QUANTIFYING AND AUDITING LLM EVALUATION VIA POSITIVE–UNLABELED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) are increasingly used as judges for scalable evaluation, yet such LLM-as-a-Judge systems exhibit systematic biases that are decoupled from semantic quality, most notably verbosity bias. Meanwhile, human supervision is costly and typically selective, yielding reliable positive judgments but leaving most outputs unlabelled and potentially mixed in quality. We formulate LLM evaluation under selective human supervision as a positive–unlabelled learning problem and propose a geometric auditing framework based on Partial Optimal Transport. By aligning a small set of human-verified positives with a reliable subset of unlabelled outputs in a fixed embedding space, our method identifies human-consistent preferences and corrects biased judges without retraining. Experiments on simulated data and Chatbot Arena show that the method can improve judge–human agreement with few verified comparisons. For each unlabelled comparison, it also returns a normalized transport-alignment score that measures relative alignment with the verified-positive set; this score is not a calibrated probability.

1 INTRODUCTION

Reliable evaluation is a foundational requirement for deploying Large Language Models (LLMs) in real-world settings (Liang et al., 2022; Gu et al., 2024). In many applications, LLM outputs cannot be judged as simply correct or incorrect. Instead, quality depends on multiple factors, such as correctness, completeness, and clarity, and often varies across tasks and contexts (Ribeiro et al., 2020). As a consequence, modern evaluation pipelines struggle to reduce performance to a single objective score. Because of this ambiguity, high-quality evaluation still leans on human supervision. Yet human annotation is costly, and therefore difficult to scale to the volume of generations produced in practice (Dong et al., 2022; Liu et al., 2023). Human review typically covers only a small slice of model outputs, creating a persistent bottleneck. To relieve this bottleneck, automated evaluation using an LLM judge has quickly become a dominant paradigm (Verga et al., 2024). Across widely used benchmarks, LLM judges can correlate with human preferences, offering an attractive trade-off between scalability and fidelity (Zheng et al., 2023; Chiang et al., 2024). Accordingly, LLM judges are now routinely embedded as core components of evaluation and model-selection pipelines.

Despite this practical success, a critical concern remains: LLM judges exhibit systematic biases that are decoupled from underlying semantic quality (Ye et al., 2024). Rather than random noise, such biases reflect stable preferences for superficial input features. Recent work demonstrates this behavior through controlled perturbations and bias attacks, showing that minor variations in presentation or style can reliably alter judgments without changing content (Wang et al., 2024; Wu & Aji, 2025; He & Xing, 2025). Consequently, these biases induce predictable evaluation distortions, such as favoring longer or more stylized responses even when content quality remains constant.

The challenge is further compounded by the manner in which human supervision is collected. Due to limited annotation budgets, human review is typically selective rather than exhaustive (Liang et al., 2022; Dubois et al., 2023). Annotations typically focus on fluent or high-impact outputs, while weaker or ambiguous generations are often left unreviewed. This creates a regime in which negative labels are structurally missing (Bekker & Davis, 2020; Bekker et al., 2019). Human-verified examples can be regarded as reliable positives, whereas the unverified pool constitutes a heterogeneous mixture of both latent high-quality and low-quality responses. Under such selective supervision, standard

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

supervised learning assumptions break down, and naively treating all unlabeled data as negative introduces severe bias (Du Plessis et al., 2014; Kato et al., 2019).

This regime is the positive–unlabelled (PU) setting (Elkan & Noto, 2008; Bekker & Davis, 2020): a judge decision that a reviewer checks and confirms is a verified positive, and the unchecked decisions form an unlabelled pool that mixes correct judgments with judge errors. Auditing the judge then amounts to deciding which unlabelled decisions resemble the verified ones.

We propose *Positive-Unlabelled Learning Auditing for LLM-as-Judge* (PUAUDIT), which audits judge decisions in this PU setting. Given fixed embeddings of the compared responses, PUAUDIT solves a single partial optimal transport (POT) problem that moves part of the unlabelled pool onto the verified positives and leaves the rest unmatched (Chapel et al., 2020); decisions that receive little mass are flipped. The audit trains no model, uses the verified agreements as its only labelled examples, and runs in seconds on a CPU.

Our contributions are summarized as follows:

- **Problem formulation and auditing viewpoint.** We formalize LLM evaluation under scarce and selective human supervision, demonstrate why standard supervised assumptions fail in this setting, and recast evaluation bias as a PU auditing problem.
- **A lightweight geometric auditing method.** We introduce a POT-based auditing approach that operates on fixed embeddings to identify latent human-consistent preferences from unlabelled data, requiring no retraining or modification of the LLM judge.
- **Empirical effectiveness and efficiency.** In controlled simulation, PUAUDIT raises the accuracy of a corrupted judge by 17–25 percentage points. Across all nine Chatbot Arena configurations at 5% verification, it improves judge–human agreement by 0.4–34.5 points (18.9 points on average). On MT-Bench at the same budget, it exceeds the strongest positive–unlabelled baseline but falls 2.7 points below the original judge, showing that the method does not uniformly improve a strong judge; it trains no model.

2 RELATED WORKS

Automated Evaluation Methodologies. Current automated evaluation approaches largely fall into two categories: reference-based metrics and reference-free model-based evaluation. Traditional metrics like BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) rely on lexical overlap with ground truth, which often correlates poorly with human judgment on open-ended generation tasks (Liu et al., 2016; Celikyilmaz et al., 2020; Zhang et al., 2019). Consequently, recent research has shifted toward *LLM-as-a-Judge*, where strong models (e.g., GPT-4) serve as evaluators (Liu et al., 2023; Zheng et al., 2023; Gilardi et al., 2023). Approaches typically employ either single-score grading or pairwise comparison protocols (Chiang et al., 2024; Dubois et al., 2023; Kim et al., 2023; 2024) to approximate human preference. While these frameworks provide scalability, they treat the judge as a fixed oracle. In contrast, our work treats the judge as a biased instrument requiring external auditing.

Bias Analysis and Calibration in LLMs. Prior work has extensively documented systematic artifacts in LLM judges. Common failures include position bias, where judges favor the first response (Wang et al., 2024; Zheng et al., 2023); verbosity bias, where longer but vacuous responses are preferred (Saito et al., 2023; Singhal et al., 2023); and self-preference, where models favor their own outputs (Panickssery et al., 2024). To address these issues, standard practices include swapping candidate positions to average out order effects (Wang et al., 2024; Zheng et al., 2023), employing majority voting or juries to reduce variance (Verga et al., 2024; Cohen et al., 2023), or using calibration prompts to enforce neutrality (Zhu et al., 2023). While such methods mitigate surface-level inconsistencies, they overlook the fundamental distributional shift arising from selective supervision. In contrast, we propose a geometric intervention based on OT to directly align judge outputs with the latent valid distribution.

PU Learning under Selection Bias. PU learning trains a classifier from labelled positives and an unlabelled mixture of both classes (Elkan & Noto, 2008; Du Plessis et al., 2014; Kiryo et al., 2017; Bekker & Davis, 2020), either by reweighting the unlabelled pool with an estimated class prior (Elkan & Noto, 2008) or by correcting the empirical risk (Du Plessis et al., 2014; Kiryo et al., 2017). Most methods assume that positives are selected completely at random; instance-dependent selection relaxes this assumption (Bekker et al., 2019; Kato et al., 2019; Zamzam et al., 2025), and

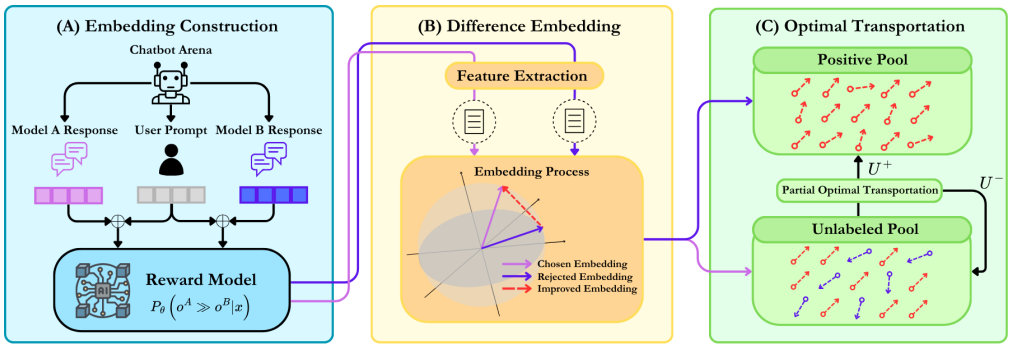

Figure 1: Overview of proposed PUAUDIT framework; (A) Embedding construction, where pairwise comparison from datasets like Chatbot Arena are processed via a reward model to derive score embedding; (B) Difference Embedding, which extracts feature representations to construct difference vectors that capture the specific direction of quality improvement between winner and loser response; (C) Optimal Transportation, where POT aligns a clean positive pool of verified improvements with the unlabeled pool.

partial optimal transport gives a geometric alternative (Chapel et al., 2020). PU learning has also been applied to LLMs, for detecting short machine-generated texts (Tian et al., 2024), distilling preference alignment (Kou et al., 2026), and in-context classification (Liu et al., 2026). These methods train a new model, whereas PUAUDIT audits the decisions of an existing, frozen judge and trains nothing.

3 METHODOLOGY

3.1 PROBLEM FORMULATION AND NOTATION

Figure 1 gives an overview of PUAUDIT, and Algorithm 1 in Appendix B.8 lists its steps. We study the problem of evaluating LLM outputs where reliable human annotations are scarce. In practice, human preferences are selectively available and concentrate on a small subset of samples, while the remaining data are unlabelled and potentially mixed in quality. We formulate this as a PU learning task as follows. Each data pair consists of a user prompt $x_i \in \mathcal{X}$, where $\mathcal{X}$ denotes the prompt space, together with two candidate response sequences o_i^A and o_i^B generated by two models, LLM_A and LLM_B, respectively. Each pair is associated with preference labels provided by an LLM judge, $\zeta_i^{(\mathrm{LLM})}$, and, when available, a human judge, $\zeta_i^{(H)}$, where $\zeta \in \{\mathrm{A}, \mathrm{B}, \mathrm{Tie}\}$ indicates whether the response from LLM_A is preferred to that from LLM_B, or whether the two responses are judged as a tie. In this work, we focus on samples for which a strict preference exists, and exclude ties, as the directional difference embedding requires an ordered winner–loser pair.

Because human preferences are observed only for a subset of samples, we partition the data into an annotated subset $\mathcal{H} = \{(x_i, o_i^A, o_i^B, \zeta_i^{(H)}, \zeta_i^{(\mathrm{LLM})})\}$, which carries both human and LLM-judge preferences, and an unannotated subset $\mathcal{H}' = \{(x_i, o_i^A, o_i^B, \zeta_i^{(\mathrm{LLM})})\}$, which carries only the LLM-judge preference. So that the representation does not depend on whether a human label exists, every comparison is oriented by the original LLM-judge decision, and o_i^w and o_i^ℓ denote the responses preferred and dispreferred by the judge. For an annotated comparison, the human-consistency indicator $y_i = \mathbb{I}\{\zeta_i^{(\mathrm{LLM})} = \zeta_i^{(H)}\}$ equals one when the judge agrees with the human. The verified-positive set is $\mathcal{D}_P = \{(x_i, o_i^w, o_i^\ell, y_i) : i \in \mathcal{H},\ y_i = 1\}$ and the unlabelled set is $\mathcal{D}_U = \{(x_i, o_i^w, o_i^\ell) : i \in \mathcal{H}'\}$, whose labels y_i are unobserved and can be either zero or one. The main notation is summarized in Table 3 in Appendix B.1.

Our goal is to construct a scoring function $f : \mathbb{R}^d \to [0, 1]$ that quantifies the degree to which an LLM-oriented comparison is consistent with human preference from its difference representation z_i defined in Eq. (1). For unlabelled samples, this corresponds to estimating the latent human-consistency signal associated with $y_i = 1$ by aligning their representations with the verified-positive reference set $\mathcal{D}_P$.

Before formally introducing our method, we define the notation used throughout the paper. We denote the ℓ_2-norm by $\|\cdot\|$, the cardinality of a set by $|\cdot|$, the inner product by $\langle\cdot,\cdot\rangle$, and the similarity function by $d(\cdot,\cdot)$, which we take to be cosine similarity throughout; $\lfloor\cdot\rfloor$ denotes the floor function.

3.2 Feature construction and preprocessing

Fixed representation space. Our auditing procedure operates entirely in a fixed representation space. Given a prompt x and a response o, we use the pretrained encoder F_θ with fixed parameters to extract a prompt–response representation $F_\theta(x,o) \in \mathbb{R}^d$. Importantly, F_θ is never trained or fine-tuned in our pipeline; all PU relabeling and auditing steps are performed on top of these fixed embeddings. In our main experiments, we instantiate F_θ with a reward-model encoder, since reward models are naturally trained on preference data and provide strong representations for response comparison. The encoder is modular, but its effectiveness depends on how well it separates preferred from dispreferred responses: Table 4 in Appendix B.1 shows that reward-model encoders such as Skywork-RM separate them more clearly than general-purpose encoders such as Gemma-300M, so we use Skywork-RM throughout.

For each pairwise comparison, let o^w and o^ℓ denote the response preferred and dispreferred by the initial judge. We represent the comparison by the normalized difference embedding

$$z_i = \frac{F_\theta(x_i, o_i^w) - F_\theta(x_i, o_i^\ell)}{\|F_\theta(x_i, o_i^w) - F_\theta(x_i, o_i^\ell)\|} \in \mathbb{S}^{d-1}. \tag{1}$$

This representation focuses on the direction of relative improvement rather than the absolute embedding of either response. It is especially useful when both candidate responses are fluent and semantically close, because the differenced representation directly encodes how the judge separates the preferred response from its counterpart.

Positive group denoising. Although human-verified positives provide valuable supervision, they may still contain annotation noise or geometrically inconsistent examples. We therefore apply a lightweight two-stage filter before POT to remove geometric outliers, rather than altering consistent human preference patterns. To avoid conflating semantic topics, this filtering is performed within each question type rather than over the pooled dataset. First, within each question type, we retain positive examples whose embeddings are closest to the corresponding positive-group center. Second, among the retained examples, we further keep comparisons whose improvement directions z_i are most aligned with the average positive improvement direction within the same question type. This produces a cleaner $\mathcal{D}'_P$ that is more geometrically coherent while reducing the risk of discarding valid positives from minority topics. The full method is provided in Appendix A.4.

3.3 Partial optimal transport and relabelling

We align the unlabelled set $\mathcal{D}_U$ to the denoised positive reference set $\mathcal{D}'_P$ using POT. Standard OT transports all mass, so in our setting it would also align the negative instances on which the LLM judge is wrong. Instead, POT allows only a fraction m of the total mass from $\mathcal{D}_U$ to be transported, while the remaining mass is left unmatched. This design reflects that $\mathcal{D}_U$ contains a mixture of positive and negative samples, whereas $\mathcal{D}'_P$ serves as a clean anchor distribution. Let $\mu'_P = \sum_{i=1}^{n'_P} a_i \delta_{z_i}$ and $\mu_U = \sum_{j=1}^{n_U} b_j \delta_{z_j}$ denote the empirical measures of the denoised positive and unlabelled sets, where $n'_P = |\mathcal{D}'_P|$, $n_U = |\mathcal{D}_U|$, $a_i = 1/n'_P$, $b_j = 1/n_U$, and δ_z is the Dirac measure at z. With the cosine cost $C_{ij} = 1 - d(z_i, z_j)$, we seek the transport plan that moves a total mass $m \in (0,1]$ at minimal cost,

$$\Gamma^\star = \arg\min_{\Gamma \in \mathbb{R}_{\geq 0}^{n'_P \times n_U}} \sum_{i=1}^{n'_P}\sum_{j=1}^{n_U} C_{ij}\Gamma_{ij} \quad \text{s.t.} \quad \Gamma \mathbf{1}_{n_U} \leq \mathbf{a}, \quad \Gamma^\top \mathbf{1}_{n'_P} \leq \mathbf{b}, \quad \mathbf{1}_{n'_P}^\top \Gamma \mathbf{1}_{n_U} = m.$$

The inequality constraints allow part of the mass to remain unmatched, avoiding the alignment of geometrically distant negatives in $\mathcal{D}_U$ with the positive anchor distribution $\mathcal{D}'_P$ (Chapel et al., 2020). For each unlabelled sample j, the alignment score $s_j = \sum_{i=1}^{n'_P} \Gamma^\star_{ij}$ and its normalized version $\bar{s}_j = s_j / \max_k s_k$ measure how strongly the sample aligns with the verified-positive manifold

represented by $\mathcal{D}'_P$. A sample with $\bar{s}_j < \eta$ is weakly aligned, suggesting that the judge preference may be affected by bias, so we flip it: the corrected preference is $\hat{\zeta}_j^{(\mathrm{LLM})} = \mathrm{Flip}(\zeta_j^{(\mathrm{LLM})})$ if $\bar{s}_j < \eta$ and $\hat{\zeta}_j^{(\mathrm{LLM})} = \zeta_j^{(\mathrm{LLM})}$ otherwise, where $\mathrm{Flip}(\cdot)$ returns the opposite preference and the threshold η is fixed to 0.5 in all experiments. This step targets cases where biases, such as verbosity bias, lead the judge to prefer responses that are geometrically distant from the verified-positive manifold.

Computational cost. The audit trains nothing: it solves one linear program over the $n'_P \times n_U$ cost matrix with the network-simplex solver of the POT library (Flamary et al., 2021).

4 Theoretical Guarantee

To characterize our geometric auditing procedure, we analyze when POT preferentially assigns transported mass to latent human-consistent samples in the unlabelled pool. Under geometric separation conditions, we show that the support of an optimal partial transport plan can be restricted to the latent positive component. We further analyze the sensitivity of the empirical cosine transport cost to nuisance perturbations.

4.1 Geometric setup

We consider the normalized deviation space on the unit hypersphere $\mathbb{S}^{d-1}$, where each point z represents a normalized difference between preferred and dispreferred response embeddings defined in Eq. (1). We assume that the unlabelled set follows the mixture $\mu_U = \alpha\mu_+ + (1-\alpha)\mu_-$, where μ_+ and μ_- are the latent distributions of human-consistent and human-inconsistent comparisons and $\alpha \in (0,1)$ is the latent positive proportion. Because verification may be selective, $\mathcal{D}_P$ need not be an unbiased sample from μ_+; we only require $\mathrm{supp}(\mu_P) \subseteq \mathrm{supp}(\mu_+)$, where μ_P is the empirical measure of $\mathcal{D}_P$.

In our proposal, the human-preferred improvements are not randomly distributed but share common semantic directions, whereas low-quality deviations are more isotropic or clustered differently. To analyze the behaviour of POT on these embeddings, we introduce the following geometric assumptions grounded in the manifold hypothesis of representation learning (Wang & Isola, 2020).

Assumption 4.1 (Directional Concentration of Improvements)**.** The positive measure μ_+ is supported on a cone $\mathcal{C}_\tau$ centred at a consensus improvement direction $\bar{z} \in \mathbb{S}^{d-1}$ with concentration parameter $\tau \in (0,1)$ such that $\mathrm{supp}(\mu_+) \subseteq \mathcal{C}_\tau(\bar{z}) := \{z \in \mathbb{S}^{d-1} : d(z,\bar{z}) \geq \tau\}$.

Assumption 4.2 (Separability of Noise)**.** Let $\tau \in (0,1)$ be the concentration parameter of the improvement cone $\mathcal{C}_\tau(\bar{z})$. There exists a safety margin $\delta > 0$ such that for any negative deviation $z^- \in \mathrm{supp}(\mu_-)$, the projection onto the consensus direction satisfies $\langle z^-, \bar{z}\rangle < \tau - \delta$. We further require that δ is sufficiently large such that $\tau^2 > \tau - \delta$ to ensure a non-empty utility gap.

This assumption is justified by the improvement-alignment step in the denoising phase, which filters samples based on their cosine alignment and encourages $\mathcal{D}_P$ to concentrate around a consistent direction. The contrastive learning theories also prove that the semantically similar pairs align on the hypersphere (Wang & Isola, 2020). In addition, the margin δ ensures that the positive support forms a disjoint region in the embedding space which is distinguishable via cosine similarity.

4.2 Algorithm guarantee: recovery and robustness

We now formalize how PUAudit exploits the latent quality structure of these datasets. To convert the abstract auditing problem into a tractable geometric task, we utilize the latent representations of a reward model to construct difference embeddings. By encoding the vector difference, we transform discrete comparison pairs into a normalized manifold that explicitly captures the gradient of quality improvement, thereby facilitating the subsequent alignment via POT.

Theorem 4.3 (High-probability positive support of POT)**.** *Suppose Assumptions 4.1 and 4.2 hold, together with the high-dimensional orthogonal-component condition stated in Lemma A.1. Consider the POT problem between μ'_P and μ_U with cosine cost $C_{ij} = 1 - d(z_i, z_j)$ and transported mass $m \leq \alpha$, where $\alpha = |U_+|/n_U$ is the empirical latent-positive mass of the unlabelled pool. If the positive–negative similarity gap derived in Appendix A is strictly positive, then, with the probability*

Table 1: Simulation results over five independent runs. Each run contains 1,200 pairwise comparisons, of which about 183 are human-verified; accuracy is measured on the remaining comparisons. The adjusted accuracy is reported as mean (max–min range) across runs; Diff. is the adjusted minus the original accuracy, and the flip rate is the fraction of unverified judgments that PUAUDIT reverses.

Noise	Verification	Orig. Acc.	Adj. Acc.	Diff.	Flip Rate
CD	SCAR	69.52%	94.46% (3.23)	**+24.94%**	29.69%
CD	SAR	69.66%	95.14% (2.40)	**+25.48%**	30.50%
DD	SCAR	69.48%	94.64% (5.19)	**+25.15%**	30.10%
DD	SAR	71.09%	88.39% (6.84)	**+17.30%**	39.15%

specified in Appendix A.1, every optimal transport plan $\Gamma^\star$ assigns the transported mass entirely to the latent positive component:

$$\mathrm{supp}(\Gamma^\star) \subseteq \mathrm{supp}(\mu'_P) \times \mathrm{supp}(\mu_+).$$

Hence every unlabelled sample that receives transported mass lies in the latent-positive region; if $m < \alpha$, a sample with zero transported mass may still be positive.

The detailed proof can be found in Appendix A.1. The key challenge in LLM-as-a-Judge is susceptibility to superficial biases (e.g., verbosity), which we model as nuisance variables. We formally assume these biases are distinct from quality.

Assumption 4.4 (Approximate nuisance orthogonality)**.** Let $\bar{z} \in \mathbb{S}^{d-1}$ denote the consensus improvement direction and let $z_{\text{bias}} \in \mathbb{S}^{d-1}$ denote the direction induced by a presentation perturbation. We assume that $|\langle \bar{z}, z_{\text{bias}} \rangle| \leq \epsilon_{\text{bias}}$ for a small constant $\epsilon_{\text{bias}} \geq 0$.

Theorem 4.5 (Stability under perturbation)**.** *Given the Assumption 4.4, when evaluating the pairwise transport cost between an empirical anchor $z_i \in \mathcal{D}'_P$ and a perturbed embedding $\tilde{z}$, the similarity introduces a first-order sensitivity due to the natural deviation of z_i from $\bar{z}$. However, because of selective anchors that are highly aligned with $\bar{z}$, the magnitude of the orthogonal component $\|z_i^\perp\|$ is bounded by a small threshold ϵ_p. Thus, the transport cost is invariant to the bias up to $\mathcal{O}(\lambda \cdot \epsilon_p) + \mathcal{O}(\lambda^2)$. The first-order term is therefore controlled by the denoising phase.*

The detailed proof is provided in Appendix A.2, and Figure 6 in Appendix B.1 illustrates the bound. The result explains why a nuisance perturbation of the difference embedding changes the transport cost only weakly. While a linear classifier might be unstable by the magnitude of λz_{bias}, cosine similarity projects this noise onto the tangent, where its effect vanishes. Additionally, the cosine similarity is involved in the later POT that the ignorable bias would not affect its robustness.

5 EXPERIMENTS

5.1 SIMULATION

Each simulated comparison has a latent human preference, drawn from two Gaussian classes whose centres lie four units apart in a two-dimensional latent space, and a judge preference obtained by corrupting 30% of the human preferences. As in the real pipeline, the difference embedding is oriented by the judge's choice, so the judge's agreements and errors occupy opposite regions of a 64-dimensional embedding space. We cross two corruption mechanisms with two verification mechanisms. Under class-dependent (CD) noise every comparison is corrupted with the same probability, whereas under distribution-dependent (DD) noise the errors concentrate near the class boundary. Under SCAR, 15% of the comparisons are verified uniformly at random; under SAR, verification favours comparisons near the boundary. Appendix B.3 gives the full design.

Table 1 shows that PUAUDIT raises accuracy from about 70% to 94–95% under CD noise and under DD noise with SCAR verification, a gain of about 25 points. DD+SAR is the hardest setting, because the judge's errors and the verified comparisons both concentrate near the class boundary, and the gain there is 17.3 points. Figure 2 shows the mechanism, and Figure 12 shows the other settings: judge errors lie outside the region covered by the verified agreements, receive no mass, and are flipped. The sweeps in Appendix B.4 cover all four settings: the gain is positive once the class centres are two

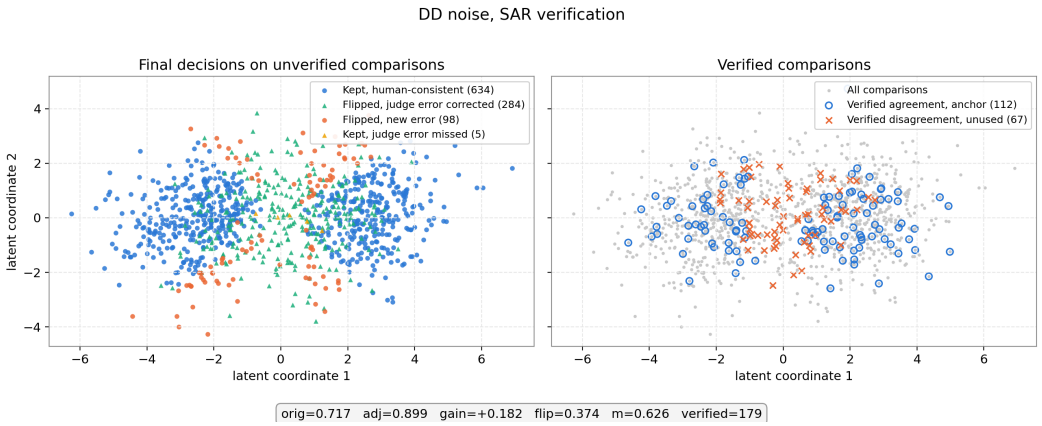

Figure 2: DD+SAR simulation (seed 1). **Left**: decisions of PUAUDIT on the unverified comparisons, split by whether each judgment agrees with the latent human preference. **Right**: the verified comparisons; only the verified agreements serve as transport anchors.

units apart, remains 16 to 22 points with only 2% of the comparisons verified, and grows with the judge's corruption rate.

5.2 REAL-DATA EVALUATION

Setup. We use pairwise comparisons with human preference labels from Chatbot Arena (Zheng et al., 2023; Chiang et al., 2024) and MT-Bench (Zheng et al., 2023); Figure 4 in Appendix B.1 shows one. Two locally deployed judges, MISTRAL-7B-INSTRUCT (Jiang et al., 2023) and LLEMMA-7B-MUINSTRUCT (Azerbayev et al., 2023), judge Chatbot Arena comparisons of six question types, and GPT-5.4-MINI (OpenAI, 2026) judges MT-Bench. For each judge and question type, a random, prompt-disjoint 5%, 10%, 20%, 50%, or 80% of the comparisons is human-verified, and both original-judge and adjusted accuracies are measured on the same remaining comparisons in each of ten splits. PUAUDIT keeps its hyperparameters fixed ($\eta = 0.5$, $\alpha_1 = \alpha_2 = 0.7$, and m equal to the verified agreement rate). The main baselines are four PU methods that receive the same information as PUAUDIT, the verified agreements: Elkan–Noto (Elkan & Noto, 2008), nnPU (Kiryo et al., 2017), SAR-PU (Bekker et al., 2019), and POT-PU (Chapel et al., 2020). Supervised classifiers are additionally fitted on verified disagreements, which selective review does not provide, so we report them only as a reference. Every method uses the same splits and the same fixed Skywork embeddings. Appendix B.5 gives the details.

Comparison with PU baselines and the original judge. On the same unverified test comparisons, PUAUDIT exceeds the strongest PU baseline in 8 of 10 Arena/MT-Bench configurations at 5% verification (mean margin 8.2 points) and in all 10 at 10% (4.5 points; Table 5). This baseline comparison does not imply an improvement over the original judge. Across the nine Arena configurations, the 5% changes range from +0.4 to +34.5 points (mean +18.9); on MT-Bench, accuracy instead falls from 77.9% to 75.2% at 5% verification and from 78.0% to 75.3% at 10%. At 5% on MT-Bench, averaged over ten splits, PUAUDIT corrects about 263 originally inconsistent comparisons but also flips about 329 originally consistent ones, giving a net loss of 2.7 points. This error decomposition identifies false flips as the immediate cause. At 20% verification, the held-out test positive proportion is the original judge accuracy, 0.777, and the default agreement-rate mass $m = 0.78$ essentially matches it. Yet this setting yields 75.7%, while $m = 0.9$ yields 80.5% (Table 9). Thus the best empirical transport budget need not equal the test positive proportion: because the thresholded alignment score makes both false flips and missed corrections, extra transport slack can improve net accuracy. The $m = 0.9$ result is an empirical operating point, not an estimate of a 90% positive prevalence, and it lies outside the theorem guarantee that requires $m \leq \alpha$.

Reference comparison with supervised methods. Supervised methods observe verified disagreements, so a comparison with them is not like-for-like and serves only as a reference. Table 2 lists all 17 methods on Llemma-7B QTC: PUAUDIT exceeds the strongest PU baseline by 16.8 points and ties

Table 2: All baselines on the configuration where PUAUDIT performs best (Llemma-7B, Chatbot Arena QTC, $n = 1297$, 5% verified; ten splits). Most rows are not directly comparable with PUAUDIT: the controls and the plain reward head use no human label, the reward head replaces the audited judge with an external preference model, and the supervised methods are trained on verified disagreements, which selective human review does not provide. Only the P group receives the same information as PUAUDIT. Accuracy is the mean (max–min range) on the unverified comparisons; Human labels counts the labelled comparisons used for fitting; Trained indicates whether the method fits parameters to the labels. Bold values are PUAUDIT; green cells are the best in each group.

Method	Adj. Acc.	Gain	Human labels	Trained	CPU time (s)
No human label					
Original LLM judge	57.1% (0.9)	+0.0%	0	no	0.00
Random flip	51.8% (5.9)	-5.3%	0	no	0.00
Always flip	42.9% (0.9)	-14.2%	0	no	0.00
Reward head (replaces the judge)	75.3% (0.7)	+18.2%	0	no	0.00
Verified agreements only (P)					
Elkan–Noto (Elkan & Noto, 2008)	52.6% (17.9)	-4.5%	39	yes	0.03
nnPU (Kiryo et al., 2017)	57.1% (0.9)	+0.0%	39	yes	0.56
SAR-PU (Bekker et al., 2019)	52.6% (13.6)	-4.4%	39	yes	0.03
POT-PU (Chapel et al., 2020)	58.7% (6.3)	+1.6%	39	yes	0.04
One-class SVM	57.1% (0.9)	-0.0%	39	yes	0.03
PUAUDIT (ours)	**75.5% (3.2)**	**+18.4%**	**39**	**no**	**0.09**
Verified agreements and disagreements (P+N)					
Logistic regression	74.7% (6.7)	+17.6%	65	yes	0.03
Random forest	69.1% (16.8)	+12.0%	65	yes	0.29
MLP	54.5% (16.3)	-2.6%	65	yes	0.04
k-NN	67.8% (11.9)	+10.7%	65	yes	0.04
Cosine centroid	74.8% (6.3)	+17.7%	65	yes	0.03
Label spreading	52.6% (26.0)	-4.5%	65	yes	0.04
Reward head + calibration	75.5% (2.4)	+18.4%	65	yes	0.00

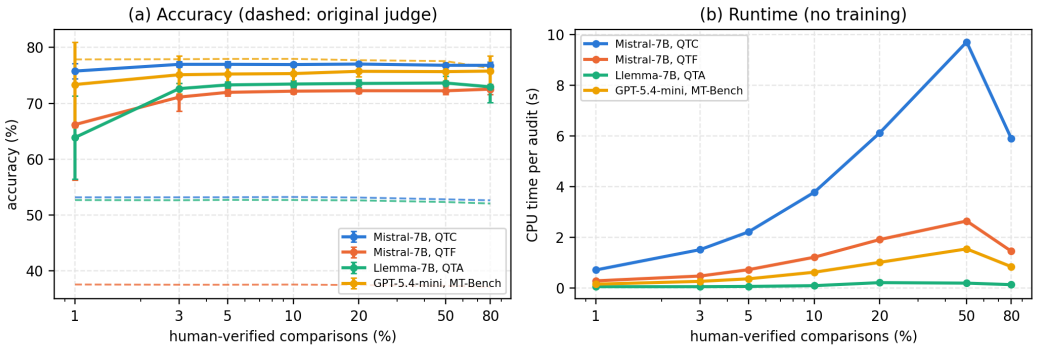

Figure 3: Label and compute efficiency. (a) Accuracy of PUAUDIT (solid, mean±sd over ten splits) and of the original judge (dashed), evaluated on the same test comparisons at each budget, as the fraction of human-verified comparisons varies from 1% to 80%. (b) CPU time of one audit, which involves no training. Full numbers are in Table 7.

with the best supervised reference, the calibrated reward head (75.5%), while fitting no parameters. When every method is fitted on the same number of labelled examples, so that PUAUDIT receives as many verified agreements as the supervised methods receive labelled comparisons, PUAUDIT comes within approximately 1.5 points of the best supervised reference on average at the 5% budget and beats it on Llemma QTA and QTC and Mistral QTE without using verified judge errors as fitting examples (Table 6); this matches fitting-set size, while PUAUDIT reviews a larger pool to obtain its verified agreements.

Label efficiency. On Mistral QTC, verifying 81 comparisons (1%) raises accuracy from 53.1% to 75.8%, close to the 77.0% reached with 5% (Figure 3), because the anchors only need to locate the direction of human-consistent improvement (Theorem A.3 in Appendix A.3). Removing partial transport returns accuracy to that of the original judge, and removing the improvement-alignment filter costs 8 to 9 points (Table 8); Appendix B.7 also reports the sensitivity to m, to the denoising thresholds, and to feature-dependent verification.

6 Discussion and future work

This work introduces a PU learning framework that leverages POT and geometric alignment to audit LLM evaluation under scarce and biased human supervision. A primary direction for future work is to remove the reliance on a specific reward model, for example with general-purpose contrastive or self-supervised encoders.

In addition, our formulation assumes a single direction of quality improvement, whereas response quality is inherently multi-dimensional. Extending the method to support multiple improvement directions or multi-manifold structures would better capture heterogeneous notions of quality and naturally connects to multi-PU or multi-class settings, such as those involving multiple LLM or human judges. Finally, future work may explore alternative distance metrics and alignment objectives beyond cosine similarity and POT, as well as more principled inference of the latent positive proportion in the unlabelled set.

Limitations. PUAudit relies on judge errors being geometrically separable from correct decisions in the fixed embedding space. The MT-Bench results show that its default transported mass and fixed alignment threshold can reduce accuracy when the original judge is already strong; adapting these choices to the judge and the verified sample is an open problem.

AI use statement

We used generative AI tools (LLM-based coding assistants) to help draft and revise the text of this paper, to generate and reformat LaTeX tables and figures from experimental result files, and to write and debug data-processing and analysis scripts (e.g., significance testing, metric computation, and job-submission scripts for the compute cluster). We did not use generative AI tools to generate the underlying research idea, the theoretical results, or the experimental design; these were developed by the authors. All AI-assisted writing, code, tables, and figures were reviewed, checked against the underlying experimental data, and edited by the authors before inclusion, and all reported numerical results were verified against the raw experiment outputs. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics statement

This work does not involve new data collection from human subjects. The real-data experiments use pairwise LLM-response comparisons and existing human preference judgments drawn from the publicly available MT-Bench and Chatbot Arena benchmarks; we do not collect, store, or process any personally identifiable information beyond what is already present in these public releases. PUAudit is intended as an auditing tool that flags LLM-judge decisions that are likely to disagree with human preference, not as a replacement for human review. We are not aware of any conflicts of interest, sponsorship-related concerns, or other ethical issues raised by this submission.

Reproducibility statement

We describe the full PUAudit procedure in Section 3, with the feature extraction and the two-stage positive-group denoising in Appendix A.4 and pseudocode in Appendix B.8. All theoretical assumptions are stated in Section 4, and complete proofs of Theorem 4.3 and Theorem 4.5, including all supporting lemmas, are given in Appendix A. The generative simulation of Section 5.1 is fully specified in Appendix B.3, including the data-generating process, the corruption and verification mechanisms, and the sweep grids. The real-data setup, including the judge models, prompt template, decoding configuration, splits, baselines, and statistical tests, is described in Appendix B.5, and the

consistency metrics are defined in Appendix A.5. All hyperparameters are fixed across datasets and reported in Section 5.2.

REFERENCES

Skywork AI. Skywork-reward-llama-3.1-8b-v0.2. https://huggingface.co/Skywork/Skywork-Reward-Llama-3.1-8B-v0.2, 2024.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: A survey. *Machine learning*, 109:719–760, 2020.

Jessa Bekker, Pieter Robberechts, and Jesse Davis. Beyond the selected completely at random assumption for learning from positive and unlabeled data. In *Joint European conference on Machine Learning and Knowledge Discovery in Databases*, pp. 71–85. Springer, 2019.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*, 2020.

Laetitia Chapel, Mokhtar Z Alaya, and Gilles Gasso. Partial optimal tranport with applications on positive-unlabeled learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2903–2913, 2020.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. In *International Conference on Machine Learning*, 2024.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. Lm vs lm: Detecting factual errors via cross examination. *arXiv preprint arXiv:2305.13281*, 2023.

Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. A survey of natural language generation. *Computing Surveys*, 55:1–38, 2022.

Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 30039–30069, 2023.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 213–220, 2008.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T. H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78): 1–8, 2021.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120:e2305016120, 2023.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

Weiyi He and Yue Xing. Impact of positional encoding: Clean and adversarial rademacher complexity for transformers under in-context regression. *arXiv preprint arXiv:2512.09275*, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. Introduces the Mistral 7B architecture and includes an instruction-tuned variant.

Masahiro Kato, Takeshi Teshima, and Junya Honda. Learning from positive and unlabeled data with a selection bias. In *International Conference on Learning Representations*, 2019.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*, 2023.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*, 2024.

Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Zhiqiang Kou, Junyang Chen, Xin-Qiang Cai, Xiaobo Xia, Ming-Kun Xie, Dong-Dong Wu, Biao Liu, Yuheng Jia, Xin Geng, Masashi Sugiyama, and Tat-Seng Chua. Positive-unlabeled reinforcement learning distillation for on-premise small models. *arXiv preprint arXiv:2601.20687*, 2026.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*, pp. 74–81, 2004.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Conference on Empirical Methods in Natural Language Processing*, pp. 2122–2132, 2016.

Siyan Liu, Yi Chang, Manli Cheng, Qinglong Tian, and Pengfei Li. In-context positive-unlabeled learning. *arXiv preprint arXiv:2605.05591*, 2026.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

OpenAI. Introducing gpt-5.4. `https://openai.com/index/introducing-gpt-5-4/`, 2026.

Arjun Panickssery, Samuel Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 68772–68802, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pp. 311–318, 2002.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*, 2020.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. Verbosity bias in preference labeling by large language models. *arXiv preprint arXiv:2310.10076*, 2023.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*, 2023.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, Qinghua Zhang, Ruifeng Li, Chao Xu, and Yunhe Wang. Multiscale positive-unlabeled detection of AI-generated texts. In *International Conference on Learning Representations*, 2024.

Hugo S. Vera et al. Embeddinggemma: Powerful and lightweight text representations. *arXiv preprint arXiv:2509.20354*, 2025.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*, 2024.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. Large language models are not fair evaluators. In *Annual Meeting of the Association for Computational Linguistics*, pp. 9440–9450, 2024.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Minghao Wu and Alham Fikri Aji. Style over substance: Evaluation biases for large language models. In *International Conference on Computational Linguistics*, pp. 297–312, 2025.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. Justice or prejudice? quantifying biases in LLM-as-a-judge. *URL https://arxiv. org/abs/2410.02736*, 2024.

Omar Zamzam, Haleh Akrami, Mahdi Soltanolkotabi, and Richard Leahy. Learning from PU data using disentangled representations. In *IEEE International Conference on Image Processing*, pp. 1624–1629, 2025.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*, 2023.

APPENDIX

A Proof **13**

A.1 Proof of theorem 4.3 13

A.2 Proof of theorem 4.5 15

A.3 Proof of insensitivity to the number of verified positives 15

A.4 Feature extraction and P-group denoising 16

A.5 Evaluation metrics 17

B Additional figures, tables, and experimental details **18**

B.1 Notation and setup 18

B.2 Data generation 21

B.3 Simulation details 21

B.4 Simulation sweeps 22

B.5 Real-data details 28

B.6 Additional real-data results 28

B.7 Ablation and sensitivity analysis 29

B.8 Algorithm 31

A PROOF

A.1 PROOF OF THEOREM 4.3

The goal of POT algorithms is to pair the limited set of verified positive samples with candidates from the unlabelled pool to maximize the total similarity. The unlabelled pool contains latent true positives and latent true negatives.

Primal Formulation. The POT problem minimizes the total cost C as $\mathcal{L}(\Gamma)$

$$\mathcal{L}(\Gamma) = \sum_{(x_i,o_i^w,o_i^\ell,y_i)\in\mathcal{D}'_P} \sum_{(x_j,o_j^w,o_j^\ell)\in\mathcal{D}_U} C_{ij}\Gamma_{ij},$$
$$\text{subject to } \Gamma\mathbf{1}_{n_U} \le \mathbf{a},\ \Gamma^\top\mathbf{1}_{n'_P} \le \mathbf{b}, \quad \sum_{(x_i,o_i^w,o_i^\ell,y_i)\in\mathcal{D}'_P} \sum_{(x_j,o_j^w,o_j^\ell)\in\mathcal{D}_U} \Gamma_{ij} = m.$$

Since $C_{ij} = 1 - d(z_i, z_j)$, the objective is equivalent to maximizing

$$\mathcal{J}(\Gamma) = \sum_{(x_i,o_i^w,o_i^\ell,y_i)\in\mathcal{D}'_P} \sum_{(x_j,o_j^w,o_j^\ell)\in\mathcal{D}_U} d(z_i, z_j)\Gamma_{ij}$$

with the same constraints.

Partitioning and Utility Bounds. Let $U = U_+ \cup U_-$, where $U_+ = U \cap \text{supp}(\mu_+)$ are the latent true positives, and $U_- = U \cap \text{supp}(\mu_-)$ are latent true negatives. Let $z_i \in \mathcal{D}'_P$ be an empirical anchor. We decompose any candidate vector $z \in \mathcal{D}_U$ into a parallel and an orthogonal component relative to the consensus direction $\overline{z}$: $z = \tau_z\overline{z} + \epsilon_z z^\perp$ where $\tau_z = \langle z, \overline{z}\rangle$, $z^\perp$ is a unit vector orthogonal to $\overline{z}$ and $\epsilon_z = \sqrt{1-\tau_z^2}$.

- ($U_+ \to P$): Consider a sample $z_j \in U_+$. By Assumption 4.1 and $\text{supp}(\mu_P) \subseteq \text{supp}(\mu_+)$, both z_i, z_j lie within the improvement cone $\mathcal{C}_\tau$, so $\tau_i, \tau_j \ge \tau$ and $\epsilon_i, \epsilon_j \le \sqrt{1-\tau^2}$. Thus,

the inner product of these two vectors satisfies

$$\langle z_i, z_j \rangle = \tau_i \tau_j + \epsilon_i \epsilon_j \langle z_i^\perp, z_j^\perp \rangle \geq \tau^2 - (1 - \tau^2)t,$$

where $t := \max_{x \in U} |\langle z_i^\perp, x^\perp \rangle|$.

- (U_- remains): Consider a target $z_k \in U_-$. By Assumption 4.2, $\tau_k = \langle z_k, \bar{z} \rangle \leq \tau - \delta$. Using the decomposition

$$z_i = \tau_i \bar{z} + \epsilon_i z_i^\perp, \qquad z_k = \tau_k \bar{z} + \epsilon_k z_k^\perp,$$

we obtain

$$\begin{aligned} \langle z_i, z_k \rangle &= \tau_i \tau_k + \epsilon_i \epsilon_k \langle z_i^\perp, z_k^\perp \rangle \\ &\leq \tau_i(\tau - \delta) + \epsilon_i \epsilon_k t \\ &\leq (\tau - \delta) + t, \end{aligned}$$

where t is defined as above and the last step uses $\tau_i \leq 1$ and $\tau - \delta \geq 0$. If $\delta > \tau$, then $\tau_i(\tau - \delta) \leq 0$ and the argument below holds with Δ_{base} replaced by τ^2.

Establishing the utility gap between a true positive versus negative that

$$\begin{aligned} \Delta = \min \langle z_i, z_j \rangle - \max \langle z_i, z_k \rangle &\geq (\tau^2 - \tau + \delta) - 2 \max_{x \in U} |\langle z_i^\perp, x^\perp \rangle| \\ &\geq (\tau^2 - \tau + \delta) - 2t \end{aligned}$$

Let the base theoretical gap be defined as $\Delta_{base} = \tau^2 - \tau + \delta$. By Assumption 4.2, we require $\delta > \tau(1-\tau)$ to ensure $\Delta_{base} > 0$. To guarantee $\Delta > 0$ we require $2t < \Delta_{base}$. Setting $t = \Delta_{base}/4$, we substitute this into the probabilistic bound established in Lemma A.1 for a fixed anchor z_i,

$$P\left(\Delta > 0\right) \geq 1 - 2n_U \exp\left(-\frac{(d-1)\Delta_{base}^2}{32}\right).$$

The support claim requires $\Delta > 0$ for every anchor. Applying Lemma A.1 to each of the n'_P anchors and taking a union bound gives

$$P\left(\Delta > 0 \text{ for all anchors in } \mathcal{D}'_P\right) \geq 1 - 2n'_P n_U \exp\left(-\frac{(d-1)\Delta_{base}^2}{32}\right).$$

On this event, every optimal plan is supported on $\mathcal{D}'_P \times U_+$. Suppose instead that an optimal plan $\Gamma^\star$ places mass $\Gamma^\star_{ik} > 0$ on some $k \in U_-$. Because the total transported mass is $m \leq |U_+|/n_U$ and part of it reaches U_-, the mass received by U_+ is strictly smaller than $|U_+|/n_U$, so some $j \in U_+$ satisfies $\sum_l \Gamma^\star_{lj} < b_j$. Moving mass $\varepsilon = \min\{\Gamma^\star_{ik}, b_j - \sum_l \Gamma^\star_{lj}\} > 0$ from (i,k) to (i,j) preserves all constraints and changes the cost by $\varepsilon(C_{ij} - C_{ik}) = -\varepsilon(\langle z_i, z_j \rangle - \langle z_i, z_k \rangle) \leq -\varepsilon\Delta < 0$, which contradicts optimality.

■

Lemma A.1. *Let $\bar{z}$ be the unit consensus direction where we define it as the centre of the positive cone $\mathcal{C}_\tau(\bar{z})$ as in Assumption 4.1 and let $z_i^\perp, x^\perp$ be unit vectors $(d-1)$-dimensional orthogonal complement subspace (Figure 5). Assuming the unlabeled samples' orthogonal components $\mathcal{X}^\perp = \{x_1^\perp, \ldots, x_{n_U}^\perp\}$ are distributed approximately uniformly on the unit sphere of the subspace, the maximum correlation for a fixed anchor $z_i^\perp$ is bounded by t with high probability:*

$$P\left(\max_{x^\perp \in \mathcal{X}^\perp} |\langle z_i^\perp, x^\perp \rangle| > t\right) \leq 2n_U \exp\left(-\frac{(d-1)t^2}{2}\right). \tag{2}$$

Proof. *For any single random unit vector $x^\perp \sim \text{Uniform}(\mathbb{S}^{d-2})$, its projection onto a fixed unit vector $z_i^\perp$ is a sub-Gaussian random variable. By the concentration of measure on the high-dimensional sphere, the tail probability is bounded by*

$$P(|\langle z_i^\perp, x^\perp \rangle| > t) \leq 2 \exp\left(-\frac{(d-1)t^2}{2}\right).$$

Since the POT algorithm searches over the entire unlabeled pool of size n_U, we must bound the worst case maximum inner product to prevent the algorithm from spuriously matching orthogonal noise. Applying the union bound over all n_U samples:

$$P(\max_{x^\perp \in \mathcal{X}^\perp} |\langle z_i^\perp, x^\perp \rangle| > t) \leq \sum_{x^\perp \in \mathcal{X}^\perp} P(|\langle z_i^\perp, x^\perp \rangle| > t) \leq 2n_U \exp\left(-\frac{(d-1)t^2}{2}\right)$$

This establishes that in high-dimensional space, the maximum orthogonal correlation is strictly bounded, mitigating the risk of POT exploiting spurious correlations.

A.2 Proof of Theorem 4.5

We consider a nuisance perturbation that adds a bias vector z_{bias} with magnitude λ to the consensus direction $\overline{z}$.

Perturbation. The perturbed embedding $\tilde{z}$ is obtained by adding a nuisance direction z_{bias} with magnitude λ to $\overline{z}$. Let $c := \langle \overline{z}, z_{\text{bias}} \rangle$, so that $|c| \leq \epsilon_{\text{bias}}$ by Assumption 4.4. The perturbed target is

$$\tilde{z} = \frac{\overline{z} + \lambda z_{bias}}{\sqrt{1 + 2\lambda c + \lambda^2}}.$$

Effect on the transport cost. We examine how the similarity between an anchor $z_i = \tau_i \overline{z} + \epsilon_i z_i^\perp$ and $\tilde{z}$ changes with the perturbation parameter λ. The cosine similarity between z_i and $\tilde{z}$ is

$$\begin{aligned} d(z_i, \tilde{z}) &= \frac{\langle \tau_i \overline{z} + \epsilon_i z_i^\perp, \overline{z} + \lambda z_{bias} \rangle}{\sqrt{1 + 2\lambda c + \lambda^2}} = \frac{\tau_i + \lambda \left(\tau_i c + \epsilon_i \langle z_i^\perp, z_{bias} \rangle\right)}{\sqrt{1 + 2\lambda c + \lambda^2}} \\ &= \tau_i + \lambda \epsilon_i \langle z_i^\perp, z_{bias} \rangle + \mathcal{O}(\lambda^2), \end{aligned}$$

where the last step uses $(1 + 2\lambda c + \lambda^2)^{-1/2} = 1 - \lambda c + \mathcal{O}(\lambda^2)$, so that the term $\lambda \tau_i c$ cancels; the constant in $\mathcal{O}(\lambda^2)$ is uniform over $|c| \leq \epsilon_{\text{bias}} < 1$. In the exactly orthogonal case $c = 0$, the expansion is $\tau_i + \lambda \epsilon_i \langle z_i^\perp, z_{bias} \rangle - \frac{1}{2}\tau_i \lambda^2 + \mathcal{O}(\lambda^3)$. Hence the first-order term does not depend on c; the approximate orthogonality in Assumption 4.4 affects only the second-order remainder and ensures that the perturbation acts mainly as a nuisance rather than as a change in quality. The change in the alignment score caused by the perturbation is $\lambda \epsilon_i \langle z_i^\perp, z_{bias} \rangle + \mathcal{O}(\lambda^2)$. The derivation reveals a first-order sensitivity term. However, the PUAudit is designed with a specific geometric intervention to mitigate this, the P-group denoising stage.

By selectively retaining the anchors that exhibit high cosine alignment with the empirical center, the denoising procedure enforces $\tau_i \to 1$. Consequently, the magnitude of $\epsilon_i = \sqrt{1 - \tau_i^2}$ is bounded by a small tolerance threshold $\epsilon_p \ll 1$.

Since $|\langle z_i^\perp, z_{bias} \rangle| \leq 1$, the first-order sensitivity is bounded by $\mathcal{O}(\epsilon_p \cdot \lambda)$. Thus, while the robustness is not purely second-order $\mathcal{O}(\lambda^2)$ as it would be with the theoretical $\overline{z}$, the first-order disturbance is heavily attenuated by ϵ_p, formalizing the robustness of the transport cost to presentation-level nuisance perturbations.

■

A.3 Proof of insensitivity to the number of verified positives

We provide a theoretical explanation for the statement in Section 5.2. The central intuition is that PUAudit utilizes the denoised positive set $\mathcal{D}'_P$ primarily to estimate a stable reference direction (the "quality vector") in the difference-embedding space. Once $\mathcal{D}'_P$ reaches a certain cardinality, it provides sufficient information to characterize this direction, rendering additional positive samples redundant.

Assumption A.2 (Directional concentration of $\mathcal{D}'_P$)**.** There exists a latent unit vector $\overline{z}$ (representing the gold-standard preference direction) such that for all $(x_i, o_i^w, o_i^\ell, y_i) \in \mathcal{D}'_P$, the corresponding normalized difference embedding z_i satisfies $\langle z_i, \overline{z} \rangle \geq 1 - \kappa$ for some $\kappa \in (0, 1)$. Furthermore, we assume that the latent distribution of these positives is symmetric around $\overline{z}$ within this spherical cap.

Theorem A.3 (Informational Sufficiency of the Positive Set)**.** *Under Assumption A.2, for any $\varepsilon > 0$, there exists a finite threshold $n^\star(\varepsilon, \kappa)$ such that whenever $|\mathcal{D}'_P| \geq n^\star$, the empirical reference direction estimated by* PUAudit *is ε-aligned with $\overline{z}$ with high probability.*

The P-group denoising procedure in Section 3.2 enforces Assumption A.2 by retaining only co-directional anchors, ensuring that all vectors in $\mathcal{D}'_P$ lie within a narrow spherical cap around $\overline{z}$. According to the standard concentration of vector averages on the unit sphere, the empirical mean direction converges to the population mean $\overline{z}$ at a rate of $\mathcal{O}(|\mathcal{D}'_P|^{-1/2})$.

In our framework, p_{ratio} determines the effective size of $\mathcal{D}'_P$. Once $|\mathcal{D}'_P| \geq n^\star$, the estimated reference direction becomes stable, and further increasing p_{ratio} yields only marginal gains in variance reduction without providing new structural information. Consequently, the reference anchor for POT remains fixed, leading to the observed plateau in OT-based scores and adjusted consistency Cons_{adj}. This demonstrates that PUAUDIT is robust to the quantity of positive data, provided the minimum threshold for representational sufficiency is met.

A.4 Feature extraction and P-group denoising

Feature extraction. We use a frozen encoder F_θ to construct the representation space. In our main experiments, F_θ is instantiated with a reward-model encoder, but the method itself does not require a specific encoder architecture. For a prompt x and response o, we format the input as a single-turn conversation with user turn x and assistant turn o using the encoder's chat template, truncate it to 1024 tokens, and use the last-token hidden representation as the prompt–response embedding $F_\theta(x, o) \in \mathbb{R}^d$.

For a pairwise comparison, let o^w and o^ℓ be the response preferred and dispreferred by the initial judge. We define the comparison representation as

$$z_i = \frac{\Delta_i}{\|\Delta_i\|}, \qquad \Delta_i = F_\theta(x_i, o_i^w) - F_\theta(x_i, o_i^\ell). \tag{3}$$

Thus, each comparison is represented by a unit vector on $\mathbb{S}^{d-1}$ indicating the direction from the dispreferred response to the preferred response. Compared with absolute response embeddings, this differenced representation directly captures relative preference information and reduces sensitivity to scale variations across examples.

P-core filtering. Although human annotations provide valuable supervision, they may still contain noise. We therefore first remove positive examples whose preferred-response embeddings are far from the positive-group center. For each positive example, let

$$e_i = F_\theta(x_i, o_i^w).$$

We compute the empirical positive center

$$\bar{e} = \frac{1}{|\mathcal{D}_P|} \sum_{(x_i, o_i^w, o_i^\ell, y_i) \in \mathcal{D}_P} e_i,$$

and score each example by cosine similarity

$$s_i^{(e)} = \frac{\langle e_i, \bar{e} \rangle}{\|e_i\| \, \|\bar{e}\|}.$$

We retain the top-α_1 fraction of examples according to $s_i^{(e)}$, yielding

$$\mathcal{D}_P^{(1)} = \text{Top}_{\lfloor \alpha_1 |\mathcal{D}_P| \rfloor} \left(\mathcal{D}_P; s_i^{(e)} \right),$$

where $\alpha_1 \in (0, 1]$ is a retention ratio.

Improvement-alignment filtering. Next, we filter examples according to whether their improvement directions are aligned with the dominant positive direction. We compute

$$\bar{z} = \frac{1}{|\mathcal{D}_P^{(1)}|} \sum_{(x_i, o_i^w, o_i^\ell, y_i) \in \mathcal{D}_P^{(1)}} z_i,$$

and assign each retained example an alignment score

$$s_i^{(z)} = \frac{\langle z_i, \bar{z} \rangle}{\|z_i\| \, \|\bar{z}\|}.$$

The final cleaned positive set is obtained by retaining the top-α_2 fraction:

$$\mathcal{D}'_P = \mathrm{Top}_{\lfloor \alpha_2 |\mathcal{D}_P^{(1)}| \rfloor} \left(\mathcal{D}_P^{(1)}; s_i^{(z)} \right),$$

where $\alpha_2 \in (0, 1]$ controls the strength of improvement-direction filtering.

In our experiments, we set $\alpha_1 = \alpha_2 = 0.7$. This choice removes geometrically inconsistent positive examples while preserving enough positive mass for the subsequent POT step.

A.5 Evaluation metrics

We evaluate LLM-as-judge decisions using a single metric: consistency. For a fixed judge model and question type, let $\zeta_i^{(\mathrm{H})}$ denote the human preference and $\zeta_i^{(\mathrm{LLM})}$ denote the LLM-judge preference. We define

$$\mathrm{Cons} \;=\; \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathbb{I}\Big\{ \zeta_i^{(\mathrm{LLM})} = \zeta_i^{(\mathrm{H})} \Big\},$$

where $\mathcal{I}$ denotes the set of test samples for which both human annotations and LLM judge prediction are available.

We report consistency before and after applying PUAudit. Concretely, $\mathrm{Cons}_{\mathrm{orig}}$ is computed using the raw judge outputs $(\zeta_i^{(\mathrm{LLM})}, \zeta_i^{(\mathrm{H})})$, while $\mathrm{Cons}_{\mathrm{adj}}$ is computed using the adjusted preferences $(\hat{\zeta}_i^{(\mathrm{LLM})}, \zeta_i^{(\mathrm{H})})$. Higher consistency indicates better alignment between the LLM judge and human judgments, reflecting more human-like evaluation behaviour.

B ADDITIONAL FIGURES, TABLES, AND EXPERIMENTAL DETAILS

B.1 NOTATION AND SETUP

Table 3: Main notation used in the paper. Additional symbols are defined locally when introduced.

Symbol	Meaning
	Data and Preference
$\mathcal{D}$	collection of pairwise comparisons
x_i	prompt (question)
o_i^A, o_i^B	two candidate responses
$\zeta_i^{(\mathrm{LLM})}$	preference given by the LLM judge
$\zeta_i^{(\mathrm{H})}$	human preference (when available)
o_i^w, o_i^ℓ	winner and loser induced by the judge
y_i	consistency indicator, $\mathbb{I}\{\zeta_i^{(\mathrm{LLM})} = \zeta_i^{(\mathrm{H})}\}$
$\mathcal{D}_P$	verified positive set (human-consistent)
$\mathcal{D}_U$	unlabeled set (mixture of latent positives and negatives)
	Representation
F_θ	frozen encoder (reward model or embedding model)
z_i	normalized difference embedding
μ_P, μ_U	empirical distributions of $\mathcal{D}_P$ and $\mathcal{D}_U$
μ_+, μ_-	latent positive and negative distributions
α	latent positive proportion in $\mathcal{D}_U$
	POT and Alignment
C_{ij}	cosine transport cost between embeddings
$\Gamma^\star$	optimal partial transport plan
m	transported mass (transport fraction)
s_j	transport-based alignment score
$\bar{s}_j$	normalized alignment score
η	decision threshold for preference correction
	Analysis Variables
p_{ratio}	fraction of verified positives used in auditing
λ	magnitude of bias perturbation
z_{bias}	nuisance (bias) direction
ϵ_{bias}	bound on $\lvert\langle \bar{z}, z_{\mathrm{bias}}\rangle\rvert$

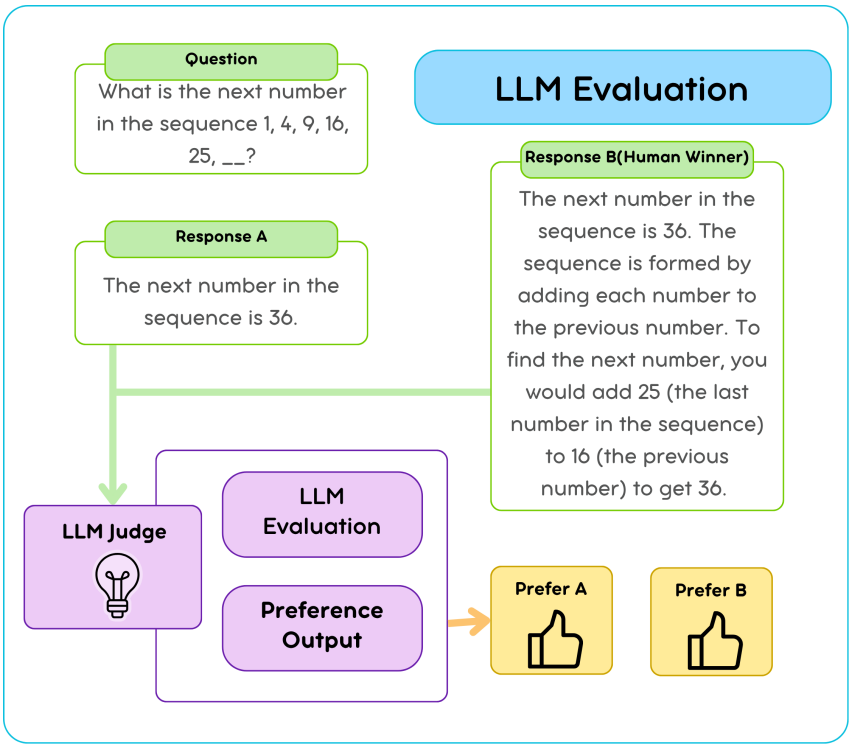

Figure 4: A pairwise comparison from our data. Two models answer the same question, and the human prefers one of the responses (here Response B). The LLM judge reads the question and both responses, writes an evaluation, and outputs a preference; the comparison is a verified positive if a human checks it and agrees.

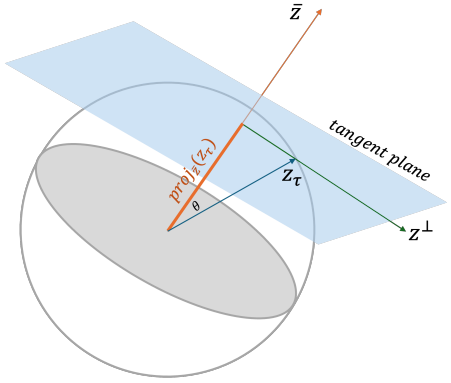

Figure 5: The vector decomposition for Lemma A.1: We denote θ as the angle between any normalized embedding z_τ and the consensus direction $\bar{z}$, and $proj_{\bar{z}}(z_\tau)$ as the projected amount of z_τ onto $\bar{z}$.

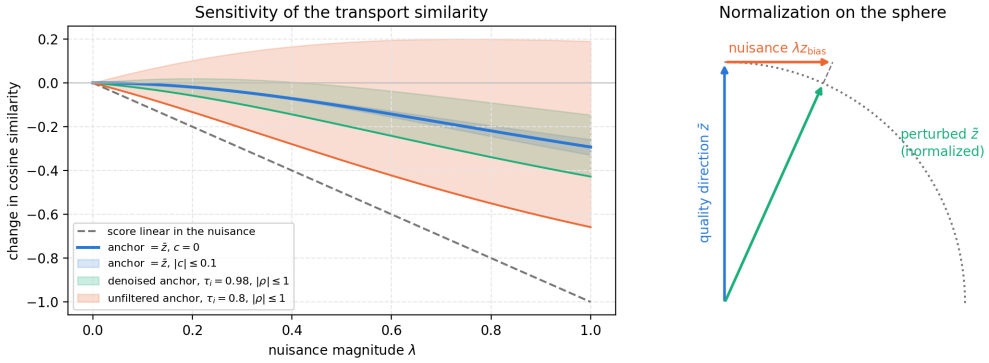

Figure 6: Illustration of Theorem 4.5. **Left**: change in the cosine similarity between an anchor $z_i = \tau_i \overline{z} + \epsilon_i z_i^{\perp}$ and the perturbed direction $\tilde{z}$ as the nuisance magnitude λ grows, for the ideal anchor $z_i = \overline{z}$ (with $c = \langle \overline{z}, z_{\text{bias}} \rangle$ in $[-0.1, 0.1]$) and for anchors with $\tau_i = 0.98$ and $\tau_i = 0.8$ (bands over the alignment $\rho \in [-1, 1]$ of $z_i^{\perp}$ with the nuisance). Denoising keeps τ_i close to one, so the first-order term $\lambda \epsilon_i \rho$ stays small. **Right**: normalization maps the perturbed vector back to the sphere.

Table 4: Effect of frozen encoder choice on the MT-Bench pipeline. BGE-M3 denotes BGE-M3 (Chen et al., 2024); Gemma-300M denotes EMBEDDINGGEMMA-300M (Vera et al., 2025); Qwen3-0.6B denotes QWEN3-EMBEDDING-0.6B (Zhang et al., 2025); Skywork-RM denotes SKYWORK-REWARD-LLAMA-3.1-8B-V0.2 (AI, 2024). Results are averaged over three local judge models and GPT-5.4-mini. Time and memory are reported in seconds and GB, respectively. Gemma-300M illustrates the failure mode that results from skipping a separability check: its flip rate reaches 1.000 and its adjusted ratio falls to 0.306, below the original judge's accuracy of 0.694.

Encoder	**Orig.↑**	**Adj.↑**	**Gain↑**	**Flip**	**Time↓**	**Mem.↓**
BGE-M3	0.694	0.708	+0.014	0.094	5.87	0.846
Gemma-300M	0.694	0.306	-0.387	1.000	5.40	0.816
Qwen3-0.6B	0.694	0.613	-0.081	0.306	5.63	0.832
Skywork-RM	0.694	0.739	+0.045	0.098	6.12	1.150

B.2 DATA GENERATION

Offline LLM-as-Judge Prompt

Instruction: Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses. Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: [[A]] if assistant A is better, [[B]] if assistant B is better, and [[C]] for a tie.

Task: Question Type Classification (Zero-Shot)
Given the **USER QUESTION**, classify it into exactly one of the following categories.

Categories:

- A: Programming (coding, debugging, devops, software engineering)
- B: Math or logical reasoning
- C: Factual or knowledge-based question
- D: Writing, translation, summarization, or creative generation
- E: Advice, recommendation, or planning
- F: Other, meta-level, or unclear questions

Rules:

- Choose exactly one category.
- Do not provide any explanation for the classification.
- The output must strictly follow the required format.

Input Format:
[User Question] {question}
[Assistant A] {response_A}
[Assistant B] {response_B}

Output Format (Strict):
The *last two non-empty lines* of the output must be:
[[A]], [[B]], or [[C]] (preference verdict)
[[TYPE:X]], where X ∈ {A,B,C,D,E,F} (question type)

Figure 7: Prompt template used by the offline LLM-as-judge to generate preference judgments and question-type labels. The strict output constraints enable robust automatic parsing.

B.3 SIMULATION DETAILS

The simulation mirrors the auditing problem of Section 3.1. Each run contains 1,200 comparisons. For comparison i, a latent human preference $h_i \in \{\mathrm{A}, \mathrm{B}\}$ is drawn with 600 comparisons in each class, together with a latent coordinate $u_i \mid h_i \sim \mathcal{N}_2(\pm\frac{\Delta}{2}e_1,\, s\Sigma)$, where the sign is positive for

A, $\Sigma = \left(\begin{smallmatrix} 1.5 & 0.3 \\ 0.3 & 1.5 \end{smallmatrix}\right)$, e_1 is the first standard basis vector, the class-centre distance is $\Delta = 4$, and the covariance scale is $s = 1$ unless stated otherwise. The judge preference $\zeta_i^{(\mathrm{LLM})}$ equals h_i except that it is flipped with probability q_i, where $\mathbb{E}[q_i] = 0.3$. Under class-dependent (CD) noise, $q_i = 0.3$ for every comparison. Under distribution-dependent (DD) noise, q_i is proportional to $\sigma(-1.2\,|u_{i1}|)$, with σ the logistic function, so errors concentrate near the class boundary. The consistency label is $y_i = \mathbb{I}\{\zeta_i^{(\mathrm{LLM})} = h_i\}$.

The encoder is simulated by a fixed random map $W \in \mathbb{R}^{2\times 64}$ with unit-norm columns: z_i is the normalized vector $u_i^\top W + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0, 0.03^2 I_{64})$, multiplied by $+1$ if the judge prefers A and by -1 otherwise, which reproduces the judge-oriented difference embedding of Eq. (1). Human preferences are revealed for about 15% of the comparisons. Under SCAR, each comparison is verified with probability 0.15. Under SAR, the probability is proportional to $(1 + \|u_i\|)^{-1}$, rescaled to the same mean, so that comparisons near the boundary are verified more often. PUAUDIT uses the verified agreements as anchors, sets m to the verified agreement rate, and keeps all hyperparameters of the real-data experiments. Each setting is repeated over five seeds, and accuracy is computed on the unverified comparisons.

B.4 SIMULATION SWEEPS

For each of the four noise and verification settings, we vary one factor of the design at a time while keeping the others at their defaults ($\Delta = 4$, $s = 1$, 15% verified, 30% corruption), with five seeds per value. Figures 8–11 report the accuracy of the original judge and of PUAUDIT, the gain, and the recall of human-consistent judgments, that is, the fraction of unverified correct judgments that PUAUDIT keeps.

Class-centre distance (Figure 8). From $\Delta = 2$ on, the gain exceeds 10 points in every setting, and under CD noise it levels off at about 27.7 points for $\Delta \geq 6$. Under DD noise, the original accuracy rises with Δ because fewer comparisons lie near the boundary where the errors concentrate, and the gain settles between 14 and 25 points.

Covariance scale (Figure 9). The gain decreases as the classes spread out, from 27.6 to 12.0 points under CD+SCAR, from 27.6 to 9.4 under DD+SCAR, and from 24.1 to 6.8 under DD+SAR as s grows from 0.25 to 4, and it stays positive throughout.

Verification rate (Figure 10). With 2% of the comparisons verified, about 24 of them, PUAUDIT already gains 21.4 to 22.4 points under CD noise, 21.5 under DD+SCAR, and 16.2 under DD+SAR. The gain stays at 12 points or more up to 30% verification in every setting.

Corruption rate (Figure 11). As the judge's corruption rate rises from 10% to 45%, the adjusted accuracy of PUAUDIT stays between 93.7% and 96.9% under CD noise and DD+SCAR and between 86.4% and 94.7% under DD+SAR, so the gain grows from 4.5–7.7 points to 25.1–40.2 points.

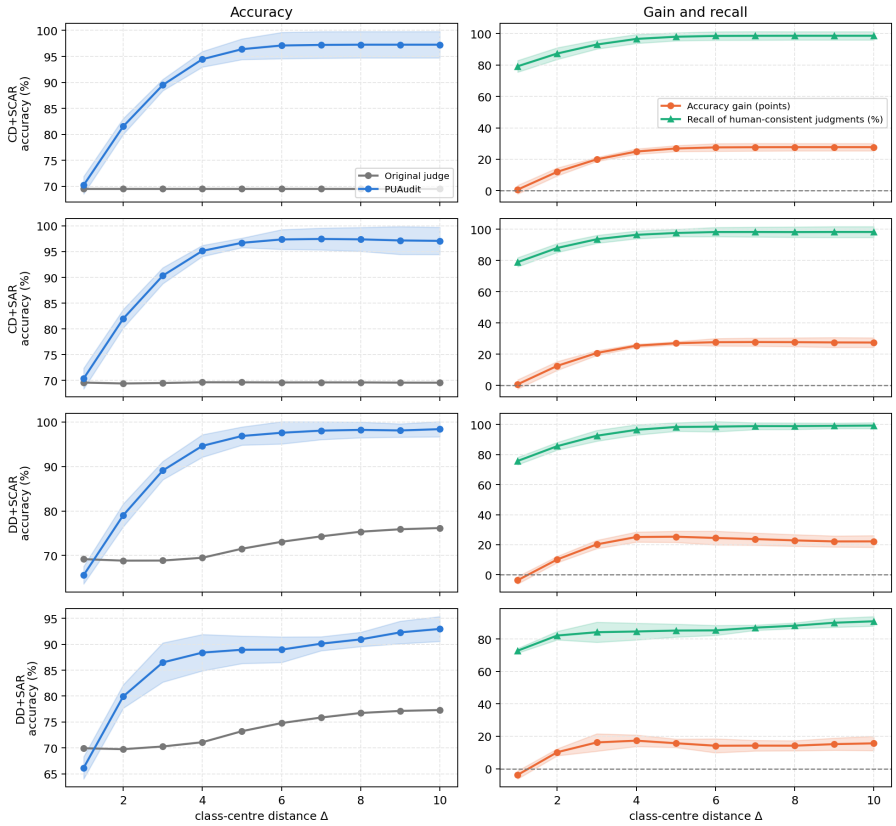

Figure 8: Sensitivity of PUAUDIT to the class-centre distance Δ in all four settings (covariance scale 1, 15% verified, 30% corruption). Shaded bands are 95% confidence intervals over five seeds.

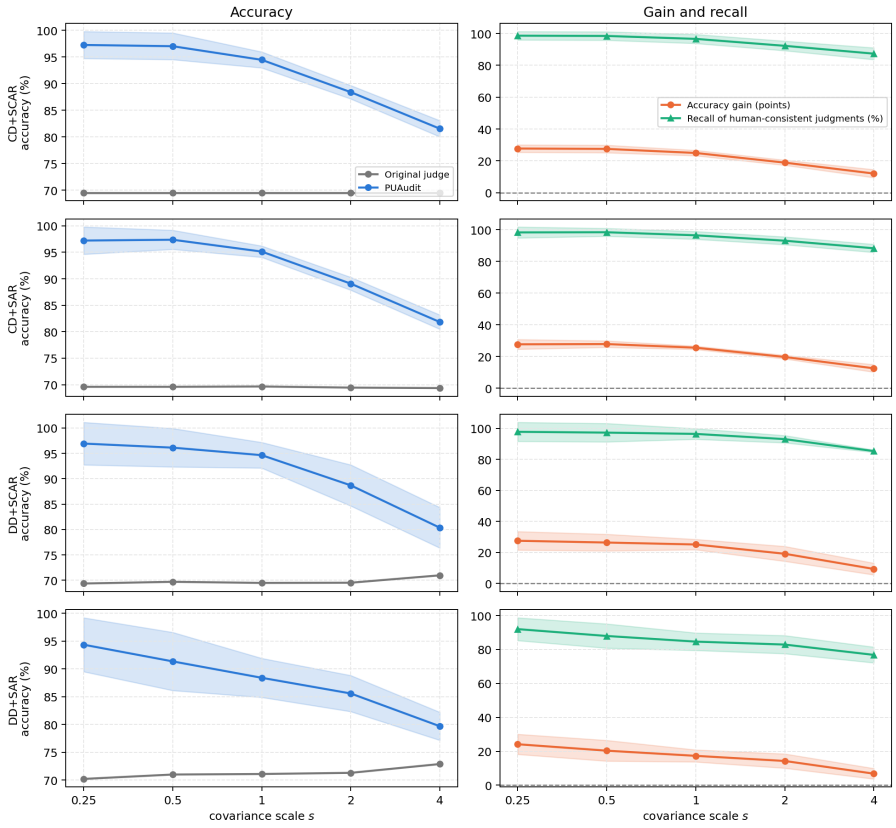

Figure 9: Sensitivity of PUAUDIT to the covariance scale s in all four settings ($\Delta = 4$, 15% verified, 30% corruption). Shaded bands are 95% confidence intervals over five seeds.

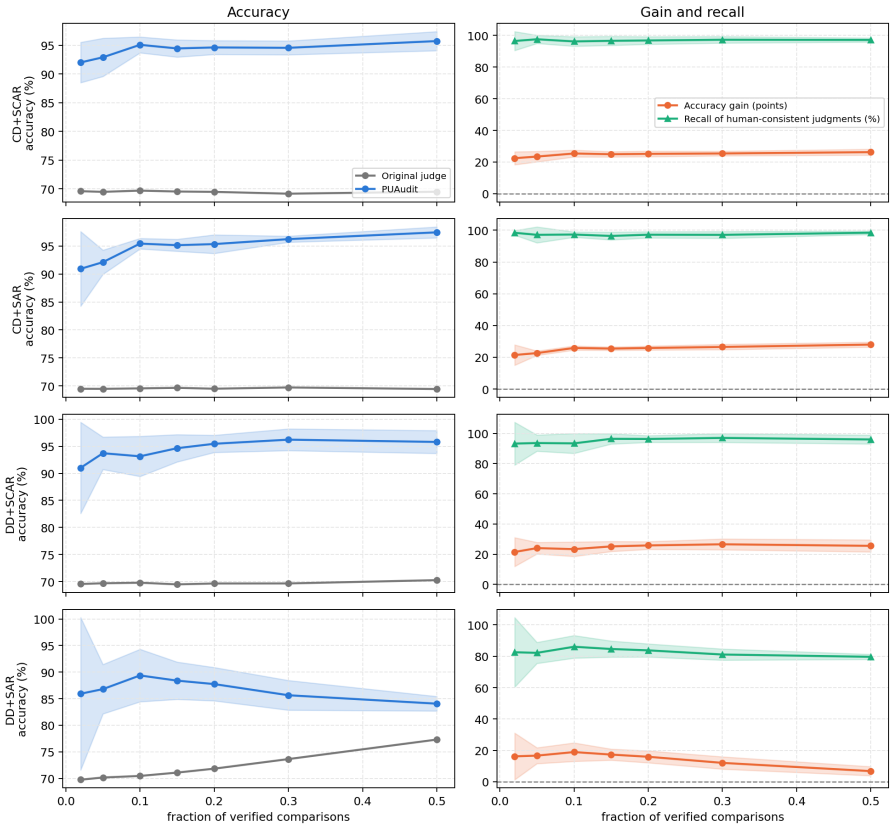

Figure 10: Sensitivity of PUAUDIT to the fraction of verified comparisons in all four settings ($\Delta = 4$, $s = 1$, 30% corruption). Shaded bands are 95% confidence intervals over five seeds.

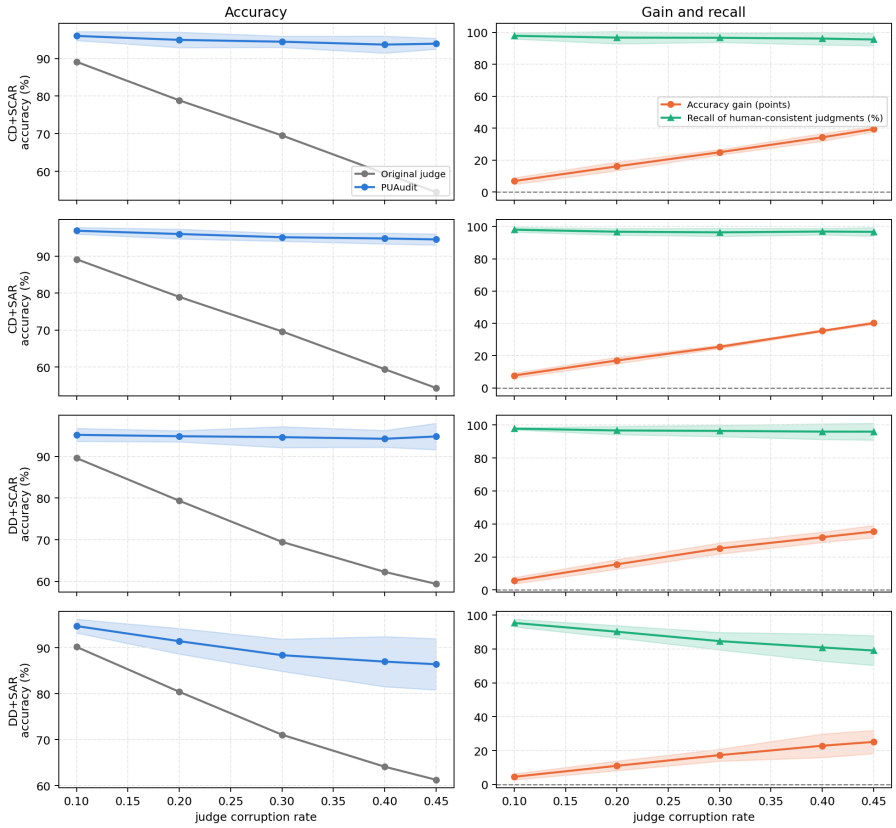

Figure 11: Sensitivity of PUAUDIT to the judge's corruption rate in all four settings ($\Delta = 4$, $s = 1$, 15% verified). Shaded bands are 95% confidence intervals over five seeds.

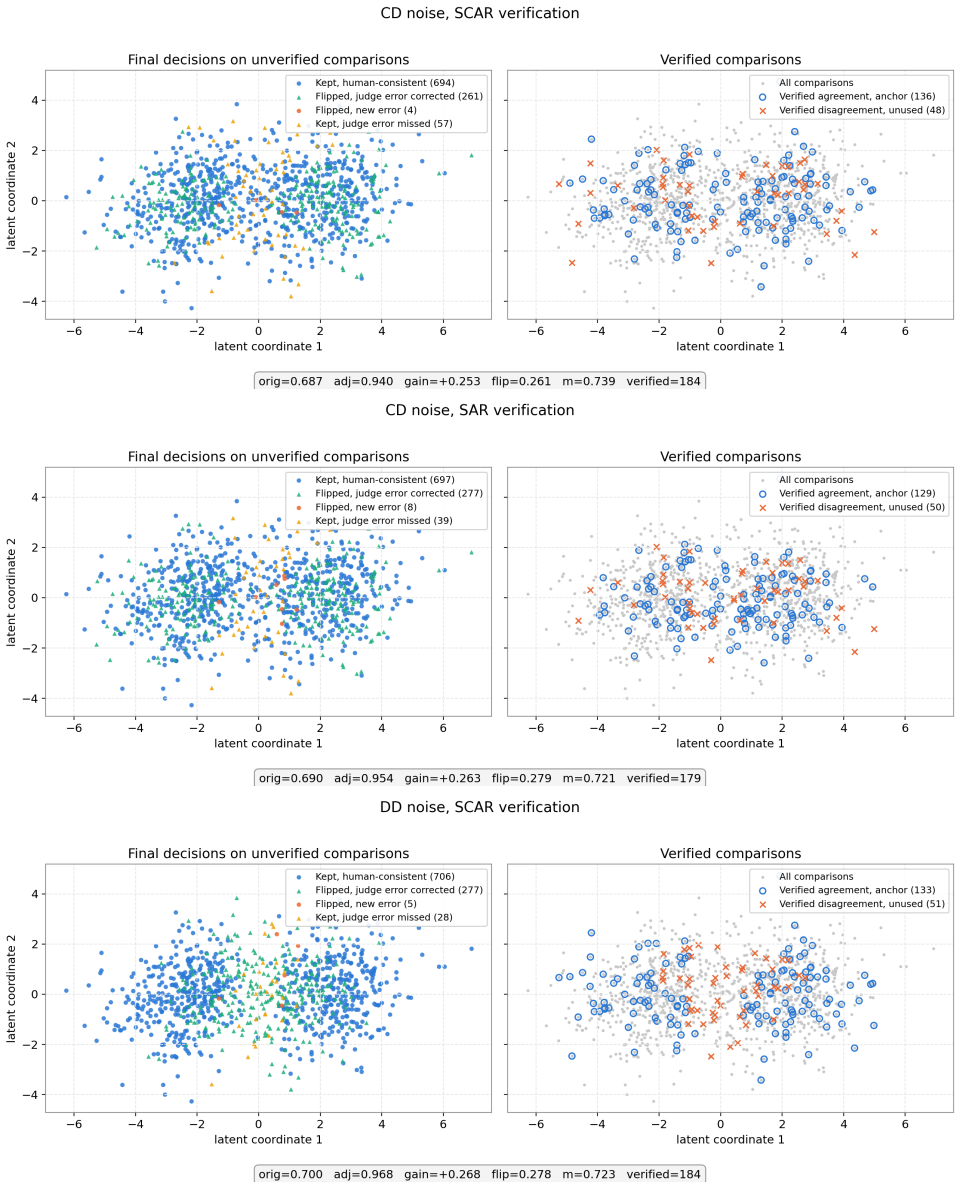

Figure 12: Decisions of PUAUDIT and the verified comparisons in the CD+SCAR, CD+SAR, and DD+SCAR settings (seed 1); DD+SAR is shown in Figure 2.

B.5 REAL-DATA DETAILS

From Chatbot Arena and MT-Bench we obtain a human-annotated dataset $\tilde{\mathcal{H}} = \{(x_i, o_i^A, o_i^B, \zeta_i^{(H)})\}$. Each LLM judge then produces a preference $\zeta_i^{(\mathrm{LLM})}$ with the prompt in Figure 7, which yields $\mathcal{H} = \{(x_i, o_i^A, o_i^B, \zeta_i^{(H)}, \zeta_i^{(\mathrm{LLM})})\}$; Algorithm 2 in Appendix B.8 summarizes the procedure. The Chatbot Arena judge checkpoints (MISTRALAI/MISTRAL-7B-INSTRUCT-V0.3 and ELEUTHERAI/LLEMMA_7B_MUINSTRUCT_CAMELMATH) use greedy decoding with temperature= 0, top_p= 1, and a maximum of 512 generated tokens. For MT-Bench, we use GPT-5.4-MINI through the OpenAI Chat Completions API with temperature= 0 and max_tokens= 512, with top_p left at the API default. Question-type labels (QTA–QTF) are assigned by the judge with the same prompt. Embeddings are the last-token hidden states of SKYWORK-REWARD-LLAMA-3.1-8B-V0.2 (AI, 2024) in bfloat16, with inputs truncated to 1,024 tokens.

The baselines receive the same splits and embeddings, reduced to at most 64 dimensions by a singular value decomposition fitted on the verified subset. The positive–unlabelled baselines (Elkan–Noto, nnPU, SAR-PU, POT-PU, and a one-class SVM) observe only the verified agreements; the supervised baselines (logistic regression, random forest, MLP, k-NN, a cosine-centroid classifier, and label spreading) observe the verified agreements and disagreements. For every seed, an exact McNemar test compares the decisions of PUAUDIT with those of the original judge on the unverified comparisons, and the p-values are Holm-corrected across all configurations and seeds. Against each baseline, a paired t-test over the ten seeds compares accuracies, Holm-corrected across baselines. Runtime is the CPU time of one audit on precomputed embeddings. In every split, original accuracy is the fraction of human-consistent judge decisions in the test pool, and adjusted accuracy is computed on those same comparisons. Reported dataset sizes count all comparisons; accuracy denominators count only the unverified comparisons. A verification budget of 80% therefore leaves approximately 20% for testing; a 20% verification budget leaves approximately 80%. Cross-method tables use the common valid seeds (eight for Mistral QTB at 5%, ten otherwise).

B.6 ADDITIONAL REAL-DATA RESULTS

Tables 5, 6, and 7 report the Arena/MT-Bench comparison, matched-label reference, and budget sweep.

Table 5: Low-budget results for all Chatbot Arena and MT-Bench configurations. Orig. and PUAUDIT are scored on identical unverified comparisons in each split. Accuracy (%) is the mean; parentheses give the max–min range for adjusted methods. Best PU has the highest mean among Elkan–Noto, nnPU, SAR-PU, and POT-PU. All entries use ten common seeds, except Mistral QTB at 5% (eight).

	5% verified			10% verified		
Configuration	Orig.	PUAUDIT	Best PU	Orig.	PUAUDIT	Best PU
Mistral, QTA	58.4	76.1 (1.0)	67.6 (8.7)	58.3	76.1 (1.4)	75.2 (2.0)
Mistral, QTB	50.6	68.8 (26.2)	75.0 (17.9)	50.3	70.0 (38.0)	69.2 (35.4)
Mistral, QTC	53.1	77.0 (1.8)	70.5 (7.8)	53.2	76.9 (1.2)	75.6 (2.4)
Mistral, QTD	58.6	73.2 (1.6)	61.5 (9.0)	58.7	73.3 (2.3)	70.2 (7.9)
Mistral, QTE	49.7	71.1 (14.1)	56.9 (8.7)	49.6	71.9 (3.4)	63.6 (17.3)
Mistral, QTF	37.5	72.0 (2.1)	67.1 (5.8)	37.5	72.2 (1.6)	71.5 (3.4)
Llemma, QTA	52.7	73.3 (1.8)	57.5 (7.6)	52.7	73.5 (1.8)	65.9 (15.3)
Llemma, QTB	64.1	64.5 (19.5)	64.8 (8.1)	64.4	71.8 (28.5)	64.4 (2.1)
Llemma, QTC	57.1	75.5 (3.2)	58.7 (6.3)	57.1	75.7 (3.7)	69.3 (5.7)
GPT-5.4-mini, MT-Bench	77.9	75.2 (6.5)	64.6 (8.8)	78.0	75.3 (5.2)	66.3 (6.0)

Table 6: Matched-fitting-label reference at the 5% supervised budget. L counts fitting labels per method; PUAUDIT obtains L agreements from a larger reviewed pool whose agreement rate sets m. Orig. and all methods are scored on its common unverified complement. Accuracy (%): mean, with max–min ranges for adjusted methods; ten splits except Mistral QTB (nine). Best PU and Best supervised select the highest mean within their respective groups; supervised methods fit to verified agreements and errors.

Configuration	L	Orig.	PUAUDIT	Best PU	Best supervised
Mistral, QTA	383	58.3	76.2 (1.2)	73.7 (4.5)	77.7 (1.7)
Mistral, QTB	4	50.5	67.9 (30.3)	74.8 (29.0)	76.2 (9.9)
Mistral, QTC	406	53.2	77.0 (1.0)	74.4 (3.9)	78.4 (1.8)
Mistral, QTD	65	58.7	73.3 (1.7)	72.1 (7.9)	74.6 (6.2)
Mistral, QTE	52	49.6	71.9 (3.8)	65.6 (18.1)	70.0 (8.7)
Mistral, QTF	250	37.5	72.3 (1.8)	72.2 (3.4)	74.7 (3.5)
Llemma, QTA	59	52.6	73.5 (1.2)	68.4 (6.7)	72.9 (4.5)
Llemma, QTB	16	64.2	73.4 (7.5)	67.5 (30.3)	75.6 (12.9)
Llemma, QTC	65	57.1	76.0 (3.6)	72.5 (8.7)	74.8 (5.9)
GPT-5.4-mini, MT-Bench	141	77.9	75.4 (5.0)	65.3 (4.6)	77.1 (10.6)

Table 7: Verification-budget sweep (mean over ten seeds). Original and adjusted accuracies (%) use the same test comparisons at each budget; their test pools change with the budget. Verified comparisons count all revealed labels. CPU time covers one audit on precomputed embeddings.

Judge	Metric	1%	3%	5%	10%	20%	50%	80%
Mistral, QTC	Orig. accuracy	53.1	53.1	53.1	53.2	53.1	52.8	52.6
	Adjusted accuracy	75.8	77.0	77.0	76.9	77.0	76.8	76.8
	Verified comparisons	81	244	406	812	1623	4058	6494
	CPU time (s)	0.71	1.51	2.21	3.77	6.12	9.70	5.90
Llemma, QTA	Orig. accuracy	52.7	52.6	52.7	52.7	52.6	52.3	52.0
	Adjusted accuracy	63.9	72.6	73.3	73.5	73.5	73.6	73.0
	Verified comparisons	12	36	59	118	237	592	947
	CPU time (s)	0.05	0.05	0.06	0.09	0.21	0.19	0.13
GPT-5.4-mini	Orig. accuracy	77.9	77.9	77.9	78.0	77.7	77.6	76.3
	Adjusted accuracy	73.4	75.1	75.2	75.3	75.7	75.7	75.8
	Verified comparisons	35	89	141	275	529	1304	2074
	CPU time (s)	0.15	0.26	0.36	0.62	1.01	1.54	0.84

B.7 ABLATION AND SENSITIVITY ANALYSIS

Table 8: Component ablation at the 20% verification budget (accuracy, mean (max–min range) over ten splits). QTC, QTF, and QTA are Chatbot Arena question types; MT-Bench uses GPT-5.4-mini. Bold values are the full method; green cells are the best variant in each column.

Variant	**QTC**	**QTF**	**Llemma QTA**	**MT-Bench**
Original judge	53.1%	37.4%	52.6%	77.7%
PUAUDIT (full)	**77.0% (1.1)**	**72.2% (1.3)**	**73.5% (2.0)**	**75.7% (2.8)**
No transport	53.1% (1.1)	37.4% (0.8)	52.6% (1.6)	77.7% (1.7)
Full OT ($m = 1$)	53.2% (0.9)	37.5% (0.8)	52.7% (1.5)	77.8% (1.8)
No positive denoising	67.0% (2.9)	62.0% (3.6)	64.5% (3.3)	74.3% (4.4)
No P-core filtering	76.8% (1.0)	72.3% (1.1)	73.4% (2.3)	75.6% (2.6)
No improvement-alignment filtering	67.8% (2.5)	63.9% (4.5)	65.4% (5.1)	73.0% (2.7)
Euclidean cost	77.0% (1.1)	72.2% (1.2)	73.5% (2.0)	75.9% (3.0)

Table 8 isolates the components of PUAUDIT. Partial transport is essential: without transport, or with full optimal transport ($m = 1$), every comparison receives mass and accuracy returns to that of

the original judge. Positive denoising is the second key component. Removing it costs 10.0 points on QTC, 10.2 on QTF, and 9.0 on Llemma QTA, and the improvement-alignment filter accounts for most of this effect, whereas the P-core filter and the choice between cosine and Euclidean cost change accuracy by at most 0.2 points.

Table 9: Accuracy (%) as a function of the transported mass m at the 20% budget (mean over ten splits). The last column (bold) uses the default m, the verified agreement rate; green cells are the best value in each row.

Judge	**Orig.**	0.1	0.3	0.5	0.7	0.8	0.85	0.9	0.95	0.99	**Default**
Mistral-7B, QTC	53.1	55.6	69.7	76.6	74.2	69.0	65.5	61.6	57.6	54.1	**77.0**
Mistral-7B, QTF	37.4	67.0	72.8	69.1	60.2	53.4	49.7	45.8	41.8	38.3	**72.2**
Llemma-7B, QTC	57.0	51.8	65.3	74.9	74.8	71.3	68.0	64.7	61.3	57.8	**75.9**
GPT-5.4-mini, MT-Bench	77.7	31.2	48.3	62.3	72.8	76.6	79.0	80.5	79.8	78.2	**75.7**

Table 9 varies the transported mass. On Chatbot Arena, accuracy peaks near the default agreement-rate setting and decreases smoothly on both sides. On MT-Bench, the held-out positive proportion equals the original judge accuracy (77.7%), so the default $m = 0.78$ already matches that proportion. The best tested value, $m = 0.9$, gives 80.5%: this is an empirical transport budget with slack for errors in the thresholded alignment decisions, not evidence that the positive proportion is 90%. Because $0.9 > 0.777$, this sensitivity point is outside the theorem guarantee for $m \leq \alpha$; the guarantee does not cover that operating point.

Table 10: Sensitivity to the denoising thresholds $\alpha_1 = \alpha_2 = \alpha$ at the 20% budget (accuracy, mean (max–min range) over ten splits); $\alpha = 1$ keeps every verified agreement. Bold values use the default $\alpha = 0.7$; green cells are the best value in each row.

Judge	**Orig.**	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$
Mistral-7B, QTC	53.1%	77.2% (1.4)	**77.0% (1.1)**	73.7% (1.9)	67.0% (2.9)
Mistral-7B, QTF	37.4%	72.7% (0.7)	**72.2% (1.3)**	68.0% (3.5)	62.0% (3.6)
Llemma-7B, QTC	57.0%	75.9% (1.9)	**75.9% (2.4)**	69.7% (4.3)	63.3% (5.8)
GPT-5.4-mini, MT-Bench	77.7%	76.0% (3.0)	**75.7% (2.8)**	75.1% (2.4)	74.3% (4.4)

Table 10 varies the denoising thresholds. Accuracy is stable for $\alpha \in \{0.5, 0.7\}$ and drops by 10 to 13 points on Chatbot Arena when no verified agreement is filtered out ($\alpha = 1$).

Table 11: Accuracy of PUAUDIT when the verified comparisons are selected at random (SCAR) or depending on their features (SAR), mean (max–min range) over ten splits. Under SAR, a comparison is verified with probability decreasing in the distance of its difference embedding from the pool mean.

	5% verified		**20% verified**	
Judge	SCAR	SAR	SCAR	SAR
Mistral-7B, QTA	76.1% (1.0)	76.0% (1.7)	76.1% (1.6)	76.1% (1.2)
Mistral-7B, QTC	77.0% (1.8)	77.1% (1.4)	77.0% (1.1)	77.2% (0.8)
Mistral-7B, QTF	72.0% (2.1)	71.4% (4.4)	72.2% (1.3)	72.7% (1.3)
Llemma-7B, QTA	73.3% (1.8)	72.3% (6.8)	73.5% (2.0)	73.8% (3.1)
Llemma-7B, QTC	75.5% (3.2)	75.9% (2.7)	75.9% (2.4)	76.3% (2.2)
GPT-5.4-mini, MT-Bench	75.2% (6.5)	75.1% (9.4)	75.7% (2.8)	75.5% (2.7)

Table 11 replaces random verification by feature-dependent verification. Accuracy changes by at most 1.0 point, so PUAUDIT does not require the verified comparisons to be a random sample.

B.8 ALGORITHM

Algorithm 1 PUAUDIT Pipeline

Require: reward encoder F_θ, transport fraction $m \in (0, 1]$, threshold $\eta \in [0, 1]$, and pairwise dataset $\mathcal{D} = \{(x_i, o_i^A, o_i^B, \zeta_i^{(H)}, \zeta_i^{(LLM)})\}_{i=1}^N$.

Recall from Algorithm 2 that we have the PU partition $(\mathcal{D}_P, \mathcal{D}_U)$ and the chosen side c_i.
for each pair $i \in \mathcal{D}_P \cup \mathcal{D}_U$ **do**
    $\mathbf{e}_i^A \leftarrow F_\theta(x_i, o_i^A)$, $\mathbf{e}_i^B \leftarrow F_\theta(x_i, o_i^B)$;
    compute the difference embedding $z_i \leftarrow \phi(\mathbf{e}_i^A, \mathbf{e}_i^B)$, the normalized difference in Eq. (1) oriented by $\zeta_i^{(\mathrm{LLM})}$.
end for
Denoise the positive group: $\mathcal{D}'_P \leftarrow \mathrm{P}_{\mathrm{clean}}(\mathcal{D}_P)$.
Construct empirical measures (μ'_P, μ_U) using the embeddings $\{z_i\}$.
Compute the partial OT plan $\Gamma^\star$ with transport fraction m.
for each $j \in \mathcal{D}_U$ **do**
    compute the normalized score $\bar{s}_j \leftarrow \mathrm{Score}(\Gamma^\star, j)$.
end for
for each $j \in \mathcal{D}_U$ **do**
    if $\bar{s}_j < \eta$ **then**
        $\hat{\zeta}_j^{(LLM)} \leftarrow \mathrm{Flip}(\zeta_j^{(LLM)})$.
    else
        $\hat{\zeta}_j^{(LLM)} \leftarrow \zeta_j^{(LLM)}$.
    end if
end for
Output: corrected labels $\{\hat{\zeta}_j^{(LLM)}\}_{j \in \mathcal{D}_U}$ and scores $\{\bar{s}_j\}_{j \in \mathcal{D}_U}$.

Algorithm 2 PU data generation

Require: Pairwise comparison dataset $\mathcal{D} = \{(x_i, o_i^A, o_i^B, \zeta_i^{(H)})\}_{i=1}^N$; offline judge $\mathcal{A}_{\mathrm{LLM}}$.

Judge and form the PU pool.
for each comparison $i = 1, \ldots, N$ **do**
    Run the offline judge $\mathcal{A}_{\mathrm{LLM}}$ to obtain the predicted preference $\zeta_i^{(\mathrm{LLM})}$ and question type $\hat{o}_i^{qt}$:

$$(\zeta_i^{(\mathrm{LLM})}, \hat{o}_i^{qt}) \leftarrow \mathcal{A}_{\mathrm{LLM}}(x_i, o_i^A, o_i^B), \qquad \zeta_i^{(\mathrm{LLM})} \in \{A, B, C\}.$$

end for
Define the human-verified positive set and the unlabelled set:

$$\mathcal{D}_P := \{\, i : \zeta_i^{(H)} \text{ is observed and } \zeta_i^{(\mathrm{LLM})} = \zeta_i^{(H)} \,\}, \qquad \mathcal{D}_U := \{\, i : \zeta_i^{(H)} \text{ is missing} \,\}.$$

For each $i \in \mathcal{D}_P \cup \mathcal{D}_U$, set the chosen side $c_i \leftarrow \zeta_i^{(\mathrm{LLM})}$; for $i \in \mathcal{D}_P$ this coincides with $\zeta_i^{(H)}$.
Output: PU partition $(\mathcal{D}_P, \mathcal{D}_U)$, LLM judge labels $\{\zeta_i^{(\mathrm{LLM})}\}$, and chosen-side labels $\{c_i\}$.