

InterPol: De-anonymizing LM Arena via Interpolated Preference Learning

Minsung Cho Jaehyung Kim

Yonsei University

{mscho2008, jaehyungk}@yonsei.ac.kr

Abstract

Voting-based leaderboards such as LM Arena are central to open-ended LLM evaluation, but their reliability depends on response anonymity. Prior work shows that surface features such as bag-of-words and TF-IDF can reveal model identity from the response, yet such signals are unreliable when models are stylistically similar or closely matched, and are easily down-weighted during vote aggregation. To examine the remaining anonymity risk, we introduce INTERPOL, a model-driven identification framework that learns target-specific response signatures from interpolated preference data. INTERPOL synthesizes controllable hard negatives by interpolating a target-mimicking copy model with its initial backbone, and trains a target-specific detector via curriculum-based triplet preference learning. Across three frontier target LLMs and two query sources, INTERPOL yields average relative gains of 32% in Accuracy and 25% in AUROC over prior methods. Moreover, in arena-style promotion simulations, INTERPOL attains the same ranking gain with about 19% fewer interactions than existing identification baselines; with competitor-aware voting, interactions are further reduced by about 69.9% relative to target-only voting.

1 Introduction

The rapid emergence of Large Language Models (LLMs) (Touvron et al., 2023; Achiam et al., 2023; Team et al., 2023) has made their fair and reliable evaluation a central challenge. Conventional answer-based benchmarks (Chen, 2021; Hendrycks et al., 2020) neither reflect the diversity of real-world usage (Liang et al., 2022) nor fairly assess open-ended tasks that admit multiple valid responses (Zheng et al., 2023), driving a shift toward evaluation pipeline grounded in user preferences (Stiennon et al., 2020; Zhou et al., 2023; Dubois et al., 2024). One of the most representative examples of this paradigm is voting-based

leaderboard such as LM Arena (Zheng et al., 2023; Chiang et al., 2024): users are shown anonymous responses from two LLMs to the same prompt, and their judgments are aggregated into scores such as Elo ratings (Boubdir et al., 2024). The core assumption here is the *anonymity of model responses*: each judgment is presumed to depend solely on response quality rather than the identity of the source model (Panickssery et al., 2024). If this assumption is violated through identification signals embedded in the responses, coordinated manipulation targeting a specific LLM becomes feasible, undermining the credibility of the leaderboard (Singh et al., 2025).

Therefore, investigating the feasibility of LLM identification is critical to the reliability of evaluation with leaderboard, yet this line of work has only begun to be explored (Min et al., 2025; Huang et al., 2025). For instance, Huang et al. (2025) show that even simple statistical features, such as Bag-of-Words (Salton et al., 1975) or TF-IDF representations (Salton and Buckley, 1988), can identify LLMs with high accuracy. However, manipulations built on such surface-level features can be systematically mitigated; for example, defenders can down-weight suspicious votes by exploiting the same statistical signals during leaderboard aggregation (LM Arena Team, 2024a,b). While this defensibility alleviates some concerns, it raises a key question: *do more advanced de-anonymization (i.e., identification) techniques exist that could threaten the robustness and credibility of leaderboards?*

In this paper, we answer this question affirmatively by proposing **INTERPOL**, a model-driven LLM identification framework that learns from **InterPolated** preference data. Our key idea is to train a target-specific detector that scores a target LLM’s responses higher (*i.e.*, prefer) than other LLMs’, using only queries and responses collected from multiple LLMs without metadata or declared identity. However, responses naively collected from other LLMs are often easy to distinguish, provid-

ing limited signal for training a strong detector. To this end, we synthesize hard negatives via *model interpolation* (Wortsman et al., 2022; Ilharco et al., 2022): we first fine-tune a separate copy model to mimic the target LLM, then construct an interpolated model between this fine-tuned model and the base model. INTERPOL uses the responses from the interpolated model as hard negatives for preference learning, where the interpolation ratio serves as an explicit control over their difficulty. With this data, INTERPOL trains the detector on the inherent preference order within triplets of ⟨target, interpolated, other LLMs⟩ responses, using adaptive curriculum learning that gradually increases the interpolation ratio to expose the detector to increasingly harder negatives (see Figure 1).

To evaluate INTERPOL, we first conduct target LLM identification experiments across three frontier target LLMs (GPT-4o, Sonnet-4, and Gemini-2.5-Pro) and two query sources (Section 4). Across these settings, INTERPOL consistently outperforms six representative baselines in both identification accuracy and AUROC; specifically, relative to the average of prior baselines, INTERPOL improves accuracy/AUROC by 32.3%/25.0%, respectively. Its advantage also persists in more challenging setups of close-Elo and within-family comparisons, where coarse capability gaps, model-scale differences, and surface-level statistical cues are less informative for attribution.

We then connect this identifiability to its practical impact through ranking manipulation simulations on Arena-style battle data (Section 5). Using reconstructed leaderboard dynamics, we quantify the number of adversarial votes and total interactions required to move target models across ranks. Across representative promotion settings, INTERPOL reduces the required interaction budget by 19% on average compared to existing identification baselines. Moreover, when voting strategy includes downvotes against competitors, required interaction further decreases by about 69.9% compared with a target-only voting strategy. Overall, our results show that response identifiability is not merely an offline attribution signal, but a concrete reliability risk that can lower the cost of strategic voting and alter leaderboard outcomes.

In summary, our contributions are threefold.

- We reveal an anonymity risk beyond surface cues: anonymous LLM responses retain target-specific signatures that enable source attribution.

- We propose INTERPOL, which learns these signatures using interpolated hard negatives and triplet preference learning with an adaptive curriculum.
- We demonstrate that this identifiability is robust across query sources and harder competitor settings, and that it reduces the interaction cost of strategic voting in Arena-style simulations.

2 Related Works

LLM evaluation and preference leaderboards.

Early LLM evaluation relied on static benchmarks such as GLUE (Wang et al., 2018), HumanEval (Chen, 2021), and MMLU (Hendrycks et al., 2020). While these offer controlled and reproducible measurement, they struggle to capture open-ended generation and diverse real-world use (Liang et al., 2022), and are increasingly susceptible to test data contamination (Magar and Schwartz, 2022). Recent evaluation has therefore shifted toward preference-based protocols, including LLM-as-a-judge methods (Zheng et al., 2023) and large-scale human preference platforms. LM Arena is a prominent example: users compare two anonymized model responses to the same prompt, and the resulting pairwise preferences are aggregated into continuously updated leaderboard ratings (Chiang et al., 2024; Boubdir et al., 2024). This evaluation paradigm makes response anonymity central to leaderboard reliability.

De-anonymization and leaderboard vulnerabilities.

The anonymity of pairwise comparisons can fail when model identity is inferable from generated responses, enabling source attribution and downstream ranking manipulation (Min et al., 2025; Huang et al., 2025; Zhao et al., 2025). Such identifiability typically arises from emergent statistical regularities in model outputs, *e.g.*, BoW or TF-IDF signatures, and is closely related to the broader problem of attributing text to its source model (Uchendu et al., 2021; Antoun et al., 2024). Also, this is distinct from approaches that insert identity signals deliberately, such as watermarking and fingerprinting (Kirchenbauer et al., 2023; Xu et al., 2024). However, emergent statistical cues become unreliable when models are stylistically similar, closely ranked, or subject to preference-bias corrections. To our knowledge, we are the first to cast leaderboard de-anonymization as a learning problem, training a target-specific detector rather than relying on hand-crafted surface features.

3 INTERPOL: Interpolated Preference Learning for LLM Identification

Problem setup. Let us denote a set of accessible LLMs as $\mathcal{M} = \{M_1, M_2, \dots, M_K\}$ where each LLM $M \in \mathcal{M}$ can generate response y for the given input query q , *i.e.*, $y \sim M(q)$. Then, our goal de-anonymizing the leaderboard (*i.e.*, identify source LLM) is to construct a detector model f_θ that assigns higher scores to response y_* generated by a target model M_* compared to response y_k from other models $M_k \in \mathcal{M} \setminus \{M_*\}$:

$$f_\theta(q, y_*) > f_\theta(q, y_k), \quad \forall M_k \in \mathcal{M} \setminus \{M_*\}. \quad (1)$$

We assume that we have a set of queries $\mathcal{Q} = \{q^i\}_{i=1}^N$ and a set of responses $\mathcal{Y} = \{y_k^i | y_k^i \sim M_k(q^i), q^i \in \mathcal{Q}, M_k \in \mathcal{M}\}$. Here, $\mathcal{Y}$ can be constructed by querying each model (*e.g.*, public API), without leaderboard access. Then, one can learn the detector model f_θ using $\mathcal{Q}$ and $\mathcal{Y}$; for instance, a simple regression model based on bag-of-words (Huang et al., 2025). However, these surface-level features often fail to capture the subtle stylistic and reasoning patterns that distinguish modern LLMs. Therefore, INTERPOL instead trains the detector by finetuning LLM on $\mathcal{Q}$ and $\mathcal{Y}$.

Threat model. We consider an attacker who is a model developer seeking to raise the ranking of their own model on a public, voting-based leaderboard such as LM Arena. The attacker participates as a normal user without access to system internals.

As shown in prior works (Zhao et al., 2025; Huang et al., 2025), strategic voting alone is sufficient to manipulate leaderboard outcomes. A crucial requirement for such manipulation is the ability to de-anonymize model responses, *i.e.*, to infer which competing LLM produced each answer in a duel so that the attacker can cast selective votes that most benefit their target model. Prior to deployment, the attacker collects model responses by querying publicly accessible models through their APIs, including the target model, and uses the collected responses to train a target-specific detector. At test time, the trained detector including INTERPOL enables the attacker to identify the target model’s responses among previously unseen anonymous responses encountered on the leaderboard, using only the response text and without access to model identity labels, leaderboard meta-data, or privileged platform information.

3.1 Identification via Preference Learning

To train the detection model, we first construct a dataset of preference pairs:

$$\mathcal{D} = \{(q^i, y_*, y_o^i)\}_{i=1}^N,$$

where for each query $q^i \in \mathcal{Q}$, y_*^i represents the response from target model M_* , and y_o^i represents the selected responses from other models $\mathcal{Y}^i = \{y_k^i | y_k^i \sim M_k(q^i), \forall M_k \in \mathcal{M} \setminus \{M_*\}\}$. To maximize discrimination difficulty during training, we select a hard negative y_k^i as y_o^i by computing semantic similarity between the target response, and selecting the most similar one:

$$y_o^i = \arg \max_{y_k^i \in \mathcal{Y}^i} \text{sim}(\mathbf{e}(y_*^i), \mathbf{e}(y_k^i)), \quad (2)$$

where $\mathbf{e}(\cdot)$ represents embeddings from the sentence encoder, and $\text{sim}(\cdot, \cdot)$ is a cosine similarity. Then, the detector model f_θ is trained to prefer target response y_*^i over others’ response y_k^i by minimizing the following binary cross-entropy loss:

$$\mathcal{L}_{\text{pair}} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathcal{L}_{\text{pref}}(q^i, y_*^i, y_o^i), \quad (3)$$

$$\mathcal{L}_{\text{pref}}(q, y_w, y_l) = -\log \sigma(f_\theta(q, y_w) - f_\theta(q, y_l)).$$

This loss function enforces the preference modeling based on Bradley-Terry (BT) model (Bradley and Terry, 1952) by maximizing the probability that the target response receives a higher likelihood than the competing response.

For training, we use only the single hardest negative y_o^i . However, during validation and test, we expand the evaluation to include responses from all non-target LLMs. Namely, given K LLMs, we evaluate against the remaining $K - 1$ LLMs (excluding the target) to comprehensively assess detector performance across the entire model space.

3.2 Hard Negative Synthesis via Distillation and Model Interpolation

The promising way to improve the performance of detector f_θ is enlarging the size of $\mathcal{D}$ (*i.e.*, larger N) or expanding the diversity of y_o^i by incorporating more LLMs (*i.e.*, larger K). However, this approach often incurs the additional costs for new data collection such as increased API callings. This motivates our approach: *Can we generate synthetic hard negatives while maintaining high discrimination difficulty?* To answer this question, we propose a new approach that synthesizes hard negatives by

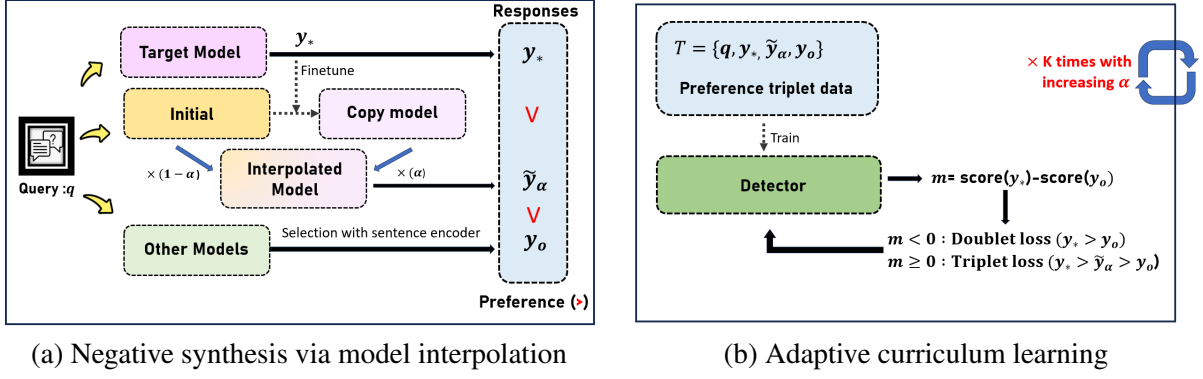


Figure 1: **Overview of INTERPOL.** (a) Target-mimicking copy model is interpolated with its initial backbone to generate synthetic hard negatives with controllable similarity (*i.e.*, difficulty) to the target LLM. These responses form training triplets together with the target response and the most similar non-target response. (b) Detector uses pairwise or triplet preference objectives according to the margin between target and non-target responses, while increasing α across K stages to progressively raise the task difficulty.

leveraging model interpolation (Wortsman et al., 2022; Ilharco et al., 2022).

Specifically, the proposed approach begins by creating *copy model* $\hat{M}_*$ that is trained to mimic the response of target model M_* through supervised fine-tuning (*i.e.*, off-policy knowledge distillation):

$$\hat{M}_* = \arg \min_{\phi} \frac{1}{|\mathcal{D}|} \sum_{(q^i, y_*^i) \in \mathcal{D}} \mathcal{L}_{CE}(\hat{M}_{\phi}(q^i), y_*^i), \quad (4)$$

where $\mathcal{L}_{CE}$ denotes a standard language modeling loss. After training, $\hat{M}_*$ serves as the foundation for generating synthetic responses by varying the levels of similarity to the target model M_* : we synthesize responses of intermediate-difficulty from the model $\tilde{M}_{\alpha}$ which is constructed with an interpolated model parameter $\tilde{\phi}_{\alpha}$:

$$\tilde{\phi}_{\alpha} = (1 - \alpha)\phi_{\text{init}} + \alpha\hat{\phi}_{\text{copy}}, \quad (5)$$

where $\alpha \in [0, 1]$ is hyperparameter to control difficulty, $\phi_{\text{init}}, \hat{\phi}_{\text{copy}}$ denote the parameters of initial copy model and trained model $\hat{M}_*$, respectively.

Then, using $\tilde{M}_{\alpha}$, we construct the response-triplet dataset as below:

$$\mathcal{T}_{\alpha} = \{(q^i, y_*^i, \tilde{y}_{\alpha}^i, y_o^i)\}_{i=1}^N, \quad (6)$$

where $\tilde{y}_{\alpha}^i = \tilde{M}_{\alpha}(q^i)$ is the response from interpolated model. Because weight-space interpolation induces smooth functional interpolation, the generations of $\tilde{M}_{\alpha}$ vary between the initializer (*easy-to-discriminate*), as $\alpha \rightarrow 0$ they resemble the initializer (*easy-to-discriminate*), as $\alpha \rightarrow 1$ they approach the target-like copy (*hard-to-discriminate*), and for $\alpha \in (0, 1)$ they yield middle-difficulty negatives, typically satisfying

$$0 < \Delta(q, \tilde{y}_{\alpha}) < \Delta(q, y_o),$$

where $\Delta(q, y) = f_{\theta}(q, y_*) - f_{\theta}(q, y)$.

3.3 Adaptive Curriculum Learning

With the constructed response-triplet dataset $\mathcal{T}_{\alpha}$, we train our detector model f_{θ} by minimizing the following linear combination of binary cross-entropy losses:

$$\begin{aligned} \mathcal{L}_{\text{fin}} &= \frac{1}{|\mathcal{T}_{\alpha}|} \sum_{i \in \mathcal{T}_{\alpha}} \mathcal{L}_{\text{trp}}^i, \\ \mathcal{L}_{\text{trp}}^i &= \lambda_1 \mathcal{L}_{\text{pref}}(q^i, y_*^i, \tilde{y}_{\alpha}^i) \\ &\quad + \lambda_2 \mathcal{L}_{\text{pref}}(q^i, \tilde{y}_{\alpha}^i, y_o^i) \\ &\quad + \lambda_3 \mathcal{L}_{\text{pref}}(q^i, y_*^i, y_o^i). \end{aligned} \quad (7)$$

where $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters that weigh the relative importance of each triplet component. While this triplet training is effective, some queries might suffer during training if they even fail to minimize a simple pair-wise loss (Eq. 3). To improve training by mitigating this issue, we consider adaptive learning. Specifically, we first introduce discrimination margin:

$$m^i = f_{\theta}(q^i, y_*^i) - f_{\theta}(q^i, y_o^i), \quad (8)$$

where $m^i < 0$ indicates that the detector f_{θ} currently fails to correctly discriminate target response y_*^i and the hard negative response y_o^i . For those samples with $m^i < 0$, we then apply $\mathcal{L}_{\text{pref}}$ instead of $\mathcal{L}_{\text{trp}}$ and focus on solving easier problem.

In addition, to further improve the performance of detection model f_{θ} , we introduce curriculum learning that progressively increase discrimination difficulty by varying the degree of interpolation α . Specifically, we initially construct easier triplet

data $\mathcal{T}_{\alpha_1}$ with small value of α_1 and training the detection model following Eq. 7. After this stage, we construct a harder dataset $\mathcal{T}_{\alpha_2}$ with a larger value $\alpha_2 > \alpha_1$ and continue training. In this way, we perform iterative data construction and training across progressively harder stages, where we set $K = 2$ (with $\alpha_1 = 0.5, \alpha_2 = 0.75$) in our experiments.

4 Experiment

Datasets and candidate pools. We evaluate INTERPOL on two query sources: the *Alpaca dataset* (Taori et al., 2023) and the *Arena human preference dataset* (Tang et al., 2025). We collect responses from 12 LLMs spanning six model families and split the data into training, validation, and test sets. We consider three primary target LLMs: *GPT-4o* (OpenAI, 2024), *Gemini-Pro* (Comanici et al., 2025), and *Claude-Sonnet-4* (Anthropic, 2025). In addition to the standard 12-model pool, we use a close-Elo competitor pool and a composed pool for generalization analysis. More details such as full dataset statistics and generation details are provided in Appendix A.2.

Baselines. We consider six representative baselines. (1) *Length-Word* and (2) *Length-Char* use response length measured by word count and character count, respectively. (3) *TF-IDF* (Salton and Buckley, 1988) and (4) *Bag-of-Words* (Salton et al., 1975) use lexical n -gram features with logistic-regression classifiers. (5) *Neural-based multi-class classifier* (Tay et al., 2020) trains a supervised source-attribution model to predict the generating LLM. (6) *LLM-as-a-Judge* (Zheng et al., 2023) with few-shot in-context learning asks a judge model to identify which candidate response is more likely to come from the target LLM. Implementation and evaluation details are in Appendix A.3.

Implementation. We use GRM-Llama-3.2-3B as the detector backbone and Phi-4-mini-instruct as the target-mimicking copy-model backbone. The detector is trained with a two-stage curriculum using $\alpha = 0.5$ and 0.75 . Full implementation details are provided in Appendix A.4.

4.1 Identification Results

Table 1 reports target-present identification results on both the Arena Human Preference and Alpaca datasets. Each comparison contains one response from the target model and one response from a non-target model to the same query, and the detector

Table 1: **Main target-present identification results.** We report average Accuracy (Acc.) and AUROC across 11 non-target LLMs on the Arena Human Preference and Alpaca datasets. Best scores are highlighted in **bold**. Method abbreviations: Len-W = Length-Word, Len-C = Length-Char, TF-IDF = TF-IDF, BoW = Bag-of-Words, Neural = Neural-based classification, and Judge = LLM-as-a-Judge.

Target	Method	Arena		Alpaca	
		Acc.	AUROC	Acc.	AUROC
GPT-4o	Len-W	0.556	0.611	0.493	0.551
	Len-C	0.564	0.610	0.495	0.552
	TF-IDF	0.697	0.766	0.655	0.704
	BoW	0.709	0.772	0.631	0.692
	Neural	0.898	0.888	0.726	0.767
	Judge	0.822	—	0.679	—
	INTERPOL	0.908	0.920	0.887	0.875
Gemini-Pro	Len-W	0.792	0.830	0.618	0.647
	Len-C	0.768	0.815	0.618	0.639
	TF-IDF	0.844	0.926	0.780	0.851
	BoW	0.887	0.948	0.798	0.911
	Neural	0.908	0.940	0.828	0.894
	Judge	0.958	—	0.620	—
	INTERPOL	1.000	0.979	0.946	0.967
Sonnet-4	Len-W	0.639	0.621	0.525	0.529
	Len-C	0.380	0.397	0.519	0.523
	TF-IDF	0.810	0.902	0.763	0.834
	BoW	0.851	0.929	0.767	0.861
	Neural	0.883	0.908	0.825	0.899
	Judge	0.902	—	0.631	—
	INTERPOL	0.946	0.976	0.982	0.954

selects the response assigned the higher target score. Across both query sources, INTERPOL achieves the strongest overall performance for all three target LLMs. On Arena, INTERPOL improves over the baseline average by 23.5% in Accuracy and 21.2% in AUROC. On Alpaca, the corresponding gains are 41.1% in Accuracy and 28.8% in AUROC. These improvements show that the advantage of INTERPOL is not confined to a single query distribution, but holds across both user-generated Arena-style queries and standard instruction-following queries.

The full model-wise Accuracy and AUROC breakdowns in Appendix Tables 8 and 9 further show that this trend also holds in within-family comparisons, including GPT-4o vs. GPT-4o-mini, Gemini-Pro vs. Gemini-Flash, and Sonnet-4 vs. Sonnet-3.5. Such comparisons are more challenging because models within the same family may share similar training pipelines, response styles, or distillation relationships. While feature-based baselines become less reliable in these cases, INTERPOL remains more stable, indicating that interpolated preference learning captures target-specific response signatures beyond broad family-level or surface statistical cues.

Close-Elo competitor evaluation. The standard 12-model pool provides a broad evaluation of target-present identification, but it can contain models that differ substantially in family, scale, and capability. To further reduce coarse separability among candidates, we evaluate INTERPOL on a close-Elo competitor pool consisting of high-ranked models clustered within a narrow range on the LM Arena text leaderboard. The pool includes our three target models, GPT-4o, Gemini-2.5-pro, and Claude-sonnet-4, together with five close competitors: Grok-4-Fast, DeepSeek-v3.1, GLM-4.6, Kimi-K2-0905, and Qwen3-Max-0923. The Elo ratings and leaderboard snapshot date used to construct this pool are described in Appendix B.3.

The results in Tables 10–12 show that INTERPOL remains effective under this close-Elo evaluation. INTERPOL achieves the strongest performance for GPT-4o and Sonnet-4, and remains competitive for Gemini-Pro, where the Neural-Based baseline is slightly stronger. Across all targets, INTERPOL consistently outperforms feature-based baselines, indicating that its gains are not driven solely by coarse separability in the standard pool but also persist among direct high-ranked competitors with similar leaderboard ratings.

4.2 Additional Analyses

We further analyze what enables INTERPOL to identify target-model responses and how its training procedure improves discrimination. We examine response-level attribution signals, component ablation, interpolation difficulty, score distributions, and interpretability. Further results and expanded evaluations are provided in Appendix C.

Response-level attribution signals. We first ask whether the attribution signal is present in the response itself, or whether the detector mainly relies on query-conditioned answer quality. If the detector only exploits how well a response matches a particular query, then removing the query should substantially weaken identification. To test this, we evaluate a response-only variant that removes the query and scores each response independently.

As shown in Table 2, the performance is mostly preserved which indicate that substantial attribution signal is already present in the responses themselves. For Gemini-Pro and Sonnet-4, the response-only variant performs comparably to, or even better than, the query-conditioned detector. Specifically, response-only evaluation achieves 0.989 Accuracy

Table 2: **Response-only ablation.** We compare the standard query-conditioned detector with a response-only variant that removes the query input.

Target	Query-conditioned		Response-only	
	Accuracy	AUROC	Accuracy	AUROC
GPT-4o	0.908	0.920	0.879	0.851
Gemini-Pro	1.000	0.979	0.989	0.991
Sonnet-4	0.946	0.976	0.981	0.983

Table 3: **Component ablation.** Comparison of different configurations of INTERPOL to train detector model. We vary the use of negative sampling strategy (Eq. 2), triplet loss (Eq. 7), adaptive curriculum (Eq. 8), and iterative training ($K = 1$ or 2). For compactness, we denote these components as **N** (Negative), **T** (Triplet), **A** (Adaptive), and **I** (Iterative). We report average Accuracy and AUROC across three target LLMs. The best scores are highlighted in **bold**.

	N	T	I	A	Accuracy	AUROC
Hard	–	–	–	–	0.863	0.882
Hard	✓	–	–	–	0.914	0.931
Hard	✓	–	–	✓	0.935	0.951
Easy	✓	–	–	✓	0.887	0.896
Random	✓	–	–	✓	0.928	0.920
Hard (Ours)	✓	✓	✓	✓	0.951	0.958

/ 0.991 AUROC for Gemini-Pro and 0.981 Accuracy / 0.983 AUROC for Sonnet-4. For GPT-4o, response-only performance decreases moderately, from 0.908 to 0.879 Accuracy and from 0.920 to 0.851 AUROC, but remains clearly effective. These results show that INTERPOL is not merely exploiting query-conditioned answer quality. Rather, anonymous responses themselves contain model-specific behavioral signatures, directly supporting the vulnerability of response anonymity in Arena-style evaluations.

Component ablation. Table 3 reports the effect of each component of INTERPOL on detector performance, averaged across three target LLMs (GPT-4o, Sonnet-4, and Gemini-Pro). First, introducing triplet-based training improves Accuracy from 0.863 to 0.914 and AUROC from 0.882 to 0.931 under hard negative sampling, demonstrating the benefit of explicitly modeling relative preference orderings. Second, adding adaptive curriculum learning further improves performance to 0.935 Accuracy and 0.951 AUROC. Third, iterative curriculum training provides an additional improvement, with the full configuration achieving the best overall performance of 0.951 Accuracy and 0.958 AUROC. Finally, the choice of negative sampling strategy also plays an important role. Under the same triplet and adaptive training configuration,

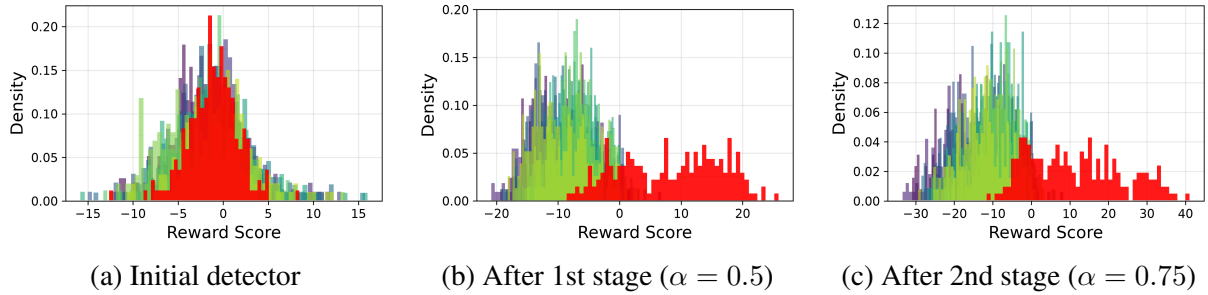


Figure 2: **Score distributions of the detector.** The detector is trained to identify Gemini-Pro. **Red** indicates the scores of the target model, and other colors indicate the score of non-target models. (a) Initial detector before any fine-tuning. (b) After the first triplet training stage with interpolation factor $\alpha = 0.5$, where the detector begins to assign higher scores to the target model; the score range spans approximately from -20 to 20 . (c) After iterative curriculum training with harder negatives ($\alpha = 0.75$), resulting in clearer separation between target and non-target models; the score range further expands from about -30 to 40 .

Table 4: **Validation of the interpolation trajectory on GPT-4o.** Target-response perplexity (PPL) decreases as α increases, while generation-level embedding similarity (Emb. Sim.) increases.

Model	PPL ↓	Emb. Sim. ↑
Base (Phi-4)	2.114	–
$\alpha = 0.25$	1.940	0.795
$\alpha = 0.50$	1.813	0.802
$\alpha = 0.75$	1.724	0.815
Copy	1.668	–

hard negative sampling achieves 0.935 Accuracy and 0.951 AUROC, outperforming both random sampling (0.928/0.920) and easy negative sampling (0.887/0.896). Overall, these results show that triplet preference learning, adaptive curriculum learning, iterative training, and semantically challenging negatives each contribute to the final performance of INTERPOL.

Validation of interpolation-based difficulty. Our curriculum assumes that increasing the interpolation factor α produces progressively more target-like responses and therefore harder negatives. To validate this ordering at the output level, we examine target-response perplexity (PPL) and generation-level embedding similarity along the interpolation trajectory.

As shown in Table 4, for GPT-4o, target PPL decreases monotonically as α increases toward the copy-model endpoint, while embedding similarity to the target responses consistently increases. The same monotonic trends are observed for Gemini-Pro and Sonnet-4; full results are in Appendix C.5.

These results provide output-level evidence that increasing α moves interpolated generations toward the target-mimicking endpoint. Consequently, responses at $\alpha = 0.5$ serve as relatively easier negatives, whereas those at $\alpha = 0.75$ are more target-

Table 5: **Interpretability analysis via erasure-based attribution.** We report the average reduction in discrimination margin after cumulatively removing the ten highest-ranked words on 20 high-margin GPT-4o examples. Larger drops indicate that the removed words contribute more strongly to the detector’s decision.

Deletion ranking	Margin drop
INTERPOL-ranked	11.567
TF-IDF-ranked	6.452
BoW-ranked	6.394
Random	6.195
Least-important	-0.690

like and therefore harder to distinguish, supporting the curriculum ordering used in Section 3.3.

Score distributions. To gain deeper insight into how these performance improvements occur, we visualize the evolution of detector score distributions in Figure 2. At the initial stage (a), the detector does not exhibit a clear preference toward the target model, resulting in substantial overlap between target and non-target responses. In (b), after the first triplet training step with interpolation factor $\alpha = 0.5$, the distribution of the target model begins to separate from the others, indicating that the detector has started to assign systematically higher scores to target responses. Finally, in (c), iterative curriculum training with harder negatives ($\alpha = 0.75$) produces a clearer margin between target and non-target distributions. This progression demonstrates that INTERPOL gradually learns a target-specific scoring function.

Interpretability analysis. To examine which response cues drive INTERPOL’s decisions beyond those captured by shallow statistical baselines, we conduct an erasure-based local attribution analysis following Nguyen (2018). We focus on GPT-4o test

pairs for which INTERPOL correctly identifies the target response while TF-IDF incorrectly favors the non-target response, and select 20 high-margin cases for detailed analysis.

Following the discrimination margin defined in Eq. 8, we use $m_i = f_\theta(q^i, y_*^i) - f_\theta(q^i, y_o^i)$, where y_*^i is the GPT-4o response and y_o^i denotes the paired non-target response at test time. A positive m_i therefore indicates that the detector assigns a higher target score to the actual GPT-4o response. For each word, we measure its importance by the decrease in m_i caused by deleting that word from the corresponding response. We then cumulatively remove the ten highest-ranked words according to each attribution method and report the average resulting margin reduction.

As shown in Table 5, removing words ranked most important by INTERPOL reduces the discrimination margin by 11.567 on average, substantially more than TF-IDF-ranked (6.452), BoW-ranked (6.394), or randomly selected words (6.195). In contrast, removing the least-important words does not weaken the target decision. These results indicate that INTERPOL’s predictions rely on decision-relevant response cues that are not fully characterized by shallow lexical statistics. A qualitative case study further examining these attribution cues is provided in Appendix C.7.

5 Simulation of Ranking Manipulation

To quantify the practical impact of model identification, we perform controlled simulations of ranking manipulation using real-world chatbot arena data. Our goal is not to estimate the exact cost of manipulating a live platform, but to provide a controlled and ethically appropriate evaluation of how model identification affects the efficiency of strategic voting under fixed leaderboard dynamics. Specifically, we adopt the setup by Huang et al. (2025), which models adversarial voting efficiency under the BT model and Elo ranking dynamics. We use the `lmarena-ai/arena-human-preference-140k` dataset released in July 2025,¹ containing approximately 140k pairwise battles. Rankings are reconstructed with BT model, and Elo scores are recalculated every 100 new votes. At each simulation step, the system draws two models uniformly at random from the candidate pool to form a synthetic battle. The attacker then receives

two anonymized responses and chooses whether to cast a vote or abstain.

Attacker objectives and metrics. In the simulation, the attacker’s objective is to shift a target model M from its initial rank to a specified target rank (e.g., Rank 1). Then, the effectiveness of the attack is evaluated using two key metrics:

- **Total interactions:** The total number of queries submitted to the leaderboard. This metric represents the overall resource cost, as the attacker must spend time generating responses even for battles where they eventually abstain.
- **Adversarial votes:** The number of votes actually cast by the attacker. This measures the direct influence exerted on the scoring system. It is primarily determined by the BT ranking dynamics and remains relatively constant across different identification methods.

Simulation policy. For a target model M , we predefine an enemy list consisting of the top-ranked models in the initial leaderboard snapshot, specifically the models ranked 1–3. This enemy list is kept fixed throughout the simulation, even as the target model’s rank changes. Then, the attacker takes a following attack policy:

- *When the target model is identified*, the attacker casts a vote that promotes the target.
- *When the target is not identified but the model on the enemy list is identified*, the attacker casts a vote against the identified enemy model.
- *Otherwise*, the attacker abstains.

Operationally, this strategy requires detectors only for the target model and the three fixed enemy models, avoiding the need to update the set of detectors as the leaderboard ranking changes.

5.1 Simulation Results

Under the above policy, detector accuracy matters because a sampled battle is exploitable only when a present target or enemy-list model is correctly identified; identification errors make the attacker either spend interactions on misdirected votes or abstain from otherwise useful battles. We therefore compare baseline detectors and INTERPOL as alternative identification modules for promoting the same target models. The targets cover distinct model families and ranking regimes, with `o3-2025-04-16`, `deepseek-v3-0324`, and `llama-4-scout-17b-16e-instruct` initialized at ranks 5, 15, and 45, re-

¹<https://news.lmarena.ai/opendata-july2025/>

Table 6: **Required interactions (votes) to promote target models.** We evaluate ranking manipulation across three target models with different initial ranks. Each entry is reported as interactions (votes), where interactions indicate the total number of interactions and votes indicate the number of adversarial votes required to reach the target rank under different detectors. Best interaction costs are highlighted in **bold**.

Method	o3-2025-04-16 current rank: #5		deepseek-v3-0324 current rank: #15		llama-4-scout-17b-16e-instruct current rank: #45	
	Rank 1	Rank 2	Rank 1	Rank 2	Rank 1	Rank 2
Len-C	26400 (2188)	25100 (2073)	76900 (6425)	52400 (4389)	109100 (9142)	67500 (5670)
Len-W	22200 (2137)	6400 (588)	78400 (7645)	44800 (4371)	99300 (9706)	58900 (5741)
TF-IDF	21800 (2468)	2500 (259)	72200 (8131)	38300 (4307)	85300 (9686)	51300 (5789)
BoW	22200 (2631)	2300 (250)	69500 (8188)	35800 (4254)	85200 (10082)	49600 (5841)
Neural	22400 (2911)	2300 (275)	62000 (8011)	32700 (4262)	70600 (9124)	45900 (5927)
Judge	22500 (2916)	2300 (275)	62600 (8073)	32800 (4269)	70600 (9100)	45900 (5916)
INTERPOL	22200 (3066)	2000 (243)	59600 (8137)	30900 (4257)	64700 (8871)	43300 (5912)

Table 7: **Effect of competitor-aware voting with INTERPOL.** We compare the target-only strategy and the enemy-list strategy for promoting the Rank 5 target model, o3-2025-04-16. Each entry reports total interactions with adversarial votes in parentheses. Interaction reduction is computed from total interactions.

Target model	Current rank	Target rank	Target-only strategy	Enemy-list strategy	Interaction reduction
o3-2025-04-16	5	Rank 1	45,800 (1,618)	22,200 (3,066)	51.5%
		Rank 2	16,800 (564)	2,000 (243)	88.1%
		Rank 3	13,000 (442)	1,300 (158)	90.0%
		Rank 4	1,000 (37)	500 (53)	50.0%

spectively. For each target, we measure the total interactions and adversarial votes required to reach Rank 1 or Rank 2. Table 6 reports the required interactions and adversarial votes for each identification method under the same enemy-list voting policy. Across most promotion objectives, INTERPOL achieves the lowest or competitive interaction cost, especially for lower-ranked targets where useful promotion opportunities are sparse. Relative to the Length-Char detector, INTERPOL reduces interactions by 22.5% for promoting deepseek-v3-0324 to Rank 1 and by 40.7% for promoting llama-4-scout-17b-16e-instruct to Rank 1. This demonstrates that stronger response identification directly lowers manipulation cost: weaker surface-statistic detectors miss more actionable battles and therefore require more Arena interactions, whereas INTERPOL converts sampled interactions into effective adversarial votes more efficiently.

5.2 Effect of Competitor-Aware Voting

To assess whether identification enables manipulation beyond direct target promotion, we compare the enemy-list strategy with a target-only baseline. The target-only baseline casts a vote only when the target model is identified and abstains otherwise, while the enemy-list strategy additionally downvotes identified higher-ranked competitors as defined above. Table 7 reports the comparison for promoting the Rank 5 target model,

o3-2025-04-16. For the most difficult objective, Rank 5 $\rightarrow$ 1, the interaction cost decreases from 45,800 to 22,200, corresponding to a 51.5% reduction. Across the remaining objectives, the reduction ranges from 50.0% to 90.0%. These results show that competitor-aware voting substantially lowers the operational cost of rank manipulation by exploiting not only direct target appearances but also battles involving higher-ranked competitors.

6 Conclusion

In this paper, we presented INTERPOL, a new model-driven approach to identify the source of LLM responses. INTERPOL leverages triplet-based training with adaptive and iterative curriculum learning, supported by synthetic negatives generated via model interpolation. Extensive experimental results demonstrate that INTERPOL consistently surpasses statistical baselines. Our findings highlight a critical vulnerability in current anonymous voting-based leaderboards, showing that despite pragmatic defenses, substantial loopholes remain exploitable. By explicitly uncovering these weaknesses, our work emphasizes the necessity of moving beyond statistical safeguards and provides a foundation for designing robust and trustworthy evaluation infrastructures for LLMs.

Limitations

INTERPOL exposes residual response identifiability in anonymous LLM evaluation, but its current scope is centered on text-only, single-turn English interactions. This matches the main setting of Arena-style text leaderboards studied in this work, yet broader multilingual, multimodal, or multi-turn settings may introduce different attribution cues. Since INTERPOL only requires query–response pairs and target-specific training signals, extending the framework to these settings is a natural direction once suitable response corpora and detector backbones are available.

In addition, our manipulation analysis is based on reconstructed BT/Elo dynamics and controlled voting policies, rather than direct access to a deployed leaderboard system. The reported interaction counts should therefore be interpreted as relative simulation-based evidence of risk, not exact deployment-level attack costs. Incorporating platform-specific traffic patterns, aggregation rules, and defense mechanisms would further refine this estimate.

Broader Impact and Ethical Implications

This work studies a vulnerability in anonymous, voting-based LLM evaluation. The intended impact of INTERPOL is defensive: it provides a way to test whether model responses remain anonymous under stronger identification methods than surface-level statistical baselines. Such analysis can help leaderboard operators and evaluation researchers audit anonymity assumptions, detect manipulation risks, and design more robust aggregation and defense mechanisms.

At the same time, there is a risk that the capability studied in this paper can be misused. A malicious user could apply response identification to infer which model produced an anonymous response and then cast selective votes to promote or suppress specific systems. Our ranking-manipulation simulations demonstrate this risk under controlled conditions: stronger identification can reduce the number of interactions required for strategic voting. Therefore, INTERPOL should not be used as a tool for manipulating public leaderboards.

All experiments in this paper are conducted offline using publicly available datasets, offline-collected model responses, reconstructed leaderboard dynamics, and simulated voting policies. We do not access private platform internals, user iden-

tities, or non-public metadata. In practical deployments, response-identification systems should be used only as audit signals for evaluating platform robustness, not as standalone evidence for punitive action. We encourage future work to use these findings to develop stronger anonymization, vote-auditing, and manipulation-resistant evaluation protocols.

Acknowledgments

Jaehyung Kim is affiliated with the Department of Artificial Intelligence at Yonsei University. This research was supported in part by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University); No. RS-2025-02215344, Development of AI Technology with Robust and Flexible Resilience Against Risk Factors).

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Anthropic. 2024. [Introducing Claude 3.5 Sonnet](#).
- Anthropic. 2025. [System Card: Claude Opus 4 & Claude Sonnet 4](#). Technical report, Anthropic.
- Wissam Antoun, Benoît Sagot, and Djamé Seddah. 2024. From text to source: Results in detecting large language model-generated content. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7531–7543.
- Meriem Boubdir, Edward Kim, Beyza Ermiş, Sara Hooker, and Marzieh Fadaee. 2024. Elo uncovered: Robustness and best practices in language model evaluation. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.
- Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, and 1 others. 2024. Chatbot arena: An

- open platform for evaluating llms by human preference. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- Gheorghe Comanici and 1 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multi-modality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpaca-eval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Yangsibo Huang, Milad Nasr, Anastasios Angelopoulos, Nicholas Carlini, Wei-Lin Chiang, Christopher A Choquette-Choo, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Ken Ziyu Liu, and 1 others. 2025. Exploring and mitigating adversarial manipulation of voting-based leaderboards. *arXiv preprint arXiv:2501.07493*.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hananeh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.
- John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.
- LM Arena Team. 2024a. [Sentiment control in lm arena: Towards fairer leaderboards](#). LM Arena Blog. Accessed: 2025-09-18.
- LM Arena Team. 2024b. [Style control in lm arena: Mitigating length and presentation bias](#). LM Arena Blog. Accessed: 2025-09-18.
- Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. *arXiv preprint arXiv:2203.08242*.
- Rui Min, Tianyu Pang, Chao Du, Qian Liu, Minhao Cheng, and Min Lin. 2025. Improving your model ranking on chatbot arena by vote rigging. *arXiv preprint arXiv:2501.17858*.
- Dong Nguyen. 2018. [Comparing automatic and human evaluation of local explanations for text classification](#). In *North American Chapter of the Association for Computational Linguistics*.
- OpenAI. 2024. [GPT-4o System Card](#). Technical report, OpenAI.
- Arjun Panickssery, Samuel Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Qwen Team. 2024a. Qwen2.5-3b-instruct. <https://huggingface.co/Qwen/Qwen2.5-3B-Instruct>.
- Qwen Team. 2024b. Qwen3-8b. <https://openlaboratory.ai/models/qwen3-8b>.
- Gerard Salton and Chris Buckley. 1988. [Term-weighting approaches in automatic text retrieval](#). *Inf. Process. Manag.*, 24:513–523.
- Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. [A vector space model for automatic indexing](#). *Commun. ACM*, 18:613–620.
- Shivalika Singh, Yiyang Nan, Alex Wang, Daniel D’Souza, Sayash Kapoor, Ahmet Üstün, Sanmi Koyejo, Yuntian Deng, Shayne Longpre, Noah A Smith, and 1 others. 2025. The leaderboard illusion. *arXiv preprint arXiv:2504.20879*.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Kelly Tang, Wei-Lin Chiang, and Anastasios N. Angelopoulos. 2025. [Arena Explorer: A topic modeling pipeline for LLM evals & analytics](#). Arena.ai blog. Blog post describing the Arena Explorer topic modeling pipeline for analyzing LLM evaluation data.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. [Stanford Alpaca: An instruction-following LLaMA model](#). GitHub repository.
- Yi Tay, Dara Bahri, Che Zheng, Clifford Brunk, Donald Metzler, and Andrew Tomkins. 2020. [Reverse engineering configurations of neural text generation models](#). *Preprint*, arXiv:2004.06201.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak

Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

LMarena Team. 2024. Arena human preference dataset. <https://huggingface.co/datasets/lmarena-ai/arena-human-preference-140k>. Large-scale dataset with over 140k human-labeled pairwise preferences.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. In *Findings of the association for computational linguistics: EMNLP 2021*, pages 2001–2016.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, pages 23965–23998. PMLR.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. 2024. Instructional fingerprinting of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3277–3306.

Wenting Zhao, Alexander M Rush, and Tanya Goyal. 2025. Challenges in trustworthy human evaluation of chatbots. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3359–3365.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhonghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. *Judging LLM-as-a-judge with mt-bench and chatbot arena*. Preprint, arXiv:2306.05685.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*.

A Experimental Details

A.1 Model Specification

The models used in our dataset are as follows:

Open-source models.

- Meta / Llama-3.2-3B-Instruct²
- Meta / Llama-3.1-8B-Instruct³
- Alibaba / Qwen2.5-3B-Instruct⁴
- Alibaba / Qwen3-8B⁵
- Google / Gemma-2-2B-it⁶
- Google / Gemma-2-9B-it⁷
- Microsoft / Phi-4-mini-instruct⁸

API models.

- Anthropic / Claude-4 Sonnet :
claude-sonnet-4-20250514
- Anthropic / Claude-3.5 Sonnet :
claude-3-5-sonnet-20241022
- Google / Gemini-Flash : gemini-2.5-flash
- Google / Gemini-Pro : gemini-2.5-pro
- OpenAI / GPT-4o : gpt-4o-20240806
- OpenAI / GPT-4o-mini :
gpt-4o-mini-20240718

Embedding model.

- [sentence-transformers/all-MiniLM-L6-v2](#)

A.2 Datasets and Metrics

For the experiments, we consider two different sources for input queries: (1) *Alpaca dataset* (Taori et al., 2023), an instruction-following corpus of approximately 52k examples generated by LLM widely used for fine-tuning purposes and (2) *Arena human preference dataset* (Team, 2024), a large-scale collection of over 140k pairwise human-labeled preferences obtained from real-world interactions in an LM arena setting. For the Alpaca dataset, we initiated a clustering process to read out 1,500 diverse samples

²<https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct>

³<https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>

⁴<https://huggingface.co/Qwen/Qwen2.5-3B-Instruct>

⁵<https://huggingface.co/Qwen/Qwen3-8B>

⁶<https://huggingface.co/google/gemma-2-2b-it>

⁷<https://huggingface.co/google/gemma-2-9b-it>

⁸<https://huggingface.co/microsoft/Phi-4-mini-instruct>

and selected the final 1,400 for our query pool. For the Arena dataset, we filtered for single-turn, English-only conversations devoid of image inputs, randomly sampling 1,500 dialogues and retaining 1,400 after a cleaning process. We then gather responses to these queries from a diverse set of 12 LLMs, which can be grouped into six families, each containing two variants: Llama-3.2-3B/Llama-3.1-8B (Grattafiori et al., 2024), Qwen2.5-3B/Qwen3-8B (Qwen Team, 2024a,b), Gemma-2-2B/Gemma-2-9B (Team et al., 2024), Claude-Sonnet-4/Claude-Sonnet-3.5 (Anthropic, 2025, 2024), Gemini-Flash/Gemini-Pro (Comanici et al., 2025), and GPT-4o-mini/GPT-4o (OpenAI, 2024).

The resulting dataset, comprising query-response pairs, is pre-split into training (1,000), validation (200), and test (200) sets using a fixed random seed (seed=42) to ensure reproducibility and prevent data leakage. For training, we employ a sentence encoder to select the most semantically similar response from other models, which serves as the negative sample in triplet construction. Meanwhile, for validation and testing, we evaluate each query against all non-target models, resulting in $200 \text{ queries} \times 11 \text{ models} = 2,200$ scored pairs per split. Our main experimental setup focuses on three primary target LLMs: GPT-4o, Gemini-Pro, and Sonnet-4. For evaluation, we report both Accuracy and AUROC on the test sets by using validation sets for model selection. These metrics are computed and averaged across the 11 non-target LLMs to assess the generalization performance of our detector.

A.3 Baseline Details

Statistical baselines. We evaluate four feature-based baselines trained with ℓ_2 -regularized logistic regression (SAGA solver, $C=1.0$), selecting the best seed by validation AUROC over five seeds (100, 200, 300, 400, 500).

- **Length–Word:** number of whitespace-separated tokens in a response; features are standardized.
- **Length–Char:** number of characters in a response; features are standardized.
- **BoW:** bag-of-words counts with n-grams up to 2 and $\text{min_df}=2$.
- **TF–IDF:** TF–IDF features with the same n-gram setting (1–2) and $\text{min_df}=2$.

For each baseline we train until convergence with a sufficiently large iteration cap (up to 2,000).

Neural-based multi-class classification baseline.

We implemented a neural network-based baseline in which a pre-trained transformer model, DeBERTa-v3-base, is fine-tuned to classify the source of each generated response among all candidate LLMs following the source-attribution formulation of Tay et al. (2020). Given a pool of N language models, the classifier is trained to predict which model produced a given answer. The training dataset comprises query-response pairs where each query is answered by all N models, yielding N labeled samples per query. We employ standard supervised learning with cross-entropy loss and select the best checkpoint based on validation accuracy. For evaluation, we apply the model to pairs of candidate responses (e.g., a response from the target model and another from a competitor model) and compare the predicted logits. The response receiving a higher logit for the target class is considered to be from the target model. To reduce position bias, each test pair is evaluated in both forward and reversed order, and only consistent predictions are counted as correct.

LLM-as-a-Judge baseline with few-shot in-context learning.

As another strong baseline, we utilize an open-source LLM as an automated judge. This baseline leverages the in-context learning ability of LLM by providing several few-shot exemplars sampled from the training data. Each prompt consists of an instruction and two candidate responses (the target and a distractor), along with the ground-truth label indicating which response was produced by the target model. For each test case, we query the judge model twice—once in each order (target vs. other and other vs. target)—while keeping the few-shot context fixed for all queries. The judge is instructed to identify the response most likely to have been generated by the target model. Only cases where the judge correctly identifies the target regardless of position are scored as correct; ambiguous or contradictory cases receive partial credit. We implement this baseline using Gemma-2-27B-it (Team et al., 2024). To ensure computational efficiency, we utilize vLLM’s prefix caching feature, as the lengthy few-shot prompt remains identical across all inference requests.

A.4 Implementation Details

To construct the query–response corpus required for detector training, we generate responses for 1,400 queries from a diverse pool of 12 LLMs

Table 8: **Model-wise Accuracy on Arena human preference dataset.** Abbreviations: C3.5 = Sonnet-3.5, C4 = Sonnet-4, G-F = Gemini-Flash, G-P = Gemini-Pro, G2-2B = Gemma-2-2B-it, G2-9B = Gemma-2-9B-it, GPT-M = GPT-4o-mini, GPT-4 = GPT-4o, L3.1-8B = Llama-3.1-8B-Instruct, L3.2-3B = Llama-3.2-3B-Instruct, Q2.5-3B = Qwen2.5-3B-Instruct, Q3-8B = Qwen3-8B. Cells highlighted in **red** indicate *family models* of the target high model (e.g., GPT-4o vs GPT-4o-mini). The best scores are highlighted in **bold**.

Target	Method	C3.5	C4	G-F	G-P	G2-2B	G2-9B	GPT-M	GPT-4	L3.1-8B	L3.2-3B	Q2.5-3B	Q3-8B
GPT-4o	Length-Word	0.385	0.415	0.665	0.708	0.613	0.488	0.515	–	0.588	0.565	0.543	0.635
	Length-Char	0.395	0.428	0.675	0.713	0.620	0.508	0.513	–	0.588	0.575	0.555	0.633
	TF-IDF	0.788	0.778	0.700	0.725	0.690	0.695	0.645	–	0.665	0.663	0.585	0.733
	BoW	0.750	0.775	0.708	0.733	0.715	0.693	0.690	–	0.698	0.690	0.605	0.748
	Neural-Based	0.865	0.965	0.970	0.980	0.985	0.985	0.590	–	0.960	0.965	0.660	0.955
	LLM-judge	0.710	0.730	0.750	0.825	0.830	0.750	0.870	–	0.890	0.915	0.915	0.855
	INTERPOL	0.875	0.970	0.965	0.970	0.975	0.990	0.740	–	0.900	0.920	0.740	0.945
Gemini-Pro	Length-Word	0.835	0.793	0.585	–	0.830	0.858	0.848	0.813	0.785	0.805	0.825	0.733
	Length-Char	0.805	0.770	0.570	–	0.795	0.833	0.825	0.788	0.768	0.778	0.795	0.723
	TF-IDF	0.905	0.883	0.728	–	0.833	0.870	0.860	0.845	0.838	0.830	0.853	0.840
	BoW	0.890	0.885	0.818	–	0.903	0.913	0.895	0.898	0.863	0.900	0.905	0.885
	Neural-Based	0.955	0.955	0.750	–	0.905	0.900	0.915	0.935	0.920	0.940	0.915	0.895
	LLM-judge	0.960	0.955	0.930	–	0.960	0.940	0.955	0.955	0.970	0.970	0.975	0.970
	INTERPOL	1.000	1.000	1.000	–	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Sonnet 4	Length-Word	0.465	–	0.753	0.803	0.683	0.553	0.560	0.593	0.655	0.643	0.600	0.715
	Length-Char	0.538	–	0.255	0.205	0.350	0.468	0.475	0.428	0.370	0.378	0.408	0.305
	TF-IDF	0.583	–	0.818	0.845	0.865	0.840	0.850	0.825	0.820	0.830	0.823	0.818
	BoW	0.635	–	0.825	0.868	0.890	0.870	0.880	0.898	0.893	0.875	0.873	0.853
	Neural-Based	0.830	–	0.910	0.935	0.930	0.930	0.850	0.860	0.865	0.875	0.860	0.865
	LLM-judge	0.860	–	0.835	0.905	0.905	0.845	0.925	0.925	0.935	0.950	0.925	0.910
	INTERPOL	0.915	–	0.970	0.955	0.975	0.975	0.920	0.920	0.960	0.955	0.920	0.940

Table 9: **Model-wise AUROC on human preference dataset.** Abbreviations: C3.5 = Sonnet-3.5, C4 = Sonnet-4, G-F = Gemini-Flash, G-P = Gemini-Pro, G2-2B = Gemma-2-2B-it, G2-9B = Gemma-2-9B-it, GPT-M = GPT-4o-mini, GPT-4 = GPT-4o, L3.1-8B = Llama-3.1-8B-Instruct, L3.2-3B = Llama-3.2-3B-Instruct, Q2.5-3B = Qwen2.5-3B-Instruct, Q3-8B = Qwen3-8B. Cells highlighted in **red** indicate *family models* of the target high model (e.g., GPT-4o vs GPT-4o-mini). The best scores are highlighted in **bold**.

Target	Method	C3.5	C4	G-F	G-P	G2-2B	G2-9B	GPT-M	GPT-4	L3.1-8B	L3.2-3B	Q2.5-3B	Q3-8B
GPT-4o	Length-Word	0.344	0.462	0.800	0.879	0.652	0.503	0.535	–	0.642	0.610	0.571	0.721
	Length-Char	0.351	0.479	0.795	0.867	0.650	0.513	0.534	–	0.632	0.602	0.576	0.706
	TF-IDF	0.866	0.883	0.780	0.808	0.749	0.755	0.713	–	0.716	0.719	0.616	0.817
	BoW	0.827	0.855	0.775	0.819	0.756	0.752	0.776	–	0.733	0.745	0.633	0.823
	Neural-Based	0.890	0.956	0.966	0.984	0.981	0.975	0.546	–	0.952	0.965	0.610	0.952
	INTERPOL	0.902	0.980	0.961	0.978	0.982	0.977	0.762	–	0.916	0.934	0.762	0.969
	INTERPOL	0.902	0.980	0.961	0.978	0.982	0.977	0.762	–	0.916	0.934	0.762	0.969
Gemini-Pro	Length-Word	0.910	0.845	0.565	–	0.844	0.885	0.879	0.860	0.834	0.844	0.858	0.806
	Length-Char	0.896	0.817	0.542	–	0.826	0.872	0.867	0.844	0.823	0.836	0.843	0.795
	TF-IDF	0.965	0.957	0.848	–	0.919	0.940	0.930	0.932	0.917	0.923	0.935	0.919
	BoW	0.952	0.937	0.905	–	0.955	0.959	0.957	0.958	0.944	0.955	0.957	0.948
	Neural-Based	0.946	0.945	0.891	–	0.941	0.940	0.947	0.947	0.947	0.948	0.947	0.946
	INTERPOL	0.983	0.987	0.972	–	0.984	0.985	0.975	0.975	0.976	0.977	0.973	0.984
	INTERPOL	0.983	0.987	0.972	–	0.984	0.985	0.975	0.975	0.976	0.977	0.973	0.984
Sonnet 4	Length-Word	0.342	–	0.783	0.845	0.659	0.553	0.539	0.550	0.647	0.623	0.587	0.699
	Length-Char	0.662	–	0.235	0.183	0.357	0.463	0.479	0.467	0.374	0.402	0.426	0.324
	TF-IDF	0.660	–	0.908	0.947	0.950	0.931	0.937	0.934	0.913	0.910	0.925	0.906
	BoW	0.756	–	0.891	0.942	0.969	0.952	0.962	0.962	0.964	0.935	0.948	0.938
	Neural-Based	0.858	–	0.904	0.914	0.907	0.909	0.916	0.918	0.918	0.919	0.919	0.908
	INTERPOL	0.961	–	0.975	0.979	0.991	0.989	0.969	0.964	0.980	0.984	0.963	0.976
	INTERPOL	0.961	–	0.975	0.979	0.991	0.989	0.969	0.964	0.980	0.984	0.963	0.976

($12 \times 1,400 = 16,800$ generations) using a unified decoding protocol. Local generations are produced with a high-throughput vLLM engine configured with sampling temperature 0.7, $\text{top-}p = 0.9$, a maximum of 4096 generated tokens, and stop sequences $\{<|eot_id|>, <|end_of_text|>\}$. For stability in cross-model comparisons, API-based generations are obtained with temperature fixed to 0 (greedy decoding).

We adopt GRM-Llama-3.2-3B⁹, which is a

Llama-3.2-3B-Instruct model fine-tuned for reward modeling and serves as our primary detector backbone. Its training is performed with the AdamW optimizer ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$) and a weight decay of 0.01. We employ a cosine learning rate schedule with 100 warmup steps. Training follows a two-stage curriculum, with each stage running for 5 epochs. Stage 1 trains on easier triplets constructed with interpolated negatives at $\alpha_1 = 0.5$, using an initial learning rate of 1×10^{-5} . Stage 2 continues from the best Stage 1 checkpoint, training on harder triplets at $\alpha_2 = 0.75$ with a

⁹Hugging Face model page

halved learning rate of 5×10^{-6} . We use a batch size of 2 with gradient accumulation of 4 (effective batch size 8) and FP16 mixed precision, leveraging DeepSpeed ZeRO-2 for memory efficiency. A weighted triplet loss is applied with coefficients $(\lambda_1, \lambda_2, \lambda_3) = (0.3, 0.3, 1.0)$. The final model checkpoint is selected based on the highest AUROC on the validation set.

For the copymodel, we fine-tune Phi-4-mini-instruct on the same pre-split datasets. The model is trained for 3 epochs with a batch size of 4 and a learning rate of 2×10^{-5} . The maximum sequence length is set to 512, and the best-performing checkpoint is selected based on the lowest validation loss. For the main development, we used a single NVIDIA RTX 6000 Ada 96GB GPU.

B Additional Results

B.1 Model-wise Breakdown Accuracy

We further examine robustness under within-family model comparisons in Table 8. Across all target LLMs, most existing methods exhibit noticeable accuracy drop when distinguishing closely related family variants. This degradation is particularly pronounced for feature-based statistical baselines, which rely on superficial cues and often fail to separate models sharing similar training data, architectures, or distillation pipelines. In contrast, INTERPOL consistently maintains high identification accuracy in these challenging within-family settings. For example, when the target is GPT-4o or Sonnet-4, statistical baselines such as TF-IDF and BoW show substantial drops in accuracy against their respective family models, whereas INTERPOL preserves strong and stable performance.

B.2 Model-wise Breakdown AUROC

In Table 9, we report model-wise AUROC results on the Arena Human preference dataset. Beyond simple classification accuracy, AUROC provides a richer and threshold-independent view of separability, capturing how consistently the detector assigns higher scores to target responses compared to non-target ones. The consistently high AUROC values achieved by INTERPOL across all non-target models confirm that our method offers not only strong accuracy but also robust and reliable discrimination, even in challenging settings such as family models (*e.g.*, GPT-4o vs. GPT-4o-mini, Sonnet-4 vs. Sonnet-3.5).

Table 10: **Target-present identification results in the close-Elo Arena setting.** We rerun representative baselines among models with highly similar Arena Elo scores, focusing on direct competitors rather than wide capability gaps. We report average Accuracy (Acc.) and AUROC across seven close-Elo non-target LLMs. Best scores are highlighted in **bold**. Method abbreviations: Len-W = Length-Word, Len-C = Length-Char, TF-IDF = TF-IDF, BoW = Bag-of-Words, and Neural = Neural-based classification.

Target	Method	Close-Elo Arena	
		Acc.	AUROC
GPT-4o	Len-W	0.491	0.517
	Len-C	0.495	0.518
	TF-IDF	0.755	0.848
	BoW	0.773	0.869
	Neural	0.766	0.852
	INTERPOL	0.949	0.932
Gemini-Pro	Len-W	0.632	0.671
	Len-C	0.627	0.663
	TF-IDF	0.713	0.782
	BoW	0.669	0.703
	Neural	0.851	0.811
	INTERPOL	0.806	0.775
Sonnet-4	Len-W	0.536	0.493
	Len-C	0.523	0.486
	TF-IDF	0.705	0.767
	BoW	0.762	0.872
	Neural	0.882	0.931
	INTERPOL	0.989	0.975

B.3 Close-Elo Competitor Evaluation

To further examine whether INTERPOL relies on broad capability gaps between models, we construct a close-Elo competitor pool from the official LM Arena text leaderboard. The leaderboard snapshot was captured on 2025-11-21. In addition to our three target models, GPT-4o, Gemini-Pro, and Sonnet-4, we include five high-ranked competitors with similar Elo ratings: GLM-4.6 (Rank 11; Elo 1428), Qwen3-Max-2025-09-23 (Rank 13; Elo 1423), Grok-4-Fast (Rank 19; Elo 1420), Kimi-K2-0905-preview (Rank 21; Elo 1416), and DeepSeek-V3.1 (Rank 22; Elo 1416).

As shown in Table 10, this close-Elo setting is substantially harder than the standard broad-pool evaluation. INTERPOL achieves the strongest average Accuracy for GPT-4o and Sonnet-4, reaching 0.949 and 0.989 respectively, while Gemini-Pro becomes more challenging and the Neural baseline is

slightly stronger. Nevertheless, INTERPOL remains competitive for Gemini-Pro and still outperforms feature-based baselines such as Length, BoW, and TF-IDF. This indicates that restricting the evaluation to high-ranked, similarly rated competitors does not remove the attribution signal learned by INTERPOL.

Tables 11 and 12 provide the full pairwise model-level breakdowns for INTERPOL. For GPT-4o and Sonnet-4, INTERPOL maintains consistently high pairwise performance across nearly all close-Elo opponents. Gemini-Pro shows lower performance against several nearby competitors, especially glm-4.6 and DeepSeek-v3.1, suggesting that this target constitutes the hardest case in the close-Elo pool. Overall, these results support that INTERPOL is not driven only by broad model-quality gaps, but captures target-specific response signatures that persist among closely rated Arena models.

C Detailed Analyses

C.1 Additional Score Distributions

In this section, we further present score distribution of the detector for GPT-4o and Sonnet-4, similar to the case of Gemini-Pro in Figure 2. The results of GPT-4o and Sonnet-4 are presented in Figures 3 and 4, respectively.

As shown in Figure 3, (a) the initial detector has no capability to distinguish the target model from the others, as their score distributions almost completely overlap. After the first triplet training stage with interpolation factor $\alpha = 0.5$ (b), the distribution of the target model (red) shifts toward the positive side, indicating that the detector begins to assign systematically higher scores to the target responses. Meanwhile, the non-target models’ scores are pushed leftward, with their average moving toward approximately -5.0 . Finally, under iterative curriculum training with $\alpha = 0.75$ (c), the separation becomes much clearer: the target distribution concentrates further on the positive axis, while the non-target models form a distinct peak around -10 , reflecting the detector’s strengthened discriminative capability.

Moreover, as shown in Figure 4, (a) the initial detector again has no ability to distinguish the target model from the others, with the score distributions almost completely overlapping. After the first triplet training stage with interpolation factor $\alpha = 0.5$ (b), the detector starts to separate certain models such as the blue one, while the distinction

between the target model (red) and another non-target (green) remains ambiguous. However, following iterative curriculum training with $\alpha = 0.75$ (c), the red, green, and blue distributions become much more clearly separated, demonstrating a substantially enhanced detection capability.

Taken together, examining the score distributions across three different target models shows that INTERPOL progressively strengthens the detector with stronger discriminative power, while emphasizing once more that such gains are achieved through nearly cost-free synthetic data generation.

C.2 Data scalability

Figure 5 examines how detection performance scales with the number of training data in the Alpaca dataset. It is observed that Accuracy shows a generally increasing trend as the number of training samples grows from 250 to 1000, suggesting that additional data could further enhance performance. Nevertheless, with only 1000 instructions which corresponds to a modest API collection cost, detector using INTERPOL already achieves strong results (average Accuracy above 0.918 across three target LLMs). This demonstrates that INTERPOL does not rely on prohibitively large datasets: even with a few hundred samples the detector remains competitive, and with 1000 samples it already provides robust accuracy in distinguishing state-of-the-art models.

C.3 Effect of Copy-Model Capacity on Identification Performance

In Table 13, we analyze the impact of copy-model capacity on identification performance by varying the size of the Qwen3 backbone from 0.6B to 8B parameters (Qwen Team, 2024b).

For these Qwen3-based experiments, all parameter-interpolated variants tended to emit the special token “<think>” at the beginning of generations. To ensure comparability across models, we strip these scaffolding tokens and retain only the final response for both training and evaluation.

We keep the evaluation setup identical to Section 4. Increasing the copy-model capacity from 0.6B to 4B improves overall identification performance, whereas further scaling to 8B provides no consistent additional gain across targets and metrics. This trend is consistent with the role of the copy model, which is used to approximate target-specific response characteristics for hard-negative synthesis rather than to maximize general-purpose

Table 11: **Model-wise Accuracy in the close-Elo pool setting.** We evaluate INTERPOL among models with highly similar Arena Elo scores, focusing on direct competitors rather than wide capability gaps. The Avg. column reports the mean across seven close-Elo non-target models.

Target	GPT-4o	Gemini-Pro	Sonnet-4	Grok-4-Fast	DeepSeek-V3.1	GLM-4.6	Kimi-K2	Qwen3-Max	Avg.
GPT-4o	–	0.985	0.975	0.935	0.950	0.975	0.950	0.875	0.949
Gemini-Pro	0.825	–	0.925	0.880	0.705	0.580	0.860	0.870	0.806
Sonnet-4	0.975	1.000	–	0.985	0.995	0.995	0.990	0.980	0.989

Table 12: **Model-wise AUROC in the close-Elo pool setting.** We evaluate INTERPOL among models with highly similar Arena Elo scores, focusing on direct competitors rather than wide capability gaps. The Avg. column reports the mean across seven close-Elo non-target models.

Target	GPT-4o	Gemini-Pro	Sonnet-4	Grok-4-Fast	DeepSeek-V3.1	GLM-4.6	Kimi-K2	Qwen3-Max	Avg.
GPT-4o	–	0.976	0.959	0.902	0.936	0.960	0.930	0.858	0.932
Gemini-Pro	0.781	–	0.872	0.853	0.684	0.554	0.827	0.854	0.775
Sonnet-4	0.967	0.985	–	0.970	0.973	0.985	0.973	0.969	0.975

language-modeling capability. These results suggest that a moderate-capacity copy model is sufficient for effective hard-negative generation, and that simply increasing copy-model size does not necessarily translate into better identification performance.

C.4 Generalization beyond Training-Time Conditions

We next ask whether this performance depends on fixed training-time conditions. In particular, we examine three possible sources of overfitting: the set of non-target models observed during training, the query source used to train the detector, and the target model’s decoding configuration at test time. If INTERPOL merely memorizes a fixed candidate pool, exploits source-specific prompt artifacts, or relies on a particular decoding setting, its performance should degrade substantially when these conditions change.

Candidate-pool generalization. We first evaluate whether the detector depends on a fixed training-time competitor set. For each target, we train the detector using only the seven close-Elo competitors and evaluate it on three splits: the seen competitors, nine unseen outsider models excluded from training, and the full composed pool of 16 non-target models.

As shown in Table 14, INTERPOL achieves the strongest performance on the seen close-Elo competitors, but also generalizes to unseen outsider models excluded from the training-time competitor pool. On the full composed pool, INTERPOL maintains 0.887 Accuracy and 0.884 AUROC on average across the three target LLMs. These results indicate that INTERPOL is not simply learning a fixed lookup over predefined competitors. Instead,

it captures target-discriminative response patterns that transfer beyond the training-time candidate pool.

Cross-source transfer. We further test whether INTERPOL relies on source-specific prompt artifacts. To do so, we train the detector on one query source and evaluate it on the other: Alpaca→Arena and Arena→Alpaca.

As shown in Table 15, INTERPOL remains effective under both transfer directions. For GPT-4o, which is the hardest target in our experiments, INTERPOL still achieves 0.868 Accuracy / 0.838 AUROC for Alpaca→Arena and 0.883 Accuracy / 0.861 AUROC for Arena→Alpaca. For Gemini-Pro and Sonnet-4, transfer performance is even stronger, with AUROC values above 0.91 across both directions.

These results demonstrate that the attribution signal learned by INTERPOL is source-robust: it transfers across substantially different query distributions and reflects model-specific response patterns rather than dataset-specific artifacts.

Robustness to decoding configurations. We further examine whether INTERPOL remains effective when the target model’s decoding configuration differs from that used to construct the training corpus. In our main experiments, API-model responses are generated using greedy decoding. We therefore regenerate only the 200 Gemini-Pro test responses using temperature 0.7 and top- $p = 0.9$, while keeping the trained detector and the responses from the eleven non-target models fixed. This results in 2,200 target–opponent test pairs and requires no detector retraining.

Under greedy decoding, INTERPOL achieves 0.946 Accuracy and 0.967 AUROC. With temper-

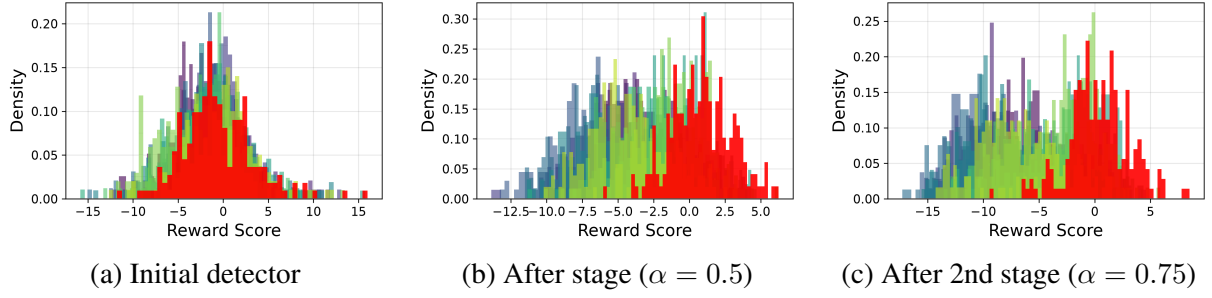


Figure 3: **Score distributions of the detector.** The detector is trained to identify GPT-4o as the target LLM. **Red** indicates the scores of the target model.

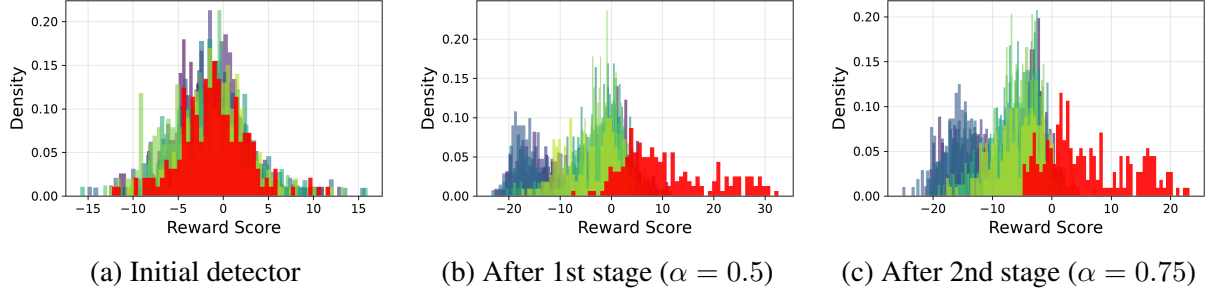


Figure 4: **Score distributions of the detector.** The detector is trained to identify Sonnet-4 as the target LLM. **Red** indicates the scores of the target model.

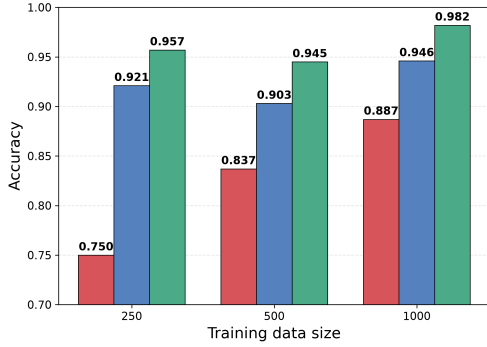


Figure 5: **Data scalability of INTERPOL.** Performance trend with the proposed training framework as the number of training samples increases from 250 to 1000. Accuracy shows a generally linear improvement with larger training sets across all three target LLMs. Red bars represent GPT-4o, blue bars represent Gemini-Pro, and green bars represent Sonnet-4.

ature 0.7 and top- $p = 0.9$, performance remains nearly unchanged at 0.949 Accuracy and 0.964 AUROC. Thus, changing the target-side decoding configuration results in only a +0.26 percentage-point change in Accuracy and a -0.29 percentage-point change in AUROC. These results indicate that the learned target-specific response signatures are robust to moderate changes in decoding and are not tied to the greedy-decoding configuration used during data collection.

C.5 Additional Interpolation-Trajectory Validation

Table 16 extends the *Validation of interpolation-based difficulty* analysis in Section 4.2 to Gemini-Pro and Sonnet-4. Consistent with GPT-4o, target-response PPL decreases monotonically with α , while embedding similarity to the target responses increases. Thus, the interpolation ordering is consistent across all three target LLMs.

C.6 Extended Evaluation on GPT-5

To examine whether INTERPOL remains effective for a more recent frontier model, we extend the target-present identification experiment to GPT-5¹⁰. Following the same evaluation protocol as Table 1, we replace GPT-4o with GPT-5 as the target model, collect responses for the same 1,400-query pool, and train a target-specific detector using the same Phi-4-mini copy-model backbone. We then compare INTERPOL against the same statistical identification baselines.

As shown in Table 17, INTERPOL achieves 0.960 Accuracy and 0.983 AUROC. The strongest statistical baseline reaches 0.821 Accuracy with TF-IDF and 0.906 AUROC with BoW, corresponding to improvements of 13.91 and 7.64 percentage points, respectively. These results indicate that INTERPOL remains effective for GPT-5 despite the substantial

¹⁰[gpt-5-2025-08-07](#)

Table 13: **Effect of copy-model capacity on identification performance.** We vary the capacity of Qwen3-based copy models from 0.6B to 8B parameters on the Alpaca dataset. Increasing capacity from 0.6B to 4B improves overall performance, while further scaling to 8B provides no consistent additional gain across targets and metrics, suggesting that effective copy-modeling does not require a large-capacity backbone.

Copy Model	GPT-4o		Gemini-Pro		Sonnet-4	
	Acc.	AUROC	Acc.	AUROC	Acc.	AUROC
Qwen3-0.6B	0.826	0.819	0.941	0.947	0.938	0.942
Qwen3-4B	0.839	0.845	0.963	0.964	0.991	0.953
Qwen3-8B	0.878	0.824	0.949	0.963	0.954	0.955

Table 14: **Candidate-pool generalization.** The detector is trained using only seven similar-Elo competitors and evaluated on seen competitors, unseen outsider models, and the full composed pool. The results show that INTERPOL generalizes beyond a fixed training-time candidate set.

Evaluation split	# of Models	Acc.	AUROC
Seen competitors	7	0.929	0.958
Unseen outsiders	9	0.854	0.814
All composed	16	0.887	0.884

Table 15: **Cross-source transfer between Alpaca and Arena** We train the detector on one query source and evaluate it on the other. Strong transfer performance indicates that INTERPOL does not merely exploit source-specific prompt artifacts.

Target	Alpaca → Arena		Arena → Alpaca	
	Acc.	AUROC	Acc.	AUROC
GPT-4o	0.868	0.838	0.883	0.861
Gemini-Pro	0.940	0.976	0.928	0.956
Sonnet-4	0.922	0.937	0.896	0.918

capability gap between the frontier target and the lightweight copy-model backbone.

While a fixed 4B-scale copy backbone may not be equally suitable for every future frontier model, INTERPOL is model-agnostic and can use stronger open-source copy backbones as target capability increases.

C.7 Qualitative Interpretability Analysis

To complement the aggregate erasure-based results in Table 5, we examine a representative GPT-4o example for which INTERPOL correctly identifies the target response while TF-IDF incorrectly favors the non-target response.

We consider the prompt “Describe the benefits of virtual reality in healthcare.” For this example, INTERPOL correctly identifies the GPT-4o response with a discrimination margin of 17.16, whereas

Table 16: **Validation of the interpolation trajectory across target models.** Target-response perplexity (PPL) decreases as α increases toward the target-mimicking copy endpoint, while generation-level embedding similarity (Emb. Sim.) increases consistently.

Target	Model	PPL ↓	Emb. Sim. ↑
GPT-4o	Base (Phi-4)	2.114	–
	$\alpha = 0.25$	1.940	0.795
	$\alpha = 0.50$	1.813	0.802
	$\alpha = 0.75$	1.724	0.815
	Copy	1.668	–
Gemini-Pro	Base (Phi-4)	4.190	–
	$\alpha = 0.25$	3.811	0.693
	$\alpha = 0.50$	3.516	0.707
	$\alpha = 0.75$	3.306	0.727
	Copy	3.187	–
Sonnet-4	Base (Phi-4)	3.977	–
	$\alpha = 0.25$	3.494	0.746
	$\alpha = 0.50$	3.125	0.758
	$\alpha = 0.75$	2.864	0.771
	Copy	2.701	–

TF-IDF incorrectly favors the opponent response. The individual words receiving high importance under INTERPOL include ordinary explanatory terms such as “improve,” “adherence,” “treatment,” “enhancing,” “outcomes,” “skills,” “education,” and “healthcare.”

To determine whether these cues are merely isolated lexical markers or belong to a coherent semantic span, we additionally measure the effect of removing each sentence from the response. The sentence causing the largest reduction in the INTERPOL margin is: “Virtual reality (VR) is increasingly being integrated into healthcare, offering a range of benefits that enhance both patient care and medical training.”

Among the 354 words in the response, words belonging to this sentence receive an average importance rank of 138.43 under INTERPOL, compared with 174.29 under TF-IDF and 194.07 under BoW, where a lower rank indicates greater im-

Table 17: **Extended identification evaluation on GPT-5**. We evaluate target-present identification with GPT-5 as the target model under the same setting as the main experiments. Best results are highlighted in **bold**.

Method	Acc.	AUROC
Len-W	0.448	0.381
Len-C	0.446	0.384
TF-IDF	0.821	0.904
BoW	0.797	0.906
INTERPOL	0.960	0.983

portance. Thus, INTERPOL assigns substantially greater importance to the same explanatory span than either lexical baseline. Together with the aggregate erasure results in Table 5, this example suggests that INTERPOL captures not only isolated word-frequency patterns but also semantically coherent explanatory phrasing characteristic of the target model’s responses.

C.8 Cost Analysis

As training detector with INTERPOL incurs the additional cost, one might ask whether it is more effective than simply collecting more target-specific responses. Regarding this, we note that additional target-model API collection and INTERPOL serve complementary roles: API scaling increases the number of real target responses available for supervision, whereas INTERPOL extracts more structured and discriminative supervision from the responses already collected through interpolated hard negatives. Nevertheless, under a fixed resource budget, it is useful to compare their marginal cost and performance gains.

As shown in Figure 5, increasing the training set from 250 to 500 responses for each of the three targets requires 750 additional API generations, costing approximately \$2.44, while improving average Accuracy from 0.876 to 0.895 (+0.019). In contrast, Table 3 shows that the full INTERPOL configuration improves average Accuracy from 0.863 to 0.951 (+0.088) under a fixed real-response budget. The additional computation required by INTERPOL—including copy-model fine-tuning, interpolated-response generation, and one additional detector-training stage—takes approximately 51.5 GPU minutes per target, corresponding to about \$1.14 at \$1.33 per GPU-hour. Thus, under comparable marginal costs, INTERPOL yields substantially larger gains by extracting richer supervision from the available target responses rather than merely increasing the number of API-collected ex-

amples.

D Usage of AI assistants

In preparing this work, we used AI-based writing assistants to improve sentence structure, correct grammatical errors, and enhance overall readability. These tools were employed solely for language refinement and did not contribute to the development of technical content, research methodology, or experimental analysis. All scientific ideas, results, and conclusions presented in paper were conceived and authored entirely by the researchers. The use of AI assistance was restricted to editorial purposes and did not affect the originality or intellectual contributions of the work.