Towards Acyclic Preference Evaluation of Language Models via Multiple Evaluators

Zhengyu Hu[1], **Jieyu Zhang**[2], **Zhihan Xiong**[2], **Alexander Ratner**[2], **Kaize Ding**[3], **Ranjay Krishna**[2]

[1] Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong SAR, China
[2] University of Washington, Seattle, WA, USA
[3] Northwestern University, Evanston, IL, USA

Abstract

Despite the remarkable success of Large Language Models (LLMs), evaluating their outputs' quality regarding *preference* remains a critical challenge. While existing works usually leverage a strong LLM as the judge for comparing LLMs' response pairwisely, such a single-evaluator approach is vulnerable to *cyclic preference* , i.e., output A is better than B, B than C, but C is better than A, causing contradictory evaluation results. To address this, we introduce PGED (Preference Graph Ensemble and Denoising), a novel approach that leverages multiple model-based evaluators to construct preference graphs, and then ensembles and denoises these graphs for acyclic, non-contradictory evaluation results. We provide theoretical guarantees for our framework, demonstrating its efficacy in recovering the ground truth preference structure. Extensive experiments on ten benchmarks demonstrate PGED's superiority in three applications: 1) model ranking for evaluation, 2) response selection for test-time scaling, and 3) data selection for model fine-tuning. Notably, PGED combines small LLM evaluators (e.g., Llama3-8B, Mistral-7B, Qwen2-7B) to outperform strong ones (e.g., Qwen2-72B), showcasing its effectiveness in enhancing evaluation reliability and improving model performance.

1 Introduction

Large Language Models (LLMs) have rapidly advanced various areas of artificial intelligence, particularly natural language processing and decision-making (Wu et al. 2023; Li et al. 2023a). As LLMs become increasingly capable, effective evaluation becomes critical (Siska et al. 2024; Boyeau et al. 2024; Chatzi et al. 2024). Among evaluation methods, preference evaluation plays a critical role in both evaluating and aligning LLMs (Rafailov et al. 2024; Yuan et al. 2024; Dubois et al. 2024b). A common practice is to rely on a strong LLM (e.g., GPT-4 (Achiam et al. 2023)) as the judge to conduct pairwise comparisons (Li et al. 2023b; Chen et al. 2023).

However, such model-based evaluator setups often lead to *cyclic preferences*, where inconsistent rankings emerge, for instance, preferring A over B, B over C, yet C over A (Naresh, Tulabandhula et al. 2024; Zhang, Yin, and Wan 2024). These cyclic patterns violate the transitivity assumption of preferences established in prior work (Ouyang et al. 2022; Song et al. 2024; Hou et al. 2024; Liu et al. 2024), thereby undermining the reliability of evaluation results. We model this *conflicting preference* using a *preference graph*, where nodes represent responses and directed edges indicate pairwise preferences. Cycles in such graphs (e.g., $A \succ B \succ C \succ A$) reflect evaluation noise, as shown in Figure 1(a). Ideally, a preference graph should be a directed acyclic graph (DAG) to maintain consistency. Empirically, we evaluated 10 Llama3-70B (AI@Meta 2024) responses on HumanEval (Chen et al. 2021) and MATH (Hendrycks et al. 2021), using GPT-4-o, GPT-4-o-mini, GPT-3.5 (Achiam et al. 2023), Qwen2-72B (Yang et al. 2024), and Llama3-8B as judges. Even with GPT-4-o, 64% of preference graphs in HumanEval and 38% in MATH contained cycles, (Figure 1(b)), demonstrating the persistent noise in LLM-based preference evaluations and motivating the need for a more robust approach.

To address this, we propose a novel framework, PGED (**P**reference **G**raph **E**nsemble and **D**enoising). Our method involves two key steps: (1) ensembling multiple preference evaluators to mitigate noise introduced by individual evaluators and (2) applying a denoising process to the resulting preference graph. By aggregating evaluations from multiple individual evaluators, we "average out" the noise and biases, resulting in a more robust approximation of the true preference structure. The denoising step further refines this aggregated graph by removing inconsistencies, ensuring the final preference graph is more reliable for downstream tasks. The overall process of PGED is illustrated in Figure 1 (c). We provide a theoretical analysis demonstrating the soundness of PGED, showing that by treating each individual preference graph as a random perturbation of a ground truth DAG, our ensemble and denoising framework can recover the ground truth DAG with high probability. To validate the practical efficacy of PGED, we conduct ten tasks ranging from response selection, model ranking to model alignment tasks, utilizing ten widely recognized benchmark datasets. In these experiments, PGED consistently outperformed baseline methods. Additionally, PGED demonstrated substantial gains in scenarios where combining preference graphs from small evaluators surpassed the performance of even stronger individual evaluators. Specifically, when using using Llama3-8B, Mistral-7B, and Qwen2-7B as evaluators, PGED exceeded the performance of using the Qwen2-72B in response selection task. These results highlight PGED's ability to mitigate preference

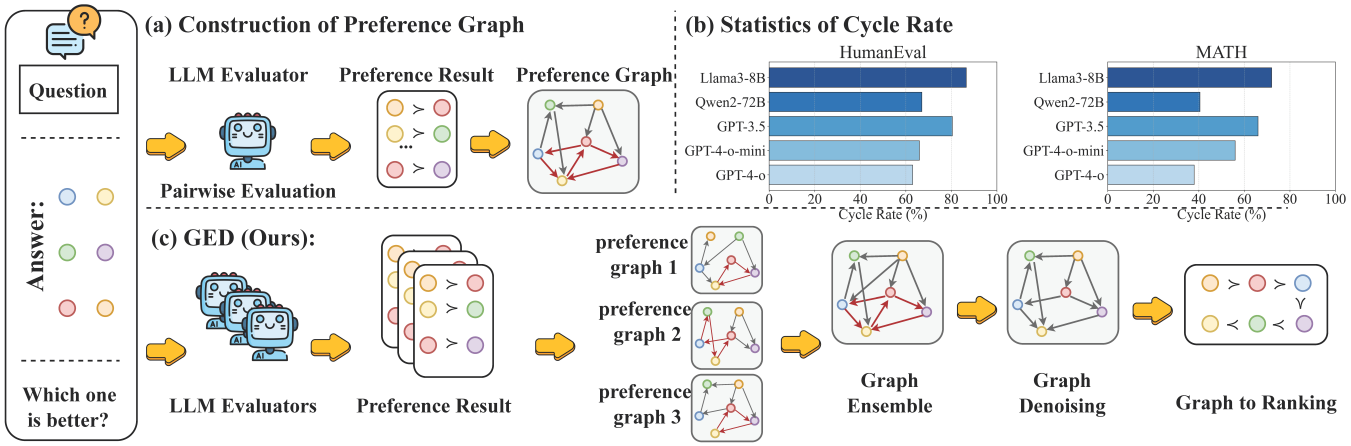

Figure 1: (a) A preference graph exhibiting cyclic inconsistencies (e.g., A $\succ$ B $\succ$ C $\succ$ A), which violate transitivity. (b) Empirical results showing that even advanced LLMs (e.g., GPT-4-o) exhibit significant noise in preference judgments, leading to inconsistent evaluations. (c) Overview of our proposed framework, PGED, which ensembles multiple preference evaluators and applies denoising to recover a directed acyclic graph.

noise, improve consistency, and enhance model performance across diverse evaluation settings. Our contributions are summarized as follows:

- We propose PGED, a novel framework that ensembles multiple preference evaluators and denoises the resulting preference graphs to produce acyclic and reliable evaluation outputs.
- We provide theoretical guarantees that PGED can recover the ground-truth preference DAG under a reasonable noise model, thereby offering provable consistency.
- We conduct extensive experiments on ten benchmark datasets and three key tasks, model ranking, response selection, and data selection for fine-tuning, demonstrating that PGED significantly improves evaluation robustness. Notably, PGED using small models outperforms even strong single evaluators like Qwen2-72B.

2 Related Work

Preference evaluation. Classical preference aggregation includes Elo, which iteratively updates item ratings from pairwise outcomes (Jiang, Ren, and Lin 2023); Bradley–Terry (BT/BTL) models, which assign latent utilities to explain win probabilities (Bradley and Terry 1952; Lu and Boutilier 2011); and MergeSort-style procedures, which sort items via pairwise comparisons with $O(n \log n)$ queries (Shyamasundar 1996). LLM-based preference evaluation increasingly relies on a strong model (e.g., GPT-4) as a zero-shot, reference-free judge of weaker models' outputs (Shen et al. 2023; Dubois et al. 2024b). PRD combines peer ranking over pairwise preferences with peer discussion between LLMs to reach consensus (Li, Patel, and Du 2023). ChatEval forms a multi-agent referee team that debates and evaluates responses for more reliable assessments (Chan et al. 2023). However, LLM judges can induce contradictory preferences and cycles when aggregating across outputs, yielding noisy and inconsistent preference graphs (Naresh, Tulabandhula et al. 2024; Zhang, Yin, and Wan 2024). SPA (Kadekodi, McTavish, and Ustun 2025) produces partial-order rankings that abstain on disputed pairs to balance comparability and disagreement, while ContraSolver (Zhang, Yin, and Wan 2024) builds a preference graph with self-annotations and flags edges likely responsible for contradictions. In contrast, our approach ensembles multiple evaluator-induced graphs and denoises them via a weighted feedback-arc-set objective to recover an acyclic structure suitable for robust ranking.

Weak supervision. The concept of weak supervision originates from the need to leverage noisy or partial labels in machine learning tasks, enabling the development of more robust models from imperfect data (Zhang, Song, and Ratner 2023). In LLMs, weak-to-strong supervision aids AI alignment by allowing weaker models to improve strong ones, enhancing performance without extensive data and supporting scalable oversight (Zheng et al. 2024; Guo and Yang 2024; Tong et al. 2024). Similarly, in task-oriented LLMs, weak-to-strong learning improves LLM's ability by enabling strong models to refine their data autonomously, boosting perfoArmance without extensive high-quality input (Yang, Ma, and Liu 2024). Through weak-to-strong supervision, LLM performance can be significantly improved by iteratively transforming low-quality labels into more reliable ones, leading to more effective model training and robust outputs (Zakershahrak and Ghodratnama 2024; Lang, Sontag, and Vijayaraghavan 2024). Recent work also shows weak LLMs can rival human feedback quality, enabling scalable and cost-efficient alignment (Tao and Li 2024; Li et al. 2025).

3 Preference Graph Ensemble and Denoising

In this section, we first introduce key definitions and assumptions underlying our approach, including the formal definition of a preference graph and the assumption of preference transitivity (Section 3.1). Building on these preliminaries, we then present our proposed framework, PGED, which con-

sists of three main steps: (1) *Graph Ensemble* (Section 3.2), aggregating multiple evaluators' preference graphs into a single unified structure; (2) *Graph Denoising* (Section 3.3), removing cycles to ensure an acyclic, consistent preference structure; and (3) *Graph-to-Ranking* (Section 3.4), converting the denoised graph into a reliable candidate ranking. The overall process of PGED is illustrated in Figure 1(c).

3.1 Preliminarily

Transitivity of Preferences. In preference evaluation, we assume that the *ground-truth* preference relation is transitive. That is, for any distinct $u, v, w \in V$, if $u \succ v$ and $v \succ w$, then it must hold that $u \succ w$:

$$(u \succ v) \wedge (v \succ w) \Longrightarrow (u \succ w). \tag{1}$$

This transitivity assumption ensures that pairwise comparisons can be consistently embedded into a global ranking.

Preference Graph. A preference graph is a directed graph $G_P = (V, A, w)$, where $V = \{v_1, v_2, \ldots, v_n\}$ represents n candidates, $A \subseteq V \times V$ is a set of directed arcs indicating pairwise preferences, and $w : A \to \mathbb{R}^+$ assigns weights to arcs, representing preference strength. For distinct $u, v \in V$, an arc $(u, v) \in A$ exists if $w(u, v) > 0$.

3.2 Graph Ensemble

Given k weighted preference graphs $G_i = (V, A_i, w_i)$ on a common vertex set V, define

$$\begin{aligned} A_E &\triangleq \Big\{(u,v) \in V \times V : \sum_{i=1}^{k} w_i(u,v) > 0\Big\}, \\ w_E(u,v) &\triangleq \sum_{i=1}^{k} w_i(u,v), \quad (u,v) \in A_E, \end{aligned} \tag{2}$$

where k is the total number of preference sources and $w_i(u, v)$ is the preference result from the i-th source and $w_i(u, v) = 0$ if $(u, v) \notin A_i$. The ensemble graph is $G_E = (V, A_E, w_E)$.

3.3 Graph Denoising

To ensure consistency in the aggregated preference graph, we apply a denoising step that transforms a potentially cyclic weighted digraph $G = (V, A, w)$ into a directed acyclic graph (DAG). We cast this as the classical *weighted feedback arc set* (WFAS) problem (Gabow 1995), which seeks a minimum-total-weight set of arcs whose removal renders the graph acyclic. Formally,

$$\begin{aligned} R^*(G) \;=\; \arg\min_{R \subseteq A} & \sum_{(u,v) \in R} w(u,v) \\ \text{s.t.} \quad & (V, A \setminus R) \text{ is acyclic.} \end{aligned} \tag{3}$$

A convenient formulation uses vertex sequencing. Given an ordering $s = \{v_1, \ldots, v_n\}$, the induced feedback arc set $R(s)$ comprises all arcs that violate the order, i.e., $(v_j \to v_i)$ with $j > i$. The denoising problem then reduces to finding s^* that minimizes the total weight of backward edges.

Algorithm 1: Preference Graph Denoising for PGED

Input: Weighted digraph $G = (V, A, w)$
Output: Denoised graph $G' = (V, A', w)$
Let $A_0 \leftarrow A$; initialize $s_1 \leftarrow [\,],\ s_2 \leftarrow [\,]$
while $V \neq \emptyset$ **do**
  while $\exists$ sink u in G **do**
    Prepend u to s_2
    Remove u and its incident arcs from G
  end while
  while $\exists$ source u in G **do**
    Append u to s_1
    Remove u and its incident arcs from G
  end while
  if $V = \emptyset$ **then break**
  end if
  Select $u = \arg\max_{v \in V} \left(d^+(v) - d^-(v)\right)$
  Append u to s_1
  Remove u and its incident arcs from G
end while
Concatenate $s = s_1 \| s_2$
Compute $R(s) = \{(v_j \to v_i) \in A_0 : j > i \text{ in } s\}$
Set $A' = A_0 \setminus R(s)$
return $G' = (V, A', w)$

Computing the optimal ordering is NP-hard (Karp 2010), so we adopt an efficient greedy procedure tailored to weighted graphs. Our algorithm iteratively builds a total order by removing structurally informative vertices. At each step, we first peel all *sinks* (zero out-degree) and place them at the end of the order, then peel all *sources* (zero in-degree) and place them at the beginning. When neither exists, we choose a vertex u maximizing the weighted degree difference

$$\delta(u) \;=\; d^+(u) - d^-(u), \tag{4}$$

where $d^+(u) = \sum_{(u,v) \in A} w(u, v)$ and $d^-(u) = \sum_{(v,u) \in A} w(v, u)$. Repeating until all vertices are removed yields a total order; the backward arcs under this order form an approximate feedback arc set $R(s)$. We then construct and *return only* the denoised graph $G' = (V, A \setminus R(s), w)$, which is a subgraph of the input and is typically sparse rather than fully connected. The full procedure is summarized in Algorithm 1. Its theoretical guarantee can be seen in Appendix M of (Hu et al. 2024b).

3.4 Graph to Ranking

Given a DAG $G = (V, A, w)$, we derive a ranking by computing the descendant count $\mathrm{desc}(v)$ for each vertex v, defined as the number of vertices reachable from v:

$$\mathrm{desc}(v) = \left|\{u \in V : v \to u\}\right|, \tag{5}$$

where $v \to u$ denotes a directed path. Vertices are ranked based on $\mathrm{desc}(v)$, with ties broken lexicographically:

$$v_1 \succ v_2 \succ \cdots \succ v_n. \tag{6}$$

This ranking reflects both individual preferences and their relative strengths in the graph.

4 Downstream Tasks

We apply PGED to three tasks: Response Selection (selecting the best response from LLM-generated candidates), Model Ranking (ranking models based on task performance), and Model Alignment (identifying the best instruction-response pairs for training). Details on how preference graphs are constructed for each of the three settings can be found in Appendix H of (Hu et al. 2024b).

4.1 Response Selection

For each question $q \in Q$, a model $\mathcal{M}$ generates n candidate answers $C_q = \{ans_1, \ldots, ans_n\}$. A set of evaluators $\mathcal{A} = \{a_1, \ldots, a_k\}$ provides pairwise preferences over C_q. For each evaluator a, we construct a weighted preference graph $G_a^q = (V_q, A_a, w_a)$, where $V_q = \{v_1, \ldots, v_n\}$ indexes candidates and $w_a(u, v)$ accumulates the number of wins of u over v (ties are ignored). We then apply PGED: (i) *Graph Ensemble* aggregates $\{G_a^q\}_{a \in \mathcal{A}}$ into a single graph G_E^q (Section 3.2); (ii) *Graph Denoising* removes a minimum-weight set of feedback arcs to produce a DAG G_q' (Section 3.3); and (iii) *Graph-to-Ranking* converts G_q' into a total order $\mathcal{R}_q = \{v_1 \succ \cdots \succ v_n\}$ (Section 3.4). The top-ranked candidate is selected as ans_q^*. Repeating this for all $q \in Q$ yields $ans^* = \{ans_1^*, \ldots, ans_t^*\}$.

4.2 Model Ranking

Given a set of models $M = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$ and questions $Q = \{q_1, \ldots, q_t\}$, our goal is to produce a global ranking of M. For each question $q \in Q$ and evaluator $a \in \mathcal{A}$, we construct a weighted preference graph $G_a^q = (V_q, A_a, w_a)$, where $V_q = \{m_1, \ldots, m_n\}$ indexes model outputs $\mathcal{M}_i(q)$ and $w_a(u, v)$ accumulates wins of u over v (ties are ignored). We then apply PGED per question: (i) *Graph Ensemble* aggregates $\{G_a^q\}_{a \in \mathcal{A}}$ into G_E^q (Section 3.2); (ii) *Graph Denoising* yields a DAG G_q' (Section 3.3); and (iii) *Graph-to-Ranking* returns a total order $\mathcal{R}_q$ over V_q (Section 3.4). Finally, we aggregate $\{\mathcal{R}_q : q \in Q\}$ into an overall model ranking $\mathcal{R}^*$ using a ranking-ensemble procedure (Appendix Q of (Hu et al. 2024b)).

4.3 Model Alignment

For each instruction $x \in X$, let the candidate set be $Y_x = \{y_1, \ldots, y_n\}$. Evaluators $\mathcal{A}$ provide pairwise preferences over Y_x. For each $a \in \mathcal{A}$, we build a weighted preference graph $G_a^x = (V_x, A_a, w_a)$, where $V_x = \{v_1, \ldots, v_n\}$ indexes responses and $w_a(u, v)$ aggregates wins of u over v (ties ignored). Applying PGED, ensemble, denoising, and graph-to-ranking, produces a total order $\mathcal{R}_x$; we select the top response y_x^* for instruction x. Repeating this for all $x \in X$ yields the aligned training set $\{(x, y_x^*) : x \in X\}$.

5 Theoretical Analysis

In this section, we provide a theoretical foundation for our method, showing that by modeling preference graphs as random perturbations of a ground truth DAG, PGED can reliably recover the true structure through graph ensemble and denoising with high probability, demonstrating its robustness in handling noisy evaluations. Theoretically, we treat each of our preference graph as a random perturbation of some ground truth DAG $G = (V, A)$. Specifically, we consider a random graph generator $\mathcal{G}(G, \delta_1, \delta_2)$ with parameters $\delta_1, \delta_2 \in [0, 1]$ such that $G_i = (V_i, A_i) \sim \mathcal{G}(G, \delta_1, \delta_2)$ satisfies $V_i = V$.

Furthermore, for each $u, v \in V$ with $u \neq v$,

1) If $(u \to v) \in A$, then

$$\mathbb{P}((u \to v) \in A_i) = 1 - \delta_1,$$
$$\mathbb{P}((v \to u) \in A_i) = \delta_1.$$

2) If $(u \to v), (v \to u) \notin A$, then

$$\mathbb{P}((u \to v), (v \to u) \notin A_i) = 1 - \delta_2,$$
$$\mathbb{P}((u \to v) \in A_i) = \frac{\delta_2}{2},$$
$$\mathbb{P}((v \to u) \in A_i) = \frac{\delta_2}{2}.$$

That is, each edge in E has probability δ_1 of being flipped and each pair of unconnected nodes has probability δ_2 of being connected with a random direction.

Now, given that $G_1, \ldots, G_N \overset{\text{i.i.d.}}{\sim} \mathcal{G}(G, \delta_1, \delta_2)$, we will show that to some extent our combination of graph ensemble and graph denoising can indeed provably recover the ground truth DAG G. For simplicity, all edges in $G_1, \ldots, G_N$ and G are considered equal weighted. Meanwhile, we use $\text{MAS}(\cdot)$ to denote the graph obtained by denoising, which stands for the maximum acyclic subgraph (MAS). Then, we have the following theorem.

Theorem 1 *Suppose $G_1, \ldots, G_N \overset{\text{i.i.d.}}{\sim} \mathcal{G}(G, \delta_1, \delta_2)$ for some ground truth $G = (V, A)$. Let $\widehat{G}$ be the graph ensembled from $G_1, \ldots, G_N$ by operations defined in Section 3. Then, as long as $\delta_1 = 0.5 - \epsilon$ for some $\epsilon > 0$, we have*

$$\mathbb{P}\left(G \subseteq \textit{MAS}(\widehat{G})\right) \geq 1 - 2|A| \exp\left(-\frac{N\epsilon^2}{2}\right) - 2U \exp\left(-\frac{N\epsilon^2}{6U^2\delta_2 + 2U\epsilon}\right),$$

where $G \subseteq \textit{MAS}(\widehat{G})$ represents that G is a subgraph of $\textit{MAS}(\widehat{G})$ and $U = \frac{|V|(|V|-1)}{2} - |A|$ is the number of pairs of unconnected nodes in G.

The full proof is given in Appendix **??**. From the theorem, we can see that the probability of failure decreases exponentially as the number of samples N increases. Meanwhile, this guarantee only requires $\delta_1 < 0.5$ and does not place restrictions on δ_2, which are very mild conditions.

6 Response Selection for Test-time Scaling

Experiment Setup. In this section, we evaluate the performance of PGED on five benchmarks: HumanEval (Chen et al. 2021), AlpacaEval (Li et al. 2023b), MATH (Hendrycks et al. 2021), GSM8k (Chen et al. 2021), and GAIA (Mialon et al. 2023). The Qwen2-72B (Yang et al. 2024) model ($\mathcal{M}$) generates ten candidate responses per question, and we assess the effectiveness of different methods in selecting the

Method		HumanEval	AlpacaEval	MATH	GSM8k	GAIA	Avg
Single model	Llama3-8B	43.90	27.29	22.08	56.67	6.78	31.34
	Mistral-7B	23.17	11.80	23.25	39.83	7.03	21.01
	Qwen2-7B	48.58	25.71	59.92	76.75	7.70	43.73
	Qwen2-72B	57.93	29.58	72.75	84.67	11.52	51.29
	ContraSolver (Qwen2-72B)	65.42	31.12	74.95	86.84	12.22	54.11
	ListPreference	61.52	31.67	71.75	85.0	10.90	52.16
	Self-consistency	60.98	29.33	73.58	84.91	8.86	51.53
Single evaluator	Elo	62.21	31.34	75.01	87.21	12.36	53.63
	Bradley-Terry	62.83	32.91	74.97	86.64	12.07	53.88
	Merge Sort	62.81	31.88	74.83	86.90	12.13	53.71
	Llama3-8B	62.19	29.31	74.27	83.16	11.31	52.04
	with graph denoising	64.02	30.18	74.73	86.00	11.72	53.33
	Mistral-7B	67.24	27.70	74.41	83.83	10.50	52.73
	with graph denoising	68.73	29.93	74.77	83.91	10.74	53.61
	Qwen2-7B	61.58	28.69	74.50	85.41	11.11	52.25
	with graph denoising	65.85	29.44	74.79	86.38	11.25	53.54
	Qwen2-72B	60.97	31.04	74.73	86.47	12.14	53.07
	with graph denoising	68.90	31.17	75.33	87.45	12.26	55.02
Multiple evaluator	Majority Voting	66.18	29.57	74.77	86.42	11.72	53.73
	Weight Majority Voting	66.43	30.12	74.79	86.68	11.84	53.97
	PRD	66.53	30.33	74.89	86.75	11.83	54.07
	ChatEval	66.51	30.26	74.79	86.85	11.92	54.07
	SPA	66.32	30.05	74.82	86.57	11.81	53.91
	PGED(w/o denoising)	69.25	30.98	74.29	87.17	12.68	54.87
	PGED	**71.86**	**32.95**	**76.57**	**89.76**	**13.74**	**56.98**

Table 1: Performance comparison of response selection methods across five benchmarks. PGED consistently outperforms baseline methods, highlighting the effectiveness of graph denoising and multi-evaluator aggregation. *Single model* denotes directly using the evaluator to answer the question. *Single evaluator* and *Multi evaluator* refer to selecting the best response from ten Qwen2-72B-generated candidates using a single or multiple evaluators, respectively.

best response. We evaluate performance using three setups. First, in the *single model* setting, the baselines include ContraSolver(Zhang, Yin, and Wan 2024), Self-consistency(Wang et al. 2022), and direct evaluation with models (Llama3-8B, Mistral-7B, Qwen2-7B and Qwen2-72B). Additionally, we include a baseline called ListPreference, where instead of pairwise comparisons, all candidate responses are input into Qwen2-72B for selecting the most appropriate response. Then, in the *single evaluator* setting, individual evaluators (Llama3-8B, Mistral-7B, Qwen2-7B, Qwen2-72B) select the best response from $\mathcal{M}$'s outputs, with and without applying PGED's graph denoising. We additionally report results for three classical baselines in this setting: Elo (Jiang, Ren, and Lin 2023), Bradley–Terry (BTL) (Bradley and Terry 1952; Lu and Boutilier 2011), and MergeSort (Shyamasundar 1996). Finally, in the *multiple evaluators* setup, we combine three small evaluators (Llama3-8B, Qwen2-7B, Mistral-7B) to select responses from Qwen2-72B with PGED. We additionally include four aggregation baselines: Majority Voting, Weighted Majority Voting, PRD (Li, Patel, and Du 2023), ChatEval (Chan et al. 2023), and SPA (Kadekodi, McTavish, and Ustun 2025). We present the results of PGED and its variant (w/o denoising), which ensembles the preference graphs without the denoising step.

Note that, among the baselines we report: (i) in the single model setting, ContraSolver, ListPreference, and Self-consistency use Qwen2-72B as the base model; (ii) in the single evaluator setting, Elo, Bradley–Terry, and MergeSort use Qwen2-72B as the base model; and (iii) in the multiple evaluators setting, Majority Voting, Weighted Majority Voting, PRD, ChatEval and SPA use Llama3-8B, Mistral-7B, and Qwen2-7B as the base model set. For details on the

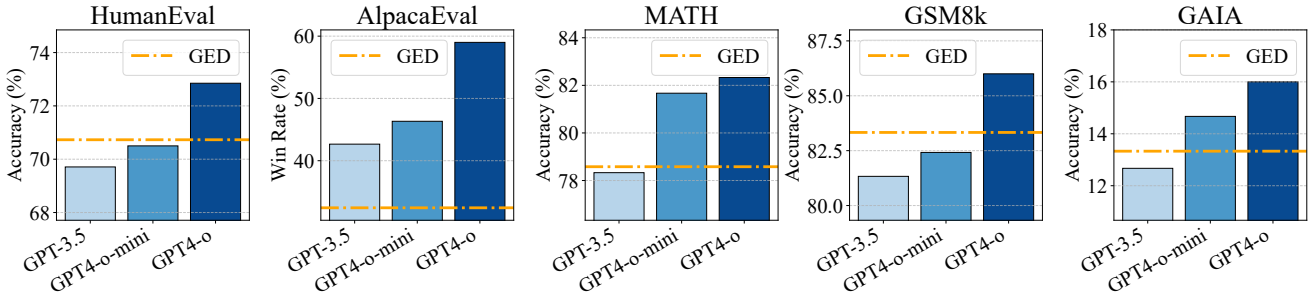

Figure 2: Comparison of PGED with GPT-3.5, GPT-4-o-mini, and GPT-4-o on 100 randomly selected tasks. PGED consistently outperforms GPT-3.5 across all tasks and surpasses GPT-4-o-mini on challenging tasks like HumanEval and GSM8k, showcasing the effectiveness of weak evaluator aggregation with graph denoising.

datasets and baselines, please refer to Appendix A of (Hu et al. 2024b).

Main results. Table 1 presents the results of the response selection task across five benchmarks. PGED consistently outperforms all baseline methods, including single model evaluations (*single model*), direct response selection by individual models (*single evaluator*) and Multiple evaluator. This demonstrates the strength of aggregating preference evaluators with PGED, particularly when coupled with graph denoising, which enhances response quality by filtering out noise and biases. Furthermore, by combining preference graphs derived from smaller models (Llama3-8B, Mistral-7B, Qwen2-7B), PGED outperforms a much larger evaluator (Qwen2-72B). This underscores the value of ensemble methods in mitigating the limitations of individual evaluators. Then, the denoising process proves to be crucial for improving consistency and overall response quality. The substantial performance gains observed when using PGED with denoising, compared to both the single evaluator setup and the ensemble without denoising, highlight its importance in refining response selection. For Majority Voting, while it improves upon individual evaluators, it still underperforms PGED, highlighting PGED's ability to capture nuanced evaluation signals and reduce inconsistencies. Additionally, we observed that the ListPreference baseline performed worse than Qwen2-72B as single evaluator, likely due to LLM limitations in handling long-text. Lastly, to further evaluate PGED, we compared its performance with GPT-3.5, GPT-4-o-mini, and GPT-4-o. Due to computational and API cost constraints, we limited the evaluation to 100 data points for each task. As shown in Figure 2, PGED consistently outperformed GPT-3.5 across all tasks and surpassed GPT-4-o-mini on challenging benchmarks like HumanEval and GSM8k. These results highlight the superiority of PGED, particularly in leveraging multi-preference evaluators and graph denoising to outperform individual state-of-the-art models. More discussion on cost is provided in Appendix R of (Hu et al. 2024b).

7 Model Ranking for Evaluation

Experiment Setup. In this section, we evaluate the effectiveness of PGED in the model ranking task within a human preference setting, using the AlpacaEval 2.0 (Li et al. 2023b). We employ 30 widely used models from the AlpacaEval dataset as our model set $\mathcal{M}$, while the benchmark's questions form the question set Q. The rankings provided by the AlpacaEval benchmark serve as ground truth for evaluating the accuracy of various ranking methods. This is justified by AlpacaEval's strong correlation with Chatbot Arena rankings, making it a reasonable proxy for human judgments (Dubois et al. 2024a). We adopt Ranking Correction, measured by the Spearman rank correlation coefficient, to evaluate the similarity. To generate rankings, we utilize outputs from the open-source models Llama3-70B, Qwen2-72B, Mistral-8$\times$7B, and Qwen1.5-72B as our evaluators. We investigate two variants of PGED: (w/o ensemble) denoises the preference graphs from different evaluators for the same question, converts each into a ranking, and then ensembles these rankings to produce the final output, while (w/o denoising) directly ensembles the preference graphs to obtain the final ranking without denoising. For details on the datasets and baselines, please refer to Appendix A of (Hu et al. 2024b).

Main results. The results, presented in Table 2, show that PGED outperforms all single-model baselines, highlighting the significant improvement in ranking accuracy achieved by leveraging preference information from multiple evaluators. Moreover, PGED surpasses the (w/o ensemble) variant, indicating that generating rankings through graph ensemble first prevents information loss compared to converting individual graphs into rankings. When the ensemble graph is not denoised (w/o denoising), residual noise can adversely affect the final ranking quality. Additionally, our denoising method also enhances results in single-model settings.

8 Data Selection for Model Fine-tuning

Experiment Setup. In this section, we explore the effects of various data selection methods for model alignment on Llama-2-7B (Touvron et al. 2023) and Mistral-7B (Jiang et al. 2023) through instruct tuning. Specifically, we randomly sampled 5,000 data points from UltraFeedback (Cui et al. 2023) and used Qwen1.5-14B (Yang et al. 2024) to generate eight responses per data point as instruct data. We then applied four different methods, Random, Longest (Zhao et al. 2024), ContraSolver (using Qwen2-72B as the evaluator) (Zhang, Yin, and Wan 2024), and our proposed PGED, which lever-

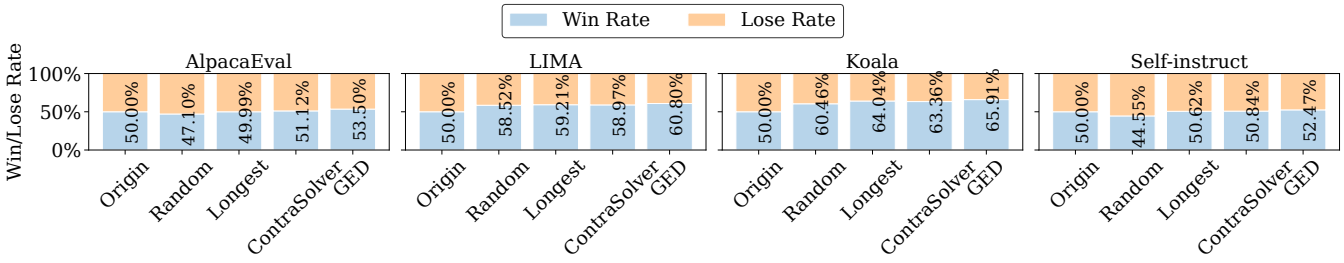

Figure 3: Performance comparison of different methods (Random, Longest, ContraSolver, and PGED) across multiple benchmarks. The results show PGED effectively filters low-quality responses, improving performance and model alignment over baselines.

Model		Weight Score	Kemeny	Weighted Kemeny	Pairwise Majority	Weighted Pairwise Majority	Avg.
Single evaluator	Llama3-70B	50.88	60.80	60.80	62.23	61.85	59.31
	with graph denoising	52.44	62.54	62.54	63.92	62.18	60.72
	Qwen2-72B	65.34	59.87	67.39	66.05	66.59	65.04
	with graph denoising	66.05	70.43	70.43	72.32	72.41	70.32
	Qwen1.5-72B	63.64	60.72	60.72	62.65	63.28	62.20
	with graph denoising	64.81	61.77	61.77	64.36	64.76	63.49
	Mistral-8×7B	64.90	68.74	68.74	73.06	72.87	69.66
	with graph denoising	65.47	70.06	69.92	73.39	73.21	70.41
Multiple evaluator	PGED(w/o ensemble)	62.82	68.44	68.44	69.34	67.34	67.27
	PGED(w/o denoising)	64.84	69.23	69.81	75.35	74.37	70.72
	PGED	**66.59**	**71.14**	**71.14**	**77.17**	**76.46**	**72.50**

Table 2: Results of the model ranking task, evaluated using Ranking Correction. Higher correlation values indicate a stronger alignment with the ground truth rankings.

ages Llama3-8B, Mistral-7B, and Qwen2-7B as evaluators, to select a subset of these responses for model alignment training. The Origin refers to the performance of the base model without alignment. To ensure a comprehensive assessment, we evaluated the models, using the Llama-2-7B backbone, on additional benchmarks, including AlpacaEval 2.0 (Li et al. 2023b), LIMA (Zhou et al. 2023), Koala (Vu et al. 2023), and Self-Instruct (Wang et al. 2022), in accordance with recent studies (Chen et al. 2023; Zhang et al. 2024; Hu et al. 2024a). The corresponding results are summarized in Figure 3. For details on the datasets and baselines, please refer to Appendix A of (Hu et al. 2024b).

Main results. From Figure 3, we observe that PGED consistently outperforms all baseline methods, demonstrating its effectiveness in selecting high-quality responses when multiple answers are available for a given instruction. PGED consistently outperforms all baselines across various datasets, demonstrating its effectiveness in selecting high-quality responses. Notably, in AlpacaEval and Self-Instruct, the Random baseline performs worse than the Origin model, highlighting that when response quality varies significantly, poor selection can degrade model performance. In contrast, PGED leverages preference graphs and denoising techniques to filter out low-quality responses, ensuring more robust and reliable performance, particularly in settings with inconsistent responses, as it removes evaluation noise and leads to more robust performance.

9 Conclusion

In this paper, we presented PGED, a framework designed to address inconsistencies in pairwise preference evaluations by LLMs. By employing graph ensemble techniques and denoising, PGED reduces cyclic patterns and enhances the reliability of evaluation outcomes. Our theoretical analysis shows that PGED can recover the ground truth DAG under reasonable conditions, improving consistency in preference rankings. Extensive experiments across response selection, model ranking, and instruct tuning demonstrate the efficacy of our method. PGED consistently outperformed baseline methods in both single-evaluator and multi-evaluator settings, particularly in scenarios where combining small evaluators led to superior results over larger individual evaluators. Future work will explore extending PGED to broader evaluation frameworks and applying its principles to more complex decision-making tasks, including multi-agent systems and human-AI interaction.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 Model Card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md. Accessed: 2025-12-12.

Boyeau, P.; Angelopoulos, A. N.; Yosef, N.; Malik, J.; and Jordan, M. I. 2024. AutoEval Done Right: Using Synthetic Data for Model Evaluation. *arXiv preprint arXiv:2403.07008*.

Bradley, R. A.; and Terry, M. E. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*.

Chan, C.-M.; Chen, W.; Su, Y.; Yu, J.; Xue, W.; Zhang, S.; Fu, J.; and Liu, Z. 2023. ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate. *ArXiv*.

Chatzi, I.; Straitouri, E.; Thejaswi, S.; and Rodriguez, M. G. 2024. Prediction-Powered Ranking of Large Language Models. *arXiv preprint arXiv:2402.17826*.

Chen, L.; Li, S.; Yan, J.; Wang, H.; Gunaratna, K.; Yadav, V.; Tang, Z.; Srinivasan, V.; Zhou, T.; Huang, H.; and Jin, H. 2023. AlpaGasus: Training A Better Alpaca with Fewer Data. *arXiv preprint arXiv:2307.08701*.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. D. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Cui, G.; Yuan, L.; Ding, N.; Yao, G.; Zhu, W.; Ni, Y.; Xie, G.; Liu, Z.; and Sun, M. 2023. UltraFeedback: Boosting Language Models with High-quality Feedback. arXiv:2310.01377.

Dubois, Y.; Galambosi, B.; Liang, P.; and Hashimoto, T. B. 2024a. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Dubois, Y.; Li, C. X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P. S.; and Hashimoto, T. B. 2024b. Alpacafarm: A simulation framework for methods that learn from human feedback. *Neurips*.

Gabow, H. N. 1995. Centroids, representations, and submodular flows. *Journal of Algorithms*.

Guo, Y.; and Yang, Y. 2024. Improving Weak-to-Strong Generalization with Reliability-Aware Alignment. *arXiv preprint arXiv:2406.19032*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hou, Y.; Zhang, J.; Lin, Z.; Lu, H.; Xie, R.; McAuley, J.; and Zhao, W. X. 2024. Large language models are zero-shot rankers for recommender systems. In *ECIR*.

Hu, Z.; Song, L.; Zhang, J.; Xiao, Z.; Wang, J.; Chen, Z.; Zhao, J.; and Xiong, H. 2024a. Rethinking LLM-based Preference Evaluation. *arXiv preprint arXiv:2407.01085*.

Hu, Z.; Zhang, J.; Xiong, Z.; Ratner, A.; Xiong, H.; and Krishna, R. 2024b. Language model preference evaluation with multiple weak evaluators. *arXiv preprint arXiv:2410.12869*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Jiang, D.; Ren, X.; and Lin, B. Y. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. In *ACL*.

Kadekodi, S.; McTavish, H.; and Ustun, B. 2025. Selective Preference Aggregation. In *2nd Workshop on Models of Human Feedback for AI Alignment*.

Karp, R. M. 2010. *Reducibility among combinatorial problems*. Springer.

Lang, H.; Sontag, D.; and Vijayaraghavan, A. 2024. Theoretical Analysis of Weak-to-Strong Generalization. *arXiv preprint arXiv:2405.16043*.

Li, G.; Hammoud, H. A. A. K.; Itani, H.; Khizbullin, D.; and Ghanem, B. 2023a. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In *Neurips*.

Li, R.; Patel, T.; and Du, X. 2023. PRD: Peer Rank and Discussion Improve Large Language Model based Evaluations. *Trans. Mach. Learn. Res.*

Li, X.; Zhang, T.; Dubois, Y.; Taori, R.; Gulrajani, I.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023b. AlpacaEval: An Automatic Evaluator of Instruction-following Models. https://github.com/tatsu-lab/alpaca_eval.

Li, Y.; Miao, X.; Xu, M.; and Qian, T. 2025. Strong Empowered and Aligned Weak Mastered Annotation for Weak-to-Strong Generalization. In *AAAI*.

Liu, Y.; Zhou, H.; Guo, Z.; Shareghi, E.; Vulic, I.; Korhonen, A.; and Collier, N. 2024. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *arXiv preprint arXiv:2403.16950*.

Lu, T.; and Boutilier, C. 2011. Learning Mallows Models with Pairwise Preferences. In *International Conference on Machine Learning*.

Mialon, G.; Fourrier, C.; Swift, C.; Wolf, T.; LeCun, Y.; and Scialom, T. 2023. Gaia: a benchmark for general ai assistants. *arXiv preprint arXiv:2311.12983*.

Naresh, N. U.; Tulabandhula, T.; et al. 2024. CURATRON: Complete and Robust Preference Data for Rigorous Alignment of Large Language Models. In *DaSH 2024*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Neurips*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Neurips*, 36.

Shen, W.; Zheng, R.; Zhan, W.; Zhao, J.; Dou, S.; Gui, T.; Zhang, Q.; and Huang, X. 2023. Loose lips sink ships: Mitigating Length Bias in Reinforcement Learning from Human Feedback. In *EMNLP*.

Shyamasundar, R. K. 1996. Introduction to algorithms. *Resonance*.

Siska, C.; Marazopoulou, K.; Ailem, M.; and Bono, J. 2024. Examining the robustness of LLM evaluation to the distributional assumptions of benchmarks. In *ACL*.

Song, F.; Yu, B.; Li, M.; Yu, H.; Huang, F.; Li, Y.; and Wang, H. 2024. Preference ranking optimization for human alignment. In *AAAI*.

Tao, L.; and Li, Y. 2024. Your Weak LLM is Secretly a Strong Teacher for Alignment. *arXiv preprint arXiv:2409.08813*.

Tong, Y.; Wang, S.; Li, D.; Wang, Y.; Han, S.; Lin, Z.; Huang, C.; Huang, J.; and Shang, J. 2024. Optimizing Language Model's Reasoning Abilities with Weak Supervision. *arXiv preprint arXiv:2405.04086*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vu, T.-T.; He, X.; Haffari, G.; and Shareghi, E. 2023. Koala: An Index for Quantifying Overlaps with Pre-training Corpora. *arXiv preprint arXiv:2303.14770*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Zhang, S.; Zhu, E.; Li, B.; Jiang, L.; Zhang, X.; and Wang, C. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. *ArXiv*, abs/2308.08155.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yang, Y.; Ma, Y.; and Liu, P. 2024. Weak-to-strong reasoning. *arXiv preprint arXiv:2407.13647*.

Yuan, W.; Pang, R. Y.; Cho, K.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Zakershahrak, M.; and Ghodratnama, S. 2024. Explanation, Debate, Align: A Weak-to-Strong Framework for Language Model Generalization. *arXiv preprint arXiv:2409.07335*.

Zhang, J.; Song, L.; and Ratner, A. 2023. Leveraging instance features for label aggregation in programmatic weak supervision. *PMLR*.

Zhang, Q.; Zhang, Y.; Wang, H.; and Zhao, J. 2024. RECOST: External Knowledge Guided Data-efficient Instruction Tuning. *arXiv preprint arXiv:2402.17355*.

Zhang, X.; Yin, X.; and Wan, X. 2024. ContraSolver: Self-Alignment of Language Models by Resolving Internal Preference Contradictions. *arXiv preprint arXiv:2406.08842*.

Zhao, H.; Andriushchenko, M.; Croce, F.; and Flammarion, N. 2024. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. *arXiv preprint arXiv:2402.04833*.

Zheng, C.; Wang, Z.; Ji, H.; Huang, M.; and Peng, N. 2024. Weak-to-strong extrapolation expedites alignment. *arXiv preprint arXiv:2404.16792*.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.