**	**0.27**	**0.53**	**0.29**
distill-deepseek-v3	0.19	0.09	0.30	0.15	0.36	0.14
distill-llama3.1-8b-it	0.27	0.06	0.29	0.12	0.31	0.09
distill-mistral-nemo	0.25	0.10	0.36	0.17	0.41	0.17
distill-llm-agg	0.35	**0.15**	0.47	0.24	0.52	0.28

of variance. Nevertheless, the aggregated variant (distill-llm-agg) performs similarly and attains the highest R^2 on YAGO-HQ.

The stratification analysis in Table 11 (left side) shows that the distilled aggregation method significantly outperforms TWCS, size stratification, and most single distilled models, with up to a 15% gain over size stratification and TWCS. As in the real LLMs results (Table 6), distill-llm-agg performs best on YAGO-HQ, achieving the highest R^2, while distill-cohere yields larger cost reductions on YAGO-MQ and YAGO-LQ, where its Spearman's ρ and R^2 are higher. The performance gap between distill-llm-agg and distill-cohere is not statistically significant, indicating practical equivalence and confirming that aggregation reduces individual-model biases and achieves top performance across all large-scale datasets. Although distill-cohere is the strongest single model, no model consistently dominates, mirroring the LLM results on NELL, DBPEDIA, and FACTBENCH; this underscores aggregation as the most reliable stratification signal for KG accuracy evaluation.

Regarding annotation costs with LLMs and their distilled variants, annotating 50K triples with the four LLMs incurs minimal machine expenses (i.e., a total of $34). By contrast, the training and inference costs of the distilled models are negligible, as they run on our shared in-house server and are effectively amortized. As shown in Table 11 (right side), total costs remain dominated by human annotation across all YAGO datasets, with LLM costs accounting for a minimal fraction of the total. As a result, the distilled setup yields monetary savings comparable to those reported in Table 6.

Hence, the large-scale analyses presented here – mirroring the small-scale ones with LLMs – demonstrate that LLM-distilled signals can be efficiently and effectively scaled to large KGs without compromising the observed performance gains. This confirms that distillation enables cost-effective KG accuracy evaluation while preserving the benefits of full LLMs.

8 RELATED WORK

Early research overlooked the problem of evaluating KG accuracy at scale. A first attempt was KGEval [35], an iterative method that alternates between crowdsourced annotations and probabilistic inference using type and Horn-clause constraints [27, 33]. Although pioneering, KGEval suffers from poor scalability and error propagation, which limits its applicability to real-world KGs [17].

To avoid error propagation and scale to large KGs, Gao et al. [17] reframed the problem as statistical estimation, proposing sampling

Table 11: Performance of KG evaluation methods on YAGO-{HQ, MQ, LQ}. Best scores are in bold. $^{\triangle}$ indicates methods significantly more expensive than distill-llm-agg (ours), based on independent t-tests with Bonferroni correction ($p < 0.01$). $^{\blacktriangledown}$ denotes the opposite case that never occurs. We also report expected monetary costs (Hum, LLM, Tot). Human costs has the same setup as in Table 6, while machine costs are LLM processing over the 50K sampled triples per KG used for distillation (Table 2). Training and inference costs of distilled models are negligible. Lowest total costs are shown in bold. Oracle stratification serves as the ideal lower bound.

YAGO-HQ ($\mu = 0.75$)						
Sampling	Signal	Triples	Cost (hours)	Hum ($)	LLM ($)	Tot ($)
TWCS	–	279 ± 41	3.16 ± 0.47$^{\triangle}$	4,779	0	4,779
STWCS	random	278 ± 64	3.15 ± 0.72$^{\triangle}$	4,762	0	4,762
STWCS	size	283 ± 66	3.20 ± 0.75$^{\triangle}$	4,848	0	4,848
STWCS	distill-cohere	263 ± 56	2.98 ± 0.63	4,505	3	4,508
STWCS	distill-deepseek-v3	271 ± 76	3.07 ± 0.86$^{\triangle}$	4,642	22	4,664
STWCS	distill-llama3.1-8b-it	260 ± 77	2.94 ± 0.87	4,454	6	4,460
STWCS	distill-mistral-nemo	272 ± 61	3.09 ± 0.69$^{\triangle}$	4,659	3	4,662
STWCS	distill-llm-agg	**256 ± 68**	**2.90 ± 0.77**	4,385	34	**4,419**
STWCS	oracle	117 ± 50	1.33 ± 0.57	2,007	0	2,007
YAGO-MQ ($\mu = 0.50$)						
Sampling	Signal	Triples	Cost (hours)	Hum ($)	LLM ($)	Tot ($)
TWCS	–	388 ± 39	4.39 ± 0.44$^{\triangle}$	6,646	0	6,646
STWCS	random	392 ± 53	4.44 ± 0.60$^{\triangle}$	6,715	0	6,715
STWCS	size	393 ± 56	4.45 ± 0.63$^{\triangle}$	6,732	0	6,732
STWCS	distill-cohere	**345 ± 61**	**3.91 ± 0.69**	5,910	3	**5,913**
STWCS	distill-deepseek-v3	364 ± 71	4.13 ± 0.81$^{\triangle}$	6,235	22	6,257
STWCS	distill-llama3.1-8b-it	373 ± 60	4.23 ± 0.68$^{\triangle}$	6,389	6	6,395
STWCS	distill-mistral-nemo	364 ± 67	4.12 ± 0.76$^{\triangle}$	6,235	3	6,238
STWCS	distill-llm-agg	350 ± 53	3.97 ± 0.59	5,996	34	6,030
STWCS	oracle	176 ± 59	1.99 ± 0.67	3,015	0	3,015
YAGO-LQ ($\mu = 0.25$)						
Sampling	Signal	Triples	Cost (hours)	Hum ($)	LLM ($)	Tot ($)
TWCS	–	293 ± 46	3.31 ± 0.52$^{\triangle}$	5,019	0	5,019
STWCS	random	294 ± 63	3.32 ± 0.72$^{\triangle}$	5,036	0	5,036
STWCS	size	293 ± 73	3.32 ± 0.82$^{\triangle}$	5,019	0	5,019
STWCS	distill-cohere	**243 ± 65**	**2.75 ± 0.73**	4,163	3	**4,166**
STWCS	distill-deepseek-v3	272 ± 58	3.08 ± 0.66$^{\triangle}$	4,659	22	4,681
STWCS	distill-llama3.1-8b-it	285 ± 64	3.22 ± 0.73$^{\triangle}$	4,882	6	4,888
STWCS	distill-mistral-nemo	261 ± 73	2.95 ± 0.83$^{\triangle}$	4,471	3	4,474
STWCS	distill-llm-agg	250 ± 66	2.83 ± 0.75	4,283	34	4,317
STWCS	oracle	119 ± 49	1.34 ± 0.56	2,038	0	2,038

strategies with theoretical guarantees. They showed that cluster sampling (TWCS) can reduce annotation cost compared to simple random sampling. However, their estimates rely on the Wald confidence interval [8], which is known to be unreliable for binomial proportions [5, 47]. Subsequent work addressed this limitation by replacing Wald with more robust intervals: the Wilson confidence interval [28], and later Bayesian HPD credible intervals [29], which offer reliable one-shot probabilistic guarantees and lower annotation costs. In this context, Marchesin and Silvello [29] proposed the *a*HPD algorithm, which removes the need for manual prior selection, enabling practical Bayesian evaluation of KG accuracy.

Recent work has focused on interval estimation while leaving sampling strategies unchanged [28, 29]. We close this gap by replacing ineffective topological stratification with LLM-based semantic stratification, significantly reducing costs compared to current SotA methods. For large-scale deployment, we further distill LLM semantic capabilities into smaller models, retaining KG accuracy-estimation benefits at a fraction of the cost.

9 CONCLUSIONS

This work revisits stratified sampling for KG accuracy evaluation and shows that its main limitation – the difficulty of constructing effective strata – can be overcome by repurposing LLMs as *stratification signals*. Although LLMs are unreliable as direct fact validators (with accuracy estimates deviating up to 45% from ground truth), their aggregated predictions provide highly informative signals. Using these signals to partition KG clusters into accuracy-homogeneous strata and applying stratified TWCS yields 11–54% cost reductions over SotA baselines while preserving statistical reliability. To scale further, we employ distillation: efficient student models trained on 0.25% of KG triples match teacher-level stratification quality at negligible cost, enabling large-scale KG evaluation.

Our approach has three principles. First, soft majority voting across multiple LLMs reduces individual hallucinations and yields robust consensus signals. Second, limiting LLMs to a *signaling role* leverages their pattern-detection strengths while avoiding their factual weaknesses. Third, combining LLM signals with human annotation preserves ground-truth reliability. Across six KGs spanning 20M+ triples, this design consistently outperforms topology-based and unstratified SotA. Overall, our results show that LLMs can play a novel and effective role in KG quality estimation – not as annotators, but as stratification signals for efficient, reliable estimation.

Beyond immediate gains, our findings also show that there is significant potential for improvement before approaching oracle-level performance, motivating several directions for future research. The dependence of stratification quality on LLM signal fidelity motivates adaptive distillation: continually updating student models as KGs evolve to counter semantic drift. The static nature of current distilled models also suggests incremental learning to reduce retraining costs as new domains or entity types appear. More broadly, the interaction between KG evaluation and construction warrants study: stratification signals could guide targeted fact repair, turning quality assessment into an active improvement process. Extending stratification to ontology-adherent KGs could further exploit richer semantic structure for stronger signals. Overall, the approach is a foundation for continued innovation in efficient, large-scale KG evaluation – essential to maintain KGs credibility and utility in downstream applications [51]. Finally, although our evaluation framework focuses on KG accuracy, the core idea of using cheap, noisy model predictions as proxy stratification signals naturally extends to other data quality tasks – such as completeness estimation, constraint validation, and entity resolution evaluation – thereby positioning our LLM-based stratification approach as a general variance-reduction paradigm for statistical data quality evaluation pipelines beyond KGs.

REFERENCES

[1] O. Alonso. 2019. *The Practice of Crowdsourcing*. Morgan & Claypool Publishers. https://doi.org/10.2200/S00904ED1V01Y201903ICR066

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007 (LNCS)*, Vol. 4825. Springer, 722–735. https://doi.org/10.1007/978-3-540-76298-0_52

[3] J. O. Berger. 1985. *Statistical Decision Theory and Bayesian Analysis, 2nd Edition*. Springer. https://doi.org/10.1007/978-1-4757-4286-2

[4] A. Bonifati, G. H. L. Fletcher, H. Voigt, and N. Yakovets. 2018. *Querying Graphs*. Morgan & Claypool Publishers. https://doi.org/10.2200/S00873ED1V01Y201808DTM051

[5] L. D. Brown, T. T. Cai, and A. DasGupta. 2001. Interval Estimation for a Binomial Proportion. *Statist. Sci.* 16, 2 (2001), 101–117. http://www.jstor.org/stable/2676784

[6] C. Bucila, R. Caruana, and A. Niculescu-Mizil. 2006. Model Compression. In *Proc. of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*. ACM, 535–541. https://doi.org/10.1145/1150402.1150464

[7] M. G. Bulmer. 1979. *Principles of Statistics*. Dover Publications. https://books.google.it/books?id=dh24EaSrmBkC

[8] G. Casella and R. L. Berger. 2002. *Statistical Inference*. Thomson Learning. https://books.google.it/books?id=0x_vAAAAMAAJ

[9] Matteo Ceccarello, Andrea Pietracaprina, Geppino Pucci, and Eli Upfal. 2015. Space and Time Efficient Parallel Graph Decomposition, Clustering, and Diameter Approximation. In *SPAA*. ACM, 182–191.

[10] W. G. Cochran. 1977. *Sampling Techniques, 3rd Edition*. John Wiley. https://doi.org/10.1017/S0013091500025724

[11] Cohere. 2025. Command A: An Enterprise-Ready Large Language Model. *CoRR* abs/2504.00698 (2025). https://doi.org/10.48550/ARXIV.2504.00698

[12] T. Dalenius and J. L. Hodges. 1959. Minimum Variance Stratification. *J. Amer. Statist. Assoc.* 54, 285 (1959), 88–101. https://doi.org/10.1080/01621459.1959.10501501

[13] DeepSeek-AI. 2024. DeepSeek-V3 Technical Report. *CoRR* abs/2412.19437 (2024). https://doi.org/10.48550/ARXIV.2412.19437

[14] O. Deshpande, D. S. Lamba, M. Tourn, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan, and A. Doan. 2013. Building, maintaining, and using knowledge bases: a report from the trenches. In *Proc. of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*. ACM, 1209–1220. https://doi.org/10.1145/2463676.2465297

[15] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. ACL, 4171–4186. https://doi.org/10.18653/V1/N19-1423

[16] W. Edwards, H. Lindman, and L. J. Savage. 1963. Bayesian Statistical Inference for Psychological Research. *Psychol. Rev.* 70, 3 (1963), 193–242. https://doi.org/10.1037/h0044139

[17] J. Gao, X. Li, Y. E. Xu, B. Sisman, X. L. Dong, and J. Yang. 2019. Efficient Knowledge Graph Accuracy Evaluation. *Proc. VLDB Endow.* 12, 11 (2019), 1679–1691. https://doi.org/10.14778/3342263.3342642

[18] D. Gerber, D. Esteves, J. Lehmann, L. Bühmann, R. Usbeck, A. C. Ngonga Ngomo, and R. Speck. 2015. DeFacto - Temporal and multilingual Deep Fact Validation. *J. Web Semant.* 35 (2015), 85–101. https://doi.org/10.1016/j.websem.2015.08.001

[19] G. E. Hinton, O. Vinyals, and J. Dean. 2015. Distilling the Knowledge in a Neural Network. *CoRR* abs/1503.02531 (2015). http://arxiv.org/abs/1503.02531

[20] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Intell.* 194 (2013), 28–61. https://doi.org/10.1016/j.artint.2012.06.001

[21] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. Labra Gayo, R. Navigli, S. Neumaier, A. C. Ngonga Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. 2021. *Knowledge Graphs*. Morgan & Claypool Publishers. https://doi.org/10.2200/S01125ED1V01Y202109DSK022

[22] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu. 2025. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.* 43, 2 (2025), 42:1–42:55. https://doi.org/10.1145/3703155

[23] N. Q. V. Hung, T. T. Nguyen, N. T. Lam, and K. Aberer. 2013. An Evaluation of Aggregation Techniques in Crowdsourcing. In *Proc. of WISE 2013 (LNCS)*, Vol. 8181. Springer, 1–15. https://doi.org/10.1007/978-3-642-41154-0_1

[24] I. F. Ilyas, J. Lacerda, Y. Li, U. F. Minhas, A. Mousavi, J. Pound, T. Rekatsinas, and C. Sumanth. 2023. Growing and Serving Large Open-domain Knowledge Graphs. In *Companion of the 2023 International Conference on Management of Data, SIGMOD/PODS 2023, Seattle, WA, USA, June 18-23, 2023*. ACM, 253–259. https://doi.org/10.1145/3555041.3589672

[25] I. F. Ilyas, T. Rekatsinas, V. Konda, J. Pound, X. Qi, and M. A. Soliman. 2022. Saga: A Platform for Continuous Construction and Serving of Knowledge at Scale. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*. ACM, 2259–2272. https://doi.org/10.1145/3514221.3526049

[26] L. Kish. 1995. Methods for Design Effects. *J. Off. Stat.* 11, 1 (1995), 55. https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/methods-for-design-effects.pdf

[27] N. Lao, T. M. Mitchell, and W. W. Cohen. 2011. Random Walk Inference and Learning in A Large Scale Knowledge Base. In *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK*. ACL, 529–539. https://aclanthology.org/D11-1049/

[28] S. Marchesin and G. Silvello. 2024. Efficient and Reliable Estimation of Knowledge Graph Accuracy. *Proc. VLDB Endow.* 17, 9 (2024), 2392–2404. https://doi.org/10.14778/3665844.3665865

[29] S. Marchesin and G. Silvello. 2025. Credible Intervals for Knowledge Graph Accuracy Estimation. *Proc. ACM Manag. Data* 3, 3 (2025), 142:1–142:26. https://doi.org/10.1145/3725279

[30] S. Marchesin, G. Silvello, and O. Alonso. 2024. Utility-Oriented Knowledge Graph Accuracy Estimation with Limited Annotations: A Case Study on DBpedia. *Proc. of the 12th AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2024, Pittsburgh, Pennsylvania, USA, October 16–19, 2024* 12, 1 (2024), 105–114. https://doi.org/10.1609/hcomp.v12i1.31605

[31] S. Marchesin, G. Silvello, and O. Alonso. 2024. Veracity Estimation for Entity-Oriented Search with Knowledge Graphs. In *Proc. of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*. ACM, 1649–1659. https://doi.org/10.1145/3627673.3679561

[32] R. McElreath. 2020. *Statistical Rethinking: A Bayesian Course with Examples in R and STAN*. Chapman & Hall. https://doi.org/10.1201/9780429029608

[33] T. M. Mitchell, W. W. Cohen, E. R. Hruschka Jr., P. P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. D. Mishra, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. A. Platanios, A. Ritter, M. Samadi, B. Settles, R. C. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2018. Never-ending learning. *Commun. ACM* 61, 5 (2018), 103–115. https://doi.org/10.1145/3191513

[34] J. Mohoney, A. Pacaci, S. R. Chowdhury, A. Mousavi, I. F. Ilyas, U. F.q Minhas, J. Pound, and T. Rekatsinas. 2023. High-Throughput Vector Similarity Search in Knowledge Graphs. *Proc. ACM Manag. Data* 1, 2 (2023), 197:1–197:25. https://doi.org/10.1145/3589777

[35] P. Ojha and P. Talukdar. 2017. KGEval: Accuracy Estimation of Automatically Constructed Knowledge Graphs. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. ACL, 1741–1750. https://doi.org/10.18653/v1/d17-1183

[36] H. Paulheim. 2018. How much is a Triple? Estimating the Cost of Knowledge Graph Creation. In *Proc. of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8-12, 2018 (CEUR)*, Vol. 2180. CEUR-WS.org. https://ceur-ws.org/Vol-2180/ISWC_2018_Outrageous_Ideas_paper_10.pdf

[37] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. H. Miller. 2019. Language Models as Knowledge Bases?. In *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. ACL, 2463–2473. https://doi.org/10.18653/V1/D19-1250

[38] S. J. Press. 2002. *Subjective and Objective Bayesian Statistics: Principles, Models, and Applications*. John Wiley & Sons. https://doi.org/10.1002/9780470317105

[39] J. Pujara, E. Augustine, and L. Getoor. 2017. Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. ACL, 1751–1756. https://doi.org/10.18653/v1/d17-1184

[40] R. Reinanda, E. Meij, and M. de Rijke. 2020. Knowledge Graphs: An Information Retrieval Perspective. *Found. Trends Inf. Retr.* 14, 4 (2020), 289–444. https://doi.org/10.1561/1500000063

[41] M. Samadi, P. P. Talukdar, M. M. Veloso, and T. M. Mitchell. 2015. AskWorld: Budget-Sensitive Query Evaluation for Knowledge-on-Demand. In *Proc. of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. AAAI Press, 837–843. http://ijcai.org/Abstract/15/123

[42] F. M. Suchanek, M. Alam, T. Bonald, L. Chen, P. H. Paris, and J. Soria. 2024. YAGO 4.5: A Large and Clean Knowledge Base with a Rich Taxonomy. In *Proc. of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*. ACM, 131–140. https://doi.org/10.1145/3626772.3657876

[43] K. Sun, Y. E. Xu, H. Zha, Y. Liu, and X. L. Dong. 2024. Head-to-Tail: How Knowledgeable are Large Language Models (LLMs)? A.K.A. Will LLMs Replace Knowledge Graphs?. In *Proc. of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*

(Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024. ACL, 311–325. https://doi.org/10.18653/V1/2024.NAACL-LONG.18

[44] Llama Team. 2024. The Llama 3 Herd of Models. *CoRR* abs/2407.21783 (2024). https://doi.org/10.48550/ARXIV.2407.21783

[45] Mistral AI Team. 2024. Mistral NeMo. https://mistral.ai/news/mistral-nemo

[46] D. Vrandecic and M. Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85. https://doi.org/10.1145/2629489

[47] S. Wallis. 2013. Binomial Confidence Intervals and Contingency Tests: Mathematical Fundamentals and the Evaluation of Alternative Methods. *J. Quant. Linguistics* 20, 3 (2013), 178–208. https://doi.org/10.1080/09296174.2013.799918

[48] C. Wang, X. Liu, and D. Song. 2020. Language Models are Open Knowledge Graphs. *CoRR* abs/2010.11967 (2020). https://arxiv.org/abs/2010.11967

[49] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus. 2022. Emergent Abilities of Large Language Models. *Trans. Mach. Learn. Res.* 2022 (2022). https://openreview.net/forum?id=yzkSU5zdwD

[50] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou. 2024. A Survey on Knowledge Distillation of Large Language Models. *CoRR* abs/2402.13116 (2024). https://doi.org/10.48550/ARXIV.2402.13116

[51] B. Xue and L. Zou. 2023. Knowledge Graph Quality Management: A Comprehensive Survey. *IEEE Trans. Knowl. Data Eng.* 35, 5 (2023), 4969–4988. https://doi.org/10.1109/TKDE.2022.3150080

[52] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. 2016. Quality assessment for Linked Data: A Survey. *Semantic Web* 7, 1 (2016), 63–93. https://doi.org/10.3233/SW-150175