UNSUPERVISED PREFERENCE ALIGNMENT OF LLMS FROM MULTI-TURN DIALOG LOGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-turn preference alignment is hindered by long-range context dependencies and costly human annotation. We present a fully unsupervised framework that constructs Direct Preference Optimization (DPO) data from production dialog logs without human labels or LLM judges during pair construction. Responses from a stronger deployed LLM system are chosen; weaker-model responses to the same context are rejected. This cross-model construction creates a source-model confound: preference-alignment methods, including DPO, can exploit *generative fingerprints*—model-specific stylistic and distributional patterns—instead of semantic quality. We formalize the confound and show that shared, sentence-level paraphrasing of both responses can reduce fingerprint separability while bounding semantic-signal drift; our pipeline combines normalization, length balancing, and both-side paraphrasing. Across 77K–11M pairs from $\sim$12M real multi-turn conversations, the regularized pipeline improves with scale, whereas standard DPO overfits at all scales. Phi-4 (14B) gains up to $+7.8$pp held out, and $+6.2$pp, $+3.1$pp, and $+5.0$pp on WildChat, AlpacaEval, and Arena-Hard prompts under our binary rubric. Using Phi-family preference data, Qwen-2.5-14B gains $+6.7$pp held out and $+2.9$pp on WildChat. Two additional judge families and a target-independent paraphraser control confirm the key held-out gain, and public WildChat data replicates the pipeline.

1 INTRODUCTION

Aligning LLMs with human preferences underlies conversational AI. Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2023) require comparisons in which one response is preferred over another, but collecting them at scale requires costly human annotation or LLM-based judges.

We instead construct preference data from approximately 12M real-world, multi-turn production conversations. This English-only corpus is broad and non-domain-specific, spans open-ended topics across diverse requests, and was not collected from a predefined community, geography, demographic group, or application domain; its breadth rules out the usual in-domain effect, in which fine-tuning gains come from fitting a narrow topic or population: a model cannot fit this corpus by specializing to a domain the way it can with domain-specific data, so any improvement must reflect a general gain in response quality. Each record is (x, q, y): x contains every preceding user and assistant turn, q is the final user message, and y is a candidate final answer. No earlier turn is dropped, so candidates and the trained policy condition on the complete history. Because q expresses unresolved intent while prior turns may be partial or exploratory, the final turn provides an implicit preference target.

For each (x, q), a stronger deployed LLM system supplies the chosen response and a weaker model from the target family supplies the rejected response (e.g., Phi-4 (Abdin et al., 2024b) and Phi-3-mini (Abdin et al., 2024a)). Pair construction uses neither human labels nor LLM judges. Individual comparisons can be wrong, but the source system is better in expectation, yielding bounded, learnable noise.

Cross-model construction also creates a source-model confound. Let Z be the generating model; $\ell \in \{0, 1\}$ the chosen/rejected training label; $Q(x, y)$ the latent semantic quality of y in context x; and $I(A; B \mid x)$ conditional mutual information. Because ℓ is assigned by source identity rather than direct observation of Q, $I(Z; \ell \mid x)$ is near-maximal. Distinct vocabulary, formatting, length, and reasoning styles form generative fingerprints, allowing preference optimization, including DPO, to identify the source instead of semantic quality.

Under bounded noise, learning remains possible if these shortcuts are controlled. Let P_{bal}, P_{para}, and P_{norm} denote length balancing, both-side paraphrasing, and style normalization; we use $P =$

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

$P_{\text{bal}} \circ P_{\text{para}} \circ P_{\text{norm}}$ (normalization first). Balancing blocks response length as a proxy, paraphrasing aligns local realizations, and normalization strips low-level formatting. Paraphrasing processes every sentence of both responses through one 14B model and shared channel. Under the stated bounded-drift and fingerprint-contraction assumptions, P reduces source information while preserving the semantic preference gap up to a bounded perturbation. This establishes noisy pairwise learnability, not guaranteed improvement in free-form generation.

Controlled 77K-pair ablations compare each component and alternative composition through training dynamics, stability, reward-fitting behavior, and effective preference-gap evolution. Normalization and balancing help partially; paraphrasing produces the strongest pattern consistent with reduced fingerprint reliance, while stronger rejectors weaken the learnable signal. Cross-family tests train Qwen-2.5-14B with Phi-3-mini rejections and Phi-4 with Qwen-2.5 rejections, testing transfer rather than dependence on one rejection model's artifacts. From 77K to 11M pairs, standard DPO immediately overfits at every tested scale and gains nothing from more data, whereas the regularized pipeline learns stably and improves consistently. Models are selected on development data and evaluated on held-out and public prompt sets. Phi-4 (14B) gains $+7.8$pp held out, and $+6.2$pp, $+3.1$pp, and $+5.0$pp on WildChat (Zhao et al., 2024), AlpacaEval (Dubois et al., 2024), and Arena-Hard (Li et al., 2024) prompts under our binary rubric; Qwen-2.5-14B gains $+6.7$pp held out and $+2.9$pp on WildChat using Phi-family data. Two additional judge families, a target-independent paraphraser control, and a public WildChat replication support the findings.

In summary, our contributions are:

- We introduce unsupervised preference learning from dialog logs using (x, q, y) and implicit final-turn feedback.
- We characterize and mitigate the source-model confound and fingerprint shortcut in preference optimization, including DPO.
- We analyze learnability under bounded noise and show, under explicit assumptions, how semantic-preserving transformations reduce shortcuts while approximately preserving semantic quality.
- We propose a progressive data-level regularization pipeline designed to reduce label-correlated fingerprint cues.
- We combine ablations, cross-model transfer, and scaling experiments, obtaining stable unsupervised learning and 6.7–7.8pp absolute gains across model families on general multi-turn dialog, with cross-judge, paraphraser-control, and public-data checks.

2 Related Work

Preference optimization and synthetic supervision. RLHF learns from human preferences and reward models (Ouyang et al., 2022), while DPO optimizes preferences directly (Rafailov et al., 2023). Related objectives include IPO (Azar et al., 2023), KTO (Ethayarajh et al., 2024), and cDPO (Mitchell, 2023); model-generated supervision includes Constitutional AI (Bai et al., 2022), UltraFeedback (Cui et al., 2023), self-rewarding models (Yuan et al., 2024), and sequential DPO (Kim et al., 2024). These methods provide or estimate preference labels; we instead study unlabeled capability-gap pairs and the confound induced by their construction.

Dialog data and model-generated responses. Rejection sampling is common in RLHF pipelines (Dong et al., 2024), and distillation transfers stronger-model behavior to weaker models (Hinton et al., 2015). Preference distillation methods such as PLaD (Zhang et al., 2024) and daDPO (Zhang et al., 2025) also pair a strong teacher answer against a weaker one. Let M_1 produce the chosen answer, M_2 the rejected one, and M_3 be the model trained. In these methods both answers are generated on task-specific or first-turn prompts and the student that produced the rejection is the one trained ($M_2{=}M_3$). We keep M_1's logged answer, generate only M_2's rejection, and train an independent M_3 on real multi-turn dialogs with complete histories; the resulting pairs are a reusable dataset rather than student-specific, so pairs built with Phi-family rejections also improve Qwen targets (Section 5.3). WildChat provides large-scale multi-turn dialogs (Zhao et al., 2024), and prior work studies multi-turn and trajectory-level preference learning (Shi et al., 2024). We instead use final-turn responses from real dialog logs as chosen and weaker-model generations as rejected, without explicit feedback; this makes source identity predictive of the preference label.

Shortcut learning and model fingerprints. Models exploit spurious correlations and annotation artifacts (Geirhos et al., 2020), (Gururangan et al., 2018), (Poliak et al., 2018), while generated-text detection and authorship attribution reveal identifiable LLM source signatures (Uchendu et al., 2020), (Mitchell et al., 2023), (Hans et al., 2024). In cross-model preference construction, this signal be-

comes label-correlated, allowing preference-optimization methods, including DPO, to separate responses by source-model fingerprint rather than semantic quality. The closest prior diagnosis is Wang et al. (2025), who show that heterogeneous synthetic safety-alignment data is source-separable and that this separability creates preference-learning shortcuts; we do not claim first observation of model-source fingerprints. Our setting differs in that the pairs come from real dialog logs where source identity is perfectly coupled to the label, and our contribution is the mitigation of this confound and its scaling to millions of pairs.

Mitigation and noisy-label learning. Data augmentation and counterfactual transformations can reduce spurious-feature reliance (Kaushik et al., 2020), (Feng et al., 2021). We apply one shared paraphrasing transformation to both responses, preserving meaning while suppressing source fingerprints. Classical bounded-noise results establish learnability when labels remain informative in expectation (Bartlett et al., 2006), (Massart & Nédélec, 2006), (Tsybakov, 2004); label smoothing and perturbation offer related regularization (Szegedy et al., 2016), (Xie et al., 2016), (Lukasik et al., 2020). Our pipeline instead targets label-correlated confounds at the data level.

3 Dialog Logs as Unsupervised Preference Signal

In this section, we describe how dialog logs can be leveraged as a source of unsupervised preference supervision, together with the theoretical justification and the construction of chosen–rejected pairs.

3.1 The Data Opportunity

Production LLM systems generate large volumes of multi-turn conversations whose responses come from a strong deployed model. Typically used only for aggregate analytics, these logs carry an implicit supervision signal: each response is a capable model's attempt to satisfy a real user need. Using them directly for supervised fine-tuning (SFT) is ineffective: the data is noisy (failed interactions, misunderstandings, incomplete answers), and SFT treats all responses equally, with no mechanism to separate better from worse answers. In contrast, policy-alignment methods, e.g., DPO, explicitly model preference relations and can learn from noisy comparisons. We exploit this by constructing preference pairs without annotation: responses from a stronger model are treated as *chosen*, while responses generated by a weaker model for the same dialog context are treated as *rejected*. Although individual comparisons may be incorrect, the stronger model is better *in expectation*, yielding a weak but learnable signal. This intuition is formalized in the next section.

3.2 Learnability of Unsupervised Model-Pair Preferences

Abstractly, we construct preference pairs without human annotation by sampling y^+ from a stronger model and y^- from a weaker model for the same dialog context x. Let $Q(x, y)$ denote latent semantic quality. We assume the stronger model is better only in expectation, not always.

Assumption 1 (Bounded preference noise). For some $\delta > 0$, $\Pr[Q(x, y^+) > Q(x, y^-)] \geq \frac{1}{2} + \delta$. This is a marginal bounded-noise condition in the spirit of Massart-type noise (Massart & Nédélec, 2006): labels are noisy but better than random on average. Being weaker than the conditional Massart condition of classical guarantees, it serves as a motivating premise, not a theorem hypothesis. Table 1 estimates the effective δ as the Opus-judged quality gap between chosen and rejected responses under each preprocessing condition.

Proposition 1 (DPO as noisy pairwise classification). Under Assumption 1, the unsupervised preference triplets (x, y^+, y^-) define a noisy pairwise classification problem whose labels are informative on average. Since the DPO objective is a logistic loss over preference differences, minimizing DPO can recover a scoring function aligned with the noisy preference ordering when the standard conditional-noise conditions of the cited results hold.

Justification. DPO minimizes $-\log \sigma(\beta f_\theta)$, where $f_\theta(x, y^+, y^-) = \log \frac{\pi_\theta(y^+|x)}{\pi_\theta(y^-|x)} - \log \frac{\pi_{\text{ref}}(y^+|x)}{\pi_{\text{ref}}(y^-|x)}$. This is the logistic loss for predicting whether y^+ is preferred to y^-, whose labels under Assumption 1 have average error below $1/2$. Classification-calibration results for Massart-type noise (Bartlett et al., 2006; Massart & Nédélec, 2006; Tsybakov, 2004) require the conditional condition, so we invoke them as motivation, not as a proof that DPO recovers the Bayes-optimal preference classifier. Proposition 1 is thus a preparatory observation; it concerns the preference function, not free-form generation quality, which also depends on capacity, optimization, decoding, and on whether the learned signal is semantic rather than a shortcut. The main analysis is Proposition 2 (Section 4.1).

3.3 Customer signals in dialog structure for building training examples

Figure 1(a) illustrates our data construction on a shortened example. A three-turn conversation provides the dialog context (turns 1–2) and final user question (turn 3). The original system answer

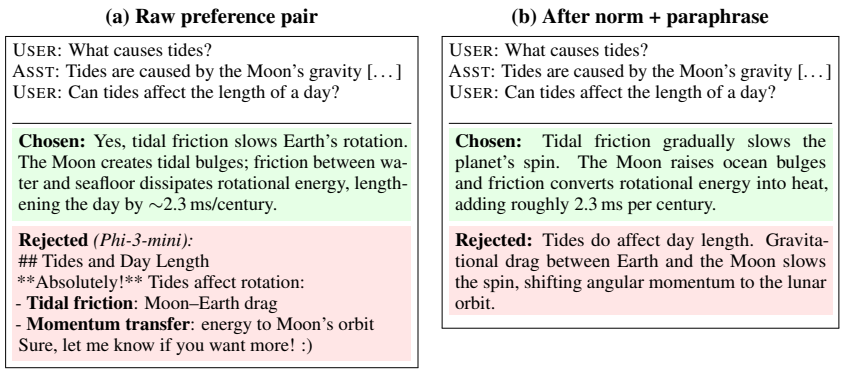


Figure 1: Preference pair construction and regularization. **(a)** The rejected response (Phi-3-mini) uses markdown, bold, bullets, and filler—trivial fingerprints. **(b)** After normalization and paraphrasing through Phi-4, surface asymmetries are reduced.

becomes the *chosen* response, while a weaker model (Phi-3-mini) generates the *rejected* counterpart. The two responses exhibit strong stylistic asymmetries—markdown headers, bold text, and bullet lists—forming generative fingerprints that preference-alignment methods, e.g., DPO, can exploit as shortcuts (Sec. 4). Multi-turn dialogs contain an implicit quality signal absent in single-turn data: the user's decision to continue the conversation. We construct training examples from the *final* question–answer pair, as it reflects the latest attempt to satisfy unresolved user intent. While earlier turns may be exploratory or incomplete, the final exchange is more likely to be satisfactory, making it a natural anchor for preference learning. This supervision signal is noisy—users may abandon conversations for reasons unrelated to quality—but it is available at scale without annotation.

Rejection Model Selection. The choice of rejection model is the most consequential design decision in our framework. We require a model that is (a) systematically less capable than the chosen model, so that preference labels are correct on average, and (b) sufficiently different in capability that the resulting signal is learnable. We compare two rejection strategies when training Phi-4 (14B): (i) Phi-3-mini (3.8B) rejections, a much weaker same-family model, and (ii) Phi-4 rejections, a similar-capability control. The former creates a substantial capability gap and thus informative noisy supervision. The latter makes rejected responses too competitive, weakening the preference signal. Empirically, this control performs poorly, consistent with the requirement that chosen responses be better than rejected responses often enough to provide a learnable signal.

4 The Fingerprint Problem and Regularization Pipeline

We use the term *generative fingerprint* for the model-specific stylistic and distributional signatures of generated text—lexical choice, formatting, sentence structure, and local discourse patterns—which persist even after surface formatting differences between chosen and rejected responses are removed. These signals are not semantic, yet they are highly predictive and easily exploited: preference optimization, including DPO, can reach near-perfect training accuracy by identifying the source model rather than learning semantic preference, separating response distributions instead of capturing quality differences, which leads to rapid overfitting and poor generalization. We thus develop a theoretical framework in which fingerprints act as a confounding signal, and design a regularization pipeline—normalization, length balancing, and both-side paraphrasing—to suppress these shortcuts.

4.1 Theory for Reducing Generative Fingerprints

We formalize semantic-preserving transformations as operators that contract local stylistic information while approximately preserving semantic content. We denote by P a transformation drawn from a class of semantic-preserving operators, including normalization, paraphrasing, and their composition; in practice, we use $P = P_{\text{bal}} \circ P_{\text{para}} \circ P_{\text{norm}}$, i.e., normalization, then paraphrasing, then balancing.

Let x be the multi-turn dialog context and $y = (s_1, \ldots, s_n)$ an answer decomposed into local units. We denote by $Q(x, y)$ the semantic quality of the response, i.e., how well y satisfies the user request given the dialog context in terms of correctness, relevance, completeness, and reasoning. In contrast, we define the model fingerprint $M(y) = h\left(\frac{1}{n}\sum_i \psi(s_i)\right)$, where ψ extracts local stylistic and token-level features. Here h is a source-identification classifier, $h : \mathcal{Y} \to \mathcal{Z}$ (equivalently $h(\phi(y))$ over response features ϕ), where $\mathcal{Z}$ is the set of generator identities; its accuracy measures source

separability. Thus, $Q(x, y)$ depends primarily on global context-answer compatibility, while $M(y)$ depends on local surface realization.

In cross-model preference construction, the observed preference signal combines both components: $\Delta_{\text{obs}} = \Delta Q + \Delta M$, where $\Delta Q = Q(x, y^+) - Q(x, y^-)$ and ΔM captures separability induced by model fingerprints. When $|\Delta M| \gg |\Delta Q|$, preference-optimization methods, including DPO, can minimize their objectives by learning model identity rather than semantic quality. This decomposition is bookkeeping, not an independence assumption: Q and M may be correlated, and a transformation P may affect both. The analysis below therefore imposes a bound on the change in Q (Assumption 2) and a contraction condition on source information (Assumption 3) rather than assuming that either quantity is untouched. Since Q depends on the full conversation history while M lives in local surface cues, transformations operating on local realization (e.g., normalization, paraphrasing) affect $M(y)$ much more strongly than $Q(x, y)$.

Assumption 2 (Bounded semantic-quality drift). For a semantic-preserving transformation P, $|Q(x, P(y)) - Q(x, y)| \leq \epsilon_Q$. This assumption reflects that normalization and paraphrasing mainly alter surface realization while largely preserving semantic content and dialog grounding. We verify this experimentally in Section 5.1. This bound permits the pairwise gap to change by up to $2\epsilon_Q$; Section 5.1 quantifies the empirical gap retained after each transformation.

Assumption 3 (Fingerprint contraction). Applying the same transformation P to both chosen and rejected responses reduces dependence on the source model: $I(\tilde{Y}; Z \mid x) < I(Y; Z \mid x)$, where $Z \in \{S, W\}$ denotes the source model (strong or weak), Y is the original response, and $\tilde{Y} = P(Y)$ is the transformed response. Here, $I(\cdot; \cdot \mid x)$ denotes conditional mutual information, i.e., how predictive the response is of its source model. Since different models exhibit distinct stylistic patterns, this signal is reduced by applying a shared transformation that maps both responses into a common realization space and weakens model-specific cues. Transformations such as normalization remove low-level formatting signals, while paraphrasing further aligns local linguistic realizations. Assum. 3 is a premise rather than a derived fact: the data-processing inequality alone guarantees only that $I(\tilde{Y}; Z \mid x) \leq I(Y; Z \mid x)$, and the strict inequality is assumed for the transformations we use.

Proposition 2. Let $Y^+ \sim p_S(\cdot \mid x)$ and $Y^- \sim p_W(\cdot \mid x)$, and let $\tilde{Y} = P(Y)$. Under assumptions 2 and 3, applying P reduces source information while preserving the semantic preference gap:

$$I(\tilde{Y}; Z \mid x) < I(Y; Z \mid x), \text{and}$$

$$\left|\left[Q(x, P(Y^+)) - Q(x, P(Y^-))\right] - \left[Q(x, Y^+) - Q(x, Y^-)\right]\right| \leq 2\epsilon_Q,$$

Proof. Since P is applied after generation and independently of Z, we have $Z \to Y \to \tilde{Y}$. By the data processing inequality, $I(\tilde{Y}; Z \mid x) \leq I(Y; Z \mid x)$, and the strict inequality is exactly Assumption 3; the first claim therefore restates the assumed contraction of source information rather than deriving it. Assumption 2 and the triangle inequality yield the semantic bound. □

Consequence for DPO. Preference optimization, including DPO, can exploit separation over $\Delta_{\text{obs}} = \Delta Q + \Delta M$. If ΔM dominates, the learner can recover model identity rather than answer quality. Indirect evidence for this behavior appears in the training reward dynamics reported in Appendix D.1. Prop. 2 shows that, under assumptions 2–3, applying P reduces source information while preserving ΔQ up to $2\epsilon_Q$, improving the effective signal-to-noise ratio. Thus, the learned preference signal is expected to become more aligned with semantic quality $Q(x, y)$ rather than source-model fingerprints; this is a motivation for the pipeline, not a proof of improved generation.

4.2 Regularization Pipeline

Motivated by the analysis above, we design a progressive regularization pipeline with three components applied to both chosen and rejected responses:

Style normalization (Algorithm 1). We remove surface formatting differences (e.g., markdown, emoji, filler phrases) that provide trivial shortcuts. Raw chosen and rejected responses differ sharply in these surface features (Table 3); normalization removes these measured cues, though not necessarily all source-identifying information. Details are in Appendix B.

Both-side paraphrasing (Algorithm 3). We paraphrase *both* chosen and rejected responses sentence by sentence: each response is split into sentences, every sentence is paraphrased independently through the same Phi-4 paraphraser (temperature 0.1) with the same prompt, and the outputs are reassembled in their original order. Preserving one natural-language sentence is substantially easier than preserving a complete long answer; in a preliminary check, whole-answer paraphrasing by a

Config	Cho.	Rej.	Gap
E0 (raw)	77.8	37.4	40.4
E1 (+norm)	76.8	36.6	40.2
E2 (+norm+bal)	76.2	36.6	39.6
E3 (+norm+para+bal)	73.4	36.4	37.0
E4 (+para)	73.2	35.6	37.6
E5 (+para+norm)	74.0	36.8	37.2

Config	Cho.	Rej.	Gap
E0 (raw)	67.2	23.4	43.8
E1 (+norm)	64.4	23.0	41.4
E2 (+norm+bal)	64.6	22.6	42.0
E3 (+norm+para+bal)	61.4	25.8	35.6
E4 (+para)	67.6	25.8	41.8
E5 (+para+norm)	63.2	24.8	38.4

Table 1: Training-data quality for OTrain and WildChat, respectively, by processing stage (500 matched-sample judge, Opus 4.6). *Gap* = chosen − rejected accuracy; larger gaps provide stronger DPO signal. All experiments use Phi-3-mini (3.8B) rejections.

model weaker than the logged source reduced chosen-answer acceptance from approximately 80% to 70%, which motivates the local operator. Because every sentence of both responses passes through one shared channel, source-specific cues are reduced, though not eliminated, and learning must rely more on content quality. See Appendix B.4.

Length balancing (Algorithm 2). We equalize the length distribution of chosen and rejected responses to prevent length from acting as a proxy for preference. This step is optional and data-dependent (Table 1).

Figure 1(b) illustrates the effect of the pipeline: surface asymmetries are reduced after normalization and paraphrasing. The corresponding text processing components are defined in Section 5.1.

5 Experiments

The experiments test two claims: that cross-model pairs induce a fingerprint shortcut, and that the operator P mitigates it while preserving the preference signal. Section 5.1 describes data, models, training, and the judge protocol, and verifies on the training pairs that each processing stage preserves the chosen–rejected quality gap (Assumption 2). Section 5.2 ablates the pipeline components on a 77K subset, contrasting training dynamics and held-out accuracy across E0–E5, varying rejection strength, and measuring source separability directly; Section 5.3 asks whether the learned signal transfers across model families by exchanging rejection generator and target; Section 5.4 scales training from 77K to 11M pairs to test whether the regularized signal, unlike raw pairs, keeps improving with data, and evaluates on held-out and public prompt sets.

5.1 Experimental Setup

Training data is drawn from proprietary multi-turn dialog logs (∼12M conversations from a production LLM), filtered through a 5-layer quality pipeline (Appendix B.1). Depending on the experiment, we construct between 77K and 11M DPO pairs, referred to as OTrain.

Test data: We use a development set (ODev) of 2K items and a test set (OTest) of 3K items, both randomly sampled from the ∼12M corpus (with no overlap with OTrain). We also evaluate on public benchmarks: WildChat (Zhao et al., 2024) (1,718 decontaminated multi-turn dialogs; decontamination removes dialogs whose IDs exactly match any training example, yielding zero measured ID overlap with training data), AlpacaEval (Dubois et al., 2024) (805 single-turn instructions), and Arena-Hard (Li et al., 2024) (500 challenging queries). Throughout, WC, AE, and AH denote prompts sampled from WildChat, AlpacaEval, and Arena-Hard and scored with our fixed binary rubric; they are cross-distribution tests, not official leaderboard metrics for those benchmarks.

Models and Pipeline Configurations: We use Phi-4 (14B), Phi-3-mini-128k-instruct (3.8B), and Qwen-2.5-14B-Instruct (Yang et al., 2024) for training and evaluation. Phi-3-mini is the rejection model, although we also use Phi-4 (14B) and Qwen-2.5-3B to study the effect of rejection strength. The paraphrasing model is Phi-4 (temperature 0.1). We define six DPO data variants: **E0** raw (system messages stripped only), **E1** normalization, **E2** normalization+balancing, **E3** normalization+paraphrasing+balancing, **E4** paraphrasing only, and **E5** paraphrasing+normalization. In the scaling experiments, **E0**$^+$ denotes the large-scale control in which only the chosen responses were normalized; the 77K E0 arm is raw. Details are in Section 4.2 and Appendices B.2, B.4. For cross-model experiments, **Phi3-E3→Qwen** denotes Qwen trained on E3 data with Phi-3-mini rejections, while **Qwen-E3→Phi4** denotes Phi-4 trained on E3 data with Qwen-2.5 rejections. The arrow points from the rejection-source to the trained model.

Training All experiments use DPO with $\beta = 0.1$, sigmoid loss, and a cosine learning rate schedule with 10% warmup. We use two configurations: (i) ablation scale, i.e., 2 nodes × 8 GPUs (H200), effective batch size 32, learning rate 5×10^{-8}, 2 epochs, max sequence length 4096, DeepSpeed

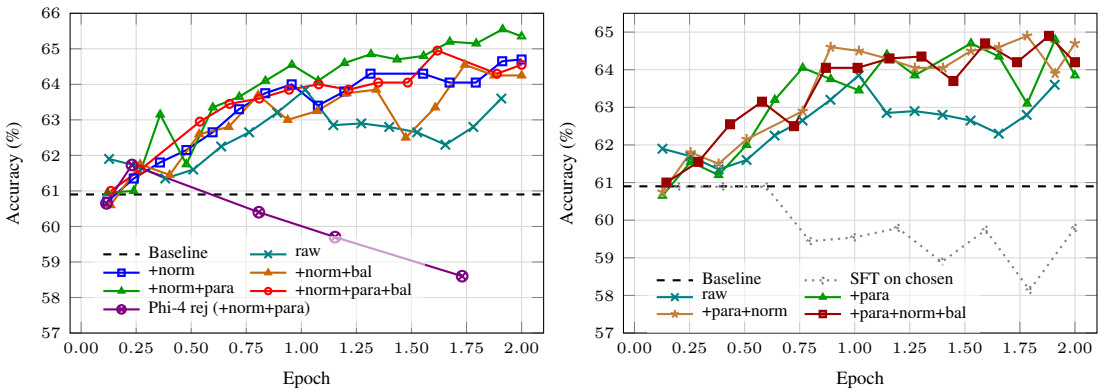


Figure 2: Pipeline ablation on 77K from OTrain. (a) Model based on normalization before paraphrasing, and a model trained on Phi-4 rejections. (b) Model based on paraphrasing first, and a baseline model trained with SFT. Curves are from the batch-80 configuration; Table 2 reports the batch-32 run.

ZeRO Stage 2; (ii) full scale, i.e., 10–50 nodes $\times$ 8 GPUs (H200 or B200), effective batch size 160–800, learning rate 3×10^{-8} (more stable at scale), 2 epochs, max sequence length 4096, DeepSpeed ZeRO Stage 2.

Evaluation We use answer accuracy computed by Claude Opus 4.6 via AWS Bedrock, using a binary accept/reject rubric: given the conversation history and the model's generated answer, the judge determines whether the response is acceptable. We generate answers with vLLM (temperature 0.1) and a fixed evaluation system prompt (SA): "You are a super AI assistant. Your task is to provide accurate, and complete answers to user queries during a chat." We use a development set (ODev) of 2K items for checkpoint selection, hyperparameter tuning, and ablation comparisons. Final performance is evaluated on OTest and public benchmarks. All benchmark items are converted to a unified format and evaluated with the same Opus 4.6 binary judge. Our protocol differs from AlpacaEval 2.0 and Arena-Hard-Auto (which use GPT-4 pairwise judging); scores are not directly comparable. Opus is used for training-data diagnostics (Table 1), checkpoint selection on ODev, and evaluation of frozen checkpoints; it never constructs or relabels training pairs. To verify that single-judge results are trustworthy, we re-evaluate the ODev-selected E3-77K model and the Phi-4 baseline with two additional judge families, gpt-oss-120B (OpenAI et al., 2025) and Qwen3-Coder-480B (Yang et al., 2025), using the identical prompt; all three judges agree on the sign and significance of the gain (Appendix E.2), so the remaining results are reported with Opus alone. A human calibration audit of the rubric ($\sim$70% agreement, $\sim$6% clear disagreement) is reported in App. C.4, and baseline accuracies with and without the evaluation system prompt in App. C.2.

DPO Training Data Quality Before training, we verify that the regularization pipeline (Section 4.2) preserves the semantic quality gap between chosen and rejected responses. Table 1 evaluates this by running the same Opus 4.6 judge on the training data: for each configuration, we sample 500 DPO pairs and measure accuracy on chosen and rejected responses. The resulting gap serves as a proxy for the effective preference signal (cf. δ in Section 3.2): larger gaps indicate stronger supervision, while smaller gaps lead to weaker, noisier learning signals.

5.2 Ablation: Contribution of Each Pipeline Step

We ablate each pipeline stage on proprietary dialog logs ($\sim$77K rows, Phi-3-mini rejections, lr $= 5 \times 10^{-8}$, 2 epochs, 16 GPUs, Opus 4.6 judge). Panel (a) applies normalization before paraphrasing; panel (b) reverses the order. Figure 2 shows three key results for Phi4 trained model. First, all configurations with Phi-3-mini rejections outperform the untrained Phi4 baseline (60.9%), confirming that unsupervised preference learning is effective despite noisy supervision. Second, +norm+para and +norm+para+bal learn steadily across both epochs, while raw training overfits after epoch 1. Appendix D.1 further shows that raw DPO reaches near-perfect reward accuracy substantially faster than regularized variants, consistent with the model exploiting fingerprint shortcuts. Reward accuracy alone does not identify which feature was learned, so we also measure source separability directly with a pairwise TF–IDF source classifier on identical pair IDs: it drops from 97.2% (E0) to 94.3% (E3) on OTrain and from 90.2% to 81.4% on WildChat, while the judged quality gap remains large (40.4$\rightarrow$37.0 on OTrain, 43.8$\rightarrow$35.6 on WildChat; Table 1). The residual accuracy is expected, since correctness itself predicts the source; what E3 changes is the contraction, and this 3–9 point

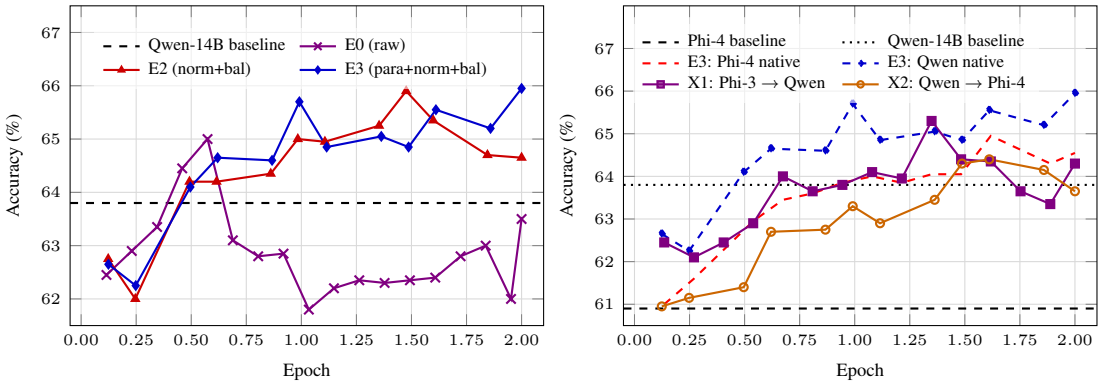


Figure 3: Qwen-2.5-14B experiments 77K from OTrain. (a) Pipeline ablation with Qwen-2.5-3B rejections. (b) Cross-family data transfer. Batch-80 configuration, as in Figure 2.

contraction on identical pairs suffices to change learning. App. E.1 gives the probe and argues why a generic-regularization reading does not fit the ablations. Third, panel (b) shows that applying paraphrasing *before* normalization yields similar final accuracy but less stable dynamics—normalizing first reduces structural asymmetries prior to paraphrasing, consistent with Sec. 4.1.

The Phi-4 model trained with Phi-4-generated rejected answers (panel a, violet) collapses below baseline, confirming that when rejected answers are too similar in quality to chosen answers, the preference signal becomes too weak for effective learning (Section 3.3). The Phi-4 model trained with SFT on chosen responses only (panel b, dotted gray) *never exceeds* its own starting point and drifts below it as training proceeds, showing that simply imitating chosen responses is insufficient and that contrastive preference learning is essential.

5.3 Cross-Model Transfer: Qwen-2.5

We evaluate cross-model transfer by training Qwen-2.5-14B-Instruct on Phi-family DPO data (∼77K rows, Qwen-2.5-3B rejections for native; Phi-3-mini rejections for transfer, $\text{lr} = 5 \times 10^{-8}$, 2 epochs, 16 GPUs). Panel (a) ablates the pipeline using Qwen-native data; panel (b) evaluates cross-family transfer. Figure 3(a) reproduces the same pattern observed with Phi-4: raw training (E0) peaks early and overfits, while regularized pipelines (E2, E3) learn steadily across epochs, with E3 performing best. Panel (b) shows that the learned preference signal transfers across model families: the Qwen-2.5-14B model trained on Phi-3-rejection E3-77K and selected by maximum ODev accuracy reaches 65.3%, close to the native E3 model at 66.0%; both exceed the 63.8% Qwen baseline, indicating that the signal captures semantic quality. In contrast, Phi models using Qwen-generated rejections underperform their native counterpart (X2, Qwen-E3→Phi4, peaks at 64.4% against 65.0% for Phi-4-native E3; X1 denotes Phi3-E3→Qwen), consistent with weaker or less transferable preference signals. These cross-family rows change the rejection generator or the target, not the paraphraser. Appendix E.3 isolates the paraphraser directly: swapping it from Phi-4 to Qwen-2.5-14B on identical pairs yields gains that no judge distinguishes ($p \geq .17$), which weighs against Phi self-style distillation within that matched control; paraphraser, rejection generator, and target are distinct roles.

5.4 Scaling and Generalization

We scale DPO training from 77K to 11M rows on proprietary dialog logs (Phi-3-mini rejections, $\text{lr} = 3 \times 10^{-8}$, 2 epochs, Opus 4.6 judge). Training uses distributed DeepSpeed ZeRO-2 across 10–50 GPU nodes (80–400 H200 GPUs); experiments at 10, 20, 30, and 50 nodes yield consistent final accuracy, indicating that effective batch size (160–800) has limited impact in this regime. The second-largest run (E3, 9.3M rows, 20 nodes) completed in ∼4 days wall-clock time. The 77K ablation (Section 5.2) shows that raw data overfits while regularized pipelines learn steadily. Scaling amplifies this contrast and tests whether improved signal quality translates into better generalization.

Learning behavior. Figure 4 reports ODev learning curves across scales. (i) E0$^+$ at 11M rows, where + denotes normalization of the chosen responses only, peaks early and then catastrophically overfits: it reaches 63.7% within 0.1 epochs and drops to 57.9%, consistent with reliance on generative fingerprints that do not transfer. At smaller scales, E0$^+$ (1M) and E0 (77K) overfit more slowly, to a similar depth, due to reduced exposure. (ii) The regularized E3 pipeline improves with scale: peak ODev improves from 66.0% (77K) to 67.4% (1M) to 67.8% (9.3M), with 6.7M and 9.3M showing comparable peaks. Two additional 1M E3 runs, trained at different times with different seeds and

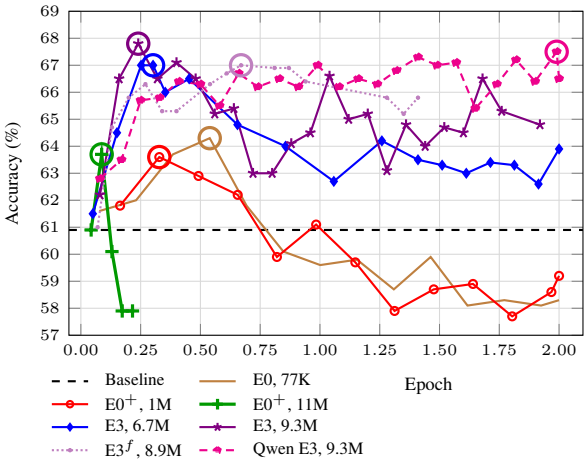


Figure 4: Scaling learning curves on ODev.

	Rows	ODev	OTest	WC	AE	AH
Phi-4	—	60.9	61.9	59.6	80.1	60.2
E0	77K	64.3	65.6	62.0	81.5	62.4
E2	77K	64.3	64.3	60.5	79.9	62.8
E3	77K	66.0	66.0	62.0	81.5	63.8
E0$^+$	1M	63.6	65.4	61.8	82.2	63.8
E2	1M	66.2	67.1	63.8	80.4	**65.4**
E3	1M	67.4	69.2	**65.8**	80.1	65.2
E2	6.7M	66.2	68.4	62.4	78.5	63.8
E3	6.7M	67.0	**69.7**	65.2	80.1	64.2
E0$^+$	11M	63.7	66.1	61.2	80.2	62.8
E3	9.3M	67.8	68.5	65.2	**83.2**	64.4
E3^f	8.9M	67.0	68.7	63.3	82.0	62.4
Qwen	—	63.8	60.9	60.9	81.0	63.6
E2	1M	66.0	64.7	62.4	81.4	64.2
E3	6.7M	**67.9**	66.1	**63.8**	80.7	64.4
E3	9.3M	**67.5**	**67.6**	63.4	**82.1**	**64.8**

Table 2: Scaling and generalization results.

node layouts and each with its own ODev-selected checkpoint, score 68.97% and 68.93% on OTest (ODev 66.5% and 66.2%), a 0.04-point spread; agreement despite the changed seed and setting makes this a stronger stability check than a seed-only replicate. Other training runs are single runs because of their cost. (iii) At 9.3M, training exhibits oscillations of several points across epochs, likely due to noise from dialogs involving external content. Filtering such cases (Appendix B.1) yields E3^f (8.9M), with noticeably more stable dynamics. (iv) Qwen-2.5-14B trained on the same E3 data achieves comparable accuracy with smoother curves, suggesting greater robustness to residual noise in the preference signal. This behavior is consistent with our theoretical analysis of fingerprint reduction and semantic signal preservation (Section 4.1).

Test set evaluation. Table 2 reports peak-checkpoint results on OTest and public benchmarks. Phi4-E3 on OTest improves from 66.0% (77K) to 69.2% (1M) to 69.7% (6.7M), while E0 (raw) and E0$^+$ (chosen-normalized) remain flat at 65.4–66.1% from 77K to 11M despite a $\sim$140$\times$ data increase. Paraphrasing consistently improves performance at matched scale: +1.7pp (77K), +2.1pp (1M), +1.3pp (6.7M) (E3 vs. E2). Public prompt sets confirm generalization under our rubric: WildChat +6.2pp, AlpacaEval +3.1pp, Arena-Hard +5.0pp. Cross-model transfer also scales: Phi3-E$k\rightarrow$Qwen improves from +3.8pp (E2, 1M) to +6.7pp (E3, 9.3M) on OTest. Gains are largest from 77K to 1M and saturate beyond 6.7M, though 9.3M yields the best average across all test sets. Qwen achieves comparable results despite using preference data constructed with Phi-3-mini rejections, while using native rejections would likely further improve performance, consistent with Figure 3. Finally, the full pipeline is reproduced on the public WildChat GPT-4 partition (208,187 source dialogs; 101,151 E3 pairs): the E0$\rightarrow$E3 hierarchy recurs, and WildChat E3 improves Phi-4 from 61.9% to 68.4% on OTest (+6.5pp). We plan to release the WildChat splits, pair IDs, preprocessing code, prompts, and training configuration (Appendix E.4).

6 Conclusions

We presented a fully unsupervised framework for preference alignment of LLMs using production dialog logs. By pairing strong-model responses with weaker same-family generations, we construct millions of preference pairs without annotation or LLM judges. Building on prior observations of multi-model source shortcuts, we characterize the *generative fingerprint* problem in real log-derived pairs—preference-alignment methods, including DPO, can learn model-specific artifacts rather than semantic quality—and address it with a regularization pipeline combining normalization, length balancing, and both-side paraphrasing. We show that, under explicit assumptions, shared transformations reduce source information while preserving the semantic preference gap. Experiments show three findings: (i) the regularized pipeline enables DPO to scale with data, improving from 77K to 9.3M while raw DPO fails at all scales; (ii) the learned signal generalizes, with gains of +6.2pp (WildChat), +3.1pp (AlpacaEval), and +5.0pp (Arena-Hard) on public prompts under our binary rubric; (iii) the approach transfers across model families, with Qwen-2.5 improving by +6.7pp using Phi-family data.

Models reach peak performance within 20–30% of the available data (roughly 2–3M pairs), suggesting redundancy in production logs; data selection or curriculum strategies are a natural next step.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024a.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024b.

M. G. Azar, Mark Rowland, Bilal Piot, D. Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

Yuntao Bai et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Ganqu Cui et al. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. RLHF workflow: From reward modeling to online RLHF. *Trans. Mach. Learn. Res.*, 2024, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Steven Y. Feng et al. A survey of data augmentation approaches for nlp. *Findings of ACL*, 2021.

Robert Geirhos et al. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020.

Suchin Gururangan et al. Annotation artifacts in natural language inference data. In *NAACL*, 2018.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning Workshop*, 2015.

Divyansh Kaushik et al. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*, 2020.

Dahyun Kim, Yungi Kim, Wonho Song, Hyeonwoo Kim, Yunsu Kim, Sanghoon Kim, and Chanjun Park. sdpo: Don't use your data all at once. *arXiv preprint arXiv:2403.19270*, 2024.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Krishna Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6448–6458. PMLR, 2020.

Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *Annals of Statistics*, 34(5): 2326–2366, 2006.

Eric Mitchell. A note on DPO with noisy preferences and relationship to IPO. Unpublished note, 2023. Available at `https://ericmitchell.ai/cdpo.pdf`.

Eric Mitchell et al. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *ICML*, 2023.

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL `https://arxiv.org/abs/2508.10925`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, pp. 27730–27744, 2022.

Adam Poliak et al. Hypothesis only baselines in natural language inference. In *SEM*, 2018.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

Wentao Shi, Mengqi Yuan, Junkang Wu, Qifan Wang, and Fuli Feng. Direct multi-turn preference optimization for language agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2312–2324, 2024. doi: 10.18653/v1/2024.emnlp-main.138.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826. IEEE, 2016.

Alexandre B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *Annals of Statistics*, 32(1):135–166, 2004.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 8384–8395. Association for Computational Linguistics, 2020.

Yifan Wang, Runjin Chen, Bolian Li, David Cho, Yihe Deng, Ruqi Zhang, Tianlong Chen, Zhangyang Wang, Ananth Grama, and Junyuan Hong. More is less: The pitfalls of multi-model synthetic preference data in DPO safety alignment. *arXiv preprint arXiv:2504.02193*, 2025.

Lingxi Xie, Jingdong Wang, Zhen Wei, Meng Wang, and Qi Tian. Disturblabel: Regularizing CNN on the loss layer. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4753–4762. IEEE, 2016.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Haorui Wang, Zhen Qin, Feng Han, Jialu Liu, Simon Baumgartner, Michael Bendersky, and Chao Zhang. PLaD: Preference-based large language model distillation with pseudo-preference pairs. *arXiv preprint arXiv:2406.02886*, 2024.

Zhengze Zhang, Shiqi Wang, Yiqun Shen, Simin Guo, Dahua Lin, Xiaoliang Wang, Cam-Tu Nguyen, and Fei Tan. daDPO: Distribution-aware DPO for distilling conversational abilities. *arXiv preprint arXiv:2506.15717*, 2025.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*, 2024.

A Why Cross-Model Preference Construction Induces a Generative Fingerprint Shortcut

We formalize why unsupervised preference construction from heterogeneous models introduces a shortcut signal that can dominate preference optimization, including DPO.

Setup. Let x be a dialog context, $Z \in \mathcal{Z}$ a discrete variable denoting the source model, and $Y \sim p(y \mid x, Z)$ the generated response. We consider binary preferences over pairs (y^+, y^-) with label $\ell \in \{0, 1\}$ indicating the constructed training preference assignment.

We decompose the response into: (i) semantic quality $Q(x, y)$, capturing correctness and relevance with respect to x, and (ii) a generative fingerprint $M(y)$, capturing model-specific stylistic and distributional signatures, including lexical choice, formatting, punctuation, and local discourse patterns.

The observed preference signal therefore combines both semantic and fingerprint components.

Standard RLHF / DPO. In standard RLHF and DPO settings, preference labels are either manually annotated or generated from comparisons that are approximately independent of the source model. Consequently, $I(Z; \ell \mid x) \approx 0$, meaning that source-model identity provides little information about the preference label. In this regime, generative fingerprint features are weakly correlated with supervision, forcing the model to rely primarily on semantic quality differences.

Unsupervised cross-model construction. In our setting, preference labels are deterministically induced by source-model assignment: responses generated by the stronger model are labeled chosen, while responses from the weaker model are labeled rejected. Thus, $I(Z; \ell \mid x)$ is very large.

Since the generative fingerprint $M(y)$ is highly predictive of the source model Z, it becomes indirectly predictive of the preference label: $I(M(Y); \ell \mid x) \gg 0$. As a result, preference-optimization methods, including DPO, can minimize their losses by exploiting fingerprint features instead of learning semantic quality: $\Delta_{obs} = \underbrace{\Delta Q}_{\text{semantic}} + \underbrace{\Delta M}_{\text{fingerprint}}$, where $\Delta Q = Q(x, y^+) - Q(x, y^-)$ and ΔM captures separability induced by generative fingerprints.

When $|\Delta M| \gg |\Delta Q|$, optimization is dominated by shortcut features rather than semantic preference, yielding high training accuracy with poor generalization.

Group-level fingerprints. This phenomenon persists even when using multiple strong and weak models. Let $Z \in \{S, W\}$ denote groups of strong and weak generators rather than individual models. Since generative fingerprints remain correlated with capability groups, $I(Z; \ell \mid x)$ remains large, and shortcut learning still emerges from group-level stylistic patterns rather than semantic quality.

Connection to semantic-preserving transformations. Let P be a semantic-preserving transformation applied independently of Z, and define $\tilde{Y} = P(Y)$. By the data processing inequality, $I(\tilde{Y}; Z \mid x) \leq I(Y; Z \mid x)$.

In practice, transformations such as normalization and paraphrasing reduce local stylistic asymmetries, making the transformed generative fingerprint $M(\tilde{Y})$ less predictive of the source model. Combined with bounded semantic-quality drift, $|Q(x, P(y)) - Q(x, y)| \leq \epsilon_Q$, this yields an asymmetric effect: fingerprint signals are contracted while the semantic preference gap is preserved up to $2\epsilon_Q$. Consequently, semantic-preserving transformations improve the effective signal-to-noise ratio for DPO learning.

Summary. The central issue is not the existence of generative fingerprints themselves, but their alignment with the constructed preference labels in unsupervised cross-model preference learning. Our regularization pipeline weakens this alignment, forcing preference-optimization methods, including DPO, to rely more strongly on semantic quality rather than source-model shortcuts.

B Data Curation Pipeline

Raw dialog logs are noisy and require substantial processing before they can serve as DPO training data. We develop a multi-stage curation pipeline implemented as fast NLP processors that can handle millions of conversations efficiently. The pipeline consists of three main stages: quality filtering (Section B.1), style normalization (Section B.2, Algorithm 1), and length balancing (Section B.3, Algorithm 2). Both-side paraphrasing (Algorithm 3) is described separately in Section B.4.

B.1 Quality Filtering

Dialog logs contain numerous artifacts from production data collection systems. Our filtering pipeline addresses four major noise sources:

Table 3: Surface feature analysis of chosen (source model) vs. rejected (Phi-3-mini) responses in E0$^+$ data (chosen normalized, rejected raw; $N = 5{,}000$ pairs).

Feature	Chosen	Rejected
Has any markdown	0.4–0.5%	38–46%
Bold text (`**...**`)	0.0–0.2%	15–26%
Headers (`#`)	0.2–0.3%	14–19%
Bullet/numbered lists	0.1–0.2%	11–18%
Filler openers ("Sure!", etc.)	0.1–0.3%	19–21%
Newline density (per 1000 chars)	0.4–0.7	54–71

Turn alternation validation. Well-formed conversations alternate between user and assistant turns. Production logs frequently contain malformed sequences: consecutive user messages (from retry behavior), consecutive assistant messages (from streaming failures), or missing turns. We validate turn alternation and discard conversations that do not follow the expected `user`→`assistant`→`user`→ $\cdots$ pattern.

Deduplication. Data collection pipelines often produce duplicate turns within conversations—identical user queries or assistant responses appearing consecutively. These arise from logging redundancies, page reloads, or retry mechanisms. We detect and collapse consecutive duplicate turns, preserving one instance of each unique message.

Truncation filtering. A subset of conversations contain truncated assistant responses—answers cut off mid-sentence due to token limits, network interruptions, or collection artifacts. We detect truncated responses using heuristic signals (missing sentence-final punctuation, cut-off code blocks, incomplete lists) and remove affected conversations, as truncated chosen responses would provide misleading preference signal.

External content filtering. Approximately 9.5% of conversations reference uploaded files, images, or documents that the user shared with the original assistant (e.g., "analyze this PDF," "what does this screenshot show?"). The chosen response correctly addresses file content that a text-only model can never observe. Training on such pairs teaches the model to prefer confident hallucination about invisible content. We detect these cases by scanning user messages for file-extension references (e.g., `.pdf`, `.png`, `.xlsx`), upload/attachment language patterns ("I uploaded," "see the attached," "in the image"), and ChatGPT-style file markers. To avoid false positives from code snippets that mention filenames, we apply a context window check: if the match appears within 120 characters of programming-language indicators (`import`, `require`, HTML tags, URL patterns), it is treated as a code reference and retained. All flagged rows are removed from the training set.

Because the experiments use separately sampled and processed source pools, there is no single pipeline-wide retention rate. The canonical OTrain ablation contains 77,130 E3 pairs; the filtered WildChat GPT-4 source contains 208,187 dialogs and yields 169,950 E0 or 101,151 E3 pairs; and the OTrain scaling artifact is an exact 1,000,000-pair E3 sample.

B.2 Style Normalization Rules

A critical discovery in our early experiments was that DPO models trained on cross-model preference data learn to classify responses by *formatting style* rather than *content quality*. The chosen responses (from a powerful LLM, here already normalized as in E0$^+$) and rejected responses (from Phi-3-mini) have systematically different surface characteristics (Table 3).

We apply aggressive style normalization to *both* chosen and rejected responses:

- Strip markdown headers, bold, italic formatting
- Remove bullet and numbered list markers
- Preserve code blocks verbatim (extracted and re-inserted); remove inline code formatting
- Strip emoji characters
- Remove filler opener phrases ("Sure!", "Of course!", "Absolutely!", etc.)
- Collapse multiple consecutive newlines to single newlines

This normalization is applied symmetrically to both sides, reducing the formatting shortcut. Critically, in our early experiments, we discovered that chosen responses were inadvertently normalized during

Algorithm 1 Style Normalization

Require: DPO pair $(x, y_{\text{chosen}}, y_{\text{rejected}})$
Ensure: Normalized pair $(x, \hat{y}_{\text{chosen}}, \hat{y}_{\text{rejected}})$
function NORMALIZESTYLE(y):
  Extract and store code blocks (fenced ``` ```...``` ```)
  Strip markdown: bold (**), italic (*), headers (#), lists, blockquotes, horizontal rules
  Remove filler openers via regex: "Sure!", "Absolutely!", "Great question!", "Happy to help!", ...
  Apply `clean-text`: fix Unicode, remove emoji, flatten newlines
  Collapse multiple spaces to single space
  Re-insert stored code blocks
  return stripped text
$\hat{y}_{\text{chosen}} \leftarrow$ NORMALIZESTYLE(last assistant message in y_{chosen})
$\hat{y}_{\text{rejected}} \leftarrow$ NORMALIZESTYLE(last assistant message in y_{rejected})
if $\hat{y}_{\text{chosen}} = \emptyset$ or $\hat{y}_{\text{rejected}} = \emptyset$ **then** discard pair

Algorithm 2 Length Balancing

Require: Dataset $\mathcal{D} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$
Ensure: Balanced dataset $\mathcal{D}'$ with mean($|y^+|$) $\approx$ mean($|y^-|$)
for $i = 1, \ldots, N$ **do**
  $\delta_i \leftarrow$ wordcount(y_i^+) $-$ wordcount(y_i^-)
end for
$\Delta \leftarrow \sum_i \delta_i$ {total length imbalance}
if $\Delta > 0$ **then**
  Sort pairs with $\delta_i > 0$ in descending order of δ_i
  Greedily remove pairs from this sorted list until $\Delta \leq 0$
else if $\Delta < 0$ **then**
  Sort pairs with $\delta_i < 0$ in ascending order of δ_i
  Greedily remove pairs from this sorted list until $\Delta \geq 0$
end if
$\mathcal{D}' \leftarrow$ remaining pairs, shuffled
return $\mathcal{D}'$

quality filtering while rejected responses were left raw—creating a *massive* asymmetric style signal. Correcting this to symmetric normalization was necessary for the pipeline ablations to isolate the effect of each step.

The full procedure is given in Algorithm 1. We use the `clean-text` Python library for Unicode normalization and emoji removal, with a regex fallback when unavailable.

B.3 LENGTH BALANCING

Length differences between chosen and rejected responses can act as a proxy for preference—DPO may learn that longer (or shorter) responses are always preferred, rather than attending to content. We balance the dataset by removing extreme-length-difference pairs until the mean word count of chosen and rejected responses are approximately equal.

This greedy algorithm maximizes data retention: it removes only the most extreme outliers (pairs where one side is much longer) until the aggregate imbalance reaches zero. In practice, this removes 5–15% of pairs depending on the data source; together with the quality filters of Appendix B.1 and the removal of dialogs exceeding the rejection model's context length, total retention from E0 to E3 can be lower (e.g., 169,950→101,151 on WildChat).

B.4 SEMANTIC-PRESERVING REGULARIZATION OPERATOR

To reduce generative fingerprints, we paraphrase *both* chosen and rejected responses using the same model with the same prompt. Paraphrasing is applied sentence by sentence: each response is split into sentences, every sentence is paraphrased independently with the same model and prompt, and the outputs are reassembled in their original order. Specifically, we use Phi-4 (14B, temperature 0.1, top-p 0.95) with the following per-sentence instruction:

Algorithm 3 Both-Side Paraphrasing (Both-Reg)

Require: Dataset $\mathcal{D} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$, paraphrase model P (Phi-4 14B, T=0.1, top-p=0.95)
Ensure: Paraphrased dataset $\tilde{\mathcal{D}}$
Define per-sentence prompt template t:
  `Paraphrase the following sentence. Keep the same meaning,`
  `structure, and length. Rephrase it but do not`
  `add, remove, or reorder any information.`
  `SENTENCE: {sentence}`
  `PARAPHRASED SENTENCE:`
SentenceSplit: rule-based sentence-boundary segmentation on terminal punctuation; JoinInOrder: concatenation with single spaces in the original order
for each pair $(x_i, y_i^+, y_i^-) \in \mathcal{D}$ **do**
  $a^+ \leftarrow$ last assistant message content from y_i^+
  $a^- \leftarrow$ last assistant message content from y_i^-
  if $|a^+| > 12{,}000$ chars or $|a^-| > 12{,}000$ chars **then**
    Skip pair (avoid context overflow)
  else
    for $a \in \{a^+, a^-\}$ **do**
      $(s_1, \ldots, s_m) \leftarrow \mathrm{SentenceSplit}(a)$
      for $j = 1, \ldots, m$ **do**
        $\tilde{s}_j \leftarrow P(t(s_j))$ {paraphrase one sentence with the same P and t}
      end for
      $\tilde{a} \leftarrow \mathrm{JoinInOrder}(\tilde{s}_1, \ldots, \tilde{s}_m)$
    end for
    Replace last assistant content in y_i^+ with $\tilde{a}^+$
    Replace last assistant content in y_i^- with $\tilde{a}^-$
  end if
end for
return $\tilde{\mathcal{D}}$

> *"Paraphrase the following sentence. Keep the same meaning, structure, and length. Rephrase it but do not add, remove, or reorder any information."*

By processing both sides through the same generative bottleneck, model-specific generative fingerprints—vocabulary preferences, sentence rhythm, transition patterns—are reduced. After paraphrasing, both chosen and rejected responses are mapped through the same Phi-4 realization channel, reducing source-model-specific stylistic differences. The remaining differences are therefore more strongly tied to content quality, such as correctness, completeness, and reasoning depth.

The paraphrasing prompt is carefully designed to preserve meaning while allowing surface-level rephrasing. The instruction to "keep the same structure and length" prevents the paraphraser from summarizing or expanding, which would alter the quality signal. Responses exceeding 12,000 characters are skipped; they are rare and would dominate paraphrasing cost.

The full procedure is given in Algorithm 3. Paraphrasing is parallelized across 8 GPUs using vLLM with data sharding (each GPU processes $N/8$ rows independently).

B.4.1 Why Both Sides Must Be Paraphrased

We tested three paraphrasing strategies:

Chosen-only paraphrasing: Paraphrase only the chosen (source model) responses, leaving rejected (Phi-3-mini) responses in their original form. This reduces the source-model fingerprint but *introduces* a new asymmetry: chosen responses now carry Phi-4's fingerprint while rejected responses retain Phi-3-mini's.

Both-side paraphrasing (Both-Reg): Paraphrase both chosen and rejected with the same model and prompt. Both sides are rewritten through the same Phi-4 paraphrasing channel, making source-model fingerprint classification substantially harder.

Rejected regeneration: Instead of paraphrasing, regenerate the chosen response from scratch using the weaker model. This obviously degrades chosen quality, undermining the preference

signal. We measured this degradation with an external judge and confirmed that regenerated chosen responses are significantly worse than the originals—defeating the purpose of using a stronger model's outputs as chosen examples.

C Evaluation Details

C.1 Test and Dev Set construction

ODev (2,000 items) is a simple random sample of held-out conversations with no ID overlap with any training set. For OTest (3,000 items) we applied the following filters and methods:

1. **Web-access filtering**: conversations referencing uploaded files, images, or URLs were removed, since the model cannot access external resources at evaluation time. A regex-based classifier with code-context disambiguation (to avoid false positives on programming discussions) removed $\sim$10% of candidates.
2. **Stratified sampling**: 1,000 single-turn, 1,000 two-turn, and 1,000 three-or-more-turn conversations were sampled with a fixed random seed, ensuring balanced representation across dialog complexity levels.
3. **Decontamination**: overlap with all training data was verified to be zero using exact-match ID checks.

C.2 Baseline results

Table 4 reports the baseline results. We use a super-AI (SA) system prompt for all evaluations. It has negligible effect on Phi-4 (<0.2pp) but provides a +1.0–2.4pp gain for Qwen-2.5. Notably, Qwen-2.5 scores higher than Phi-4 on ODev (63.8% vs. 60.9%) but lower on OTest (60.9% vs. 61.9%), suggesting some benchmark sensitivity.

Table 4: Baseline accuracy (no DPO training) on both evaluation sets. SA = evaluation system prompt; NP = no system prompt.

Model	Prompt	ODev (2K)	OTest (3K)
Phi-4 14B	NP	60.8%	62.0%
Phi-4 14B	SA	60.9%	61.9%
Qwen-2.5 14B	NP	62.8%	58.5%
Qwen-2.5 14B	SA	63.8%	60.9%

C.3 Experimental Protocol

All ablation experiments share the following configuration to ensure comparability:

Training.

- Base model: Phi-4 (14B)
- Rejection model: Phi-3-mini-128k-instruct (3.8B)
- DPO $\beta = 0.1$, cosine learning rate schedule
- Learning rate: 5×10^{-8}
- Per-device batch size: 2, gradient accumulation: 1
- Nodes: 2 (16 GPUs), effective batch size: 32
- Epochs: 2
- Max sequence length: 4,096 tokens
- Checkpoints saved every 500 steps (all retained)
- DeepSpeed ZeRO Stage 2
- No system prompt in training data
- No length-decay regularization ($\alpha = 0$)

Evaluation.

- Dev set: 2,000-item (ODev)
- Generation: vLLM with tensor parallelism 2, temperature 0.1
- SA system prompt during evaluation: "You are a super AI assistant. Your task is to provide accurate, and complete answers to user queries during a chat."
- Judge: Opus 4.6 (`us.anthropic.claude-opus-4-6-v1`) via AWS Bedrock, 20 parallel workers
- All checkpoints evaluated to capture the full training curve

C.4 Evaluation Protocol

All experiments are evaluated using an LLM-as-judge approach with Claude Opus 4.6 (`us.anthropic.claude-opus-4-6-v1`) via AWS Bedrock. The judge operates as a binary classifier: given a conversation history and a generated answer, it determines whether the answer is correct and appropriate.

Judge system prompt. The following prompt is provided as the system message:

```
You are an AI Judge, expert in evaluating the quality and
correctness of AI-generated responses.
You will be provided with:  1.  A conversation
history/context that includes the user's question  2.  A
generated answer that needs to be evaluated
Your task is to determine if the generated answer is
correct, appropriate, and helpful given the context and
the user's question.
You must return a JSON object with exactly two key-value
pairs:  1.  "label":  "yes" or "no"  2.  "explanation":  A
clear and concise justification
Evaluation criteria:  1.  Factual correctness  2.  Relevance
to the user's question  3.  Completeness of information  4.
Appropriateness of tone and style 5.  Coherence and structure
```

User prompt template. Each evaluation instance is formatted as:

```
Context and conversation history: {chat_history}
Generated Answer to evaluate: {generated_answer}
Please evaluate if the generated answer is correct and
appropriate given the context and the user's question.
Provide your assessment in JSON format.
```

Generation system prompt. During answer generation with trained models, we use the following system prompt for consistency across all experiments:

```
You are a super AI assistant.  Your task is to provide
accurate, and complete answers to user queries during a
chat.
```

Settings. The judge uses temperature 0, max 512 output tokens, and 20 parallel workers. Generation uses temperature 0.1 with vLLM (8 GPUs, tensor parallelism 2). Accuracy is computed as the fraction of test items receiving `"label": "yes"` from the judge.

Judge calibration audit. We calibrated the Opus rubric through human annotation of 20 samples of 50 answers, typically drawn from different model checkpoints, primarily on ODev and several times on OTest. For each sample, the annotators manually inspected at least 20 answers with the complete dialog context. Annotator–judge agreement was approximately 70%; about 80% of disagreements were genuinely ambiguous cases for which either label was defensible, leaving clear disagreement in approximately 6% of inspected answers.

D Additional Results

D.1 DPO Training Reward Accuracy: Indirect Evidence of Fingerprint Reduction

Figure 5 plots the implicit DPO reward accuracy, i.e., the fraction of training examples where the model's implicit reward assigns higher value to the chosen response, across the three pipeline variants

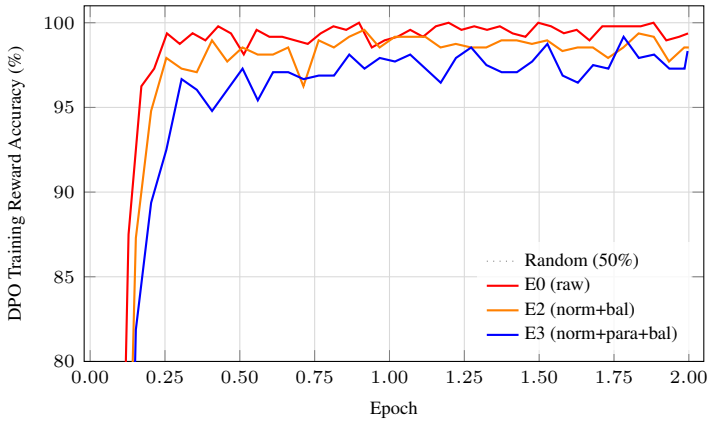


Figure 5: DPO implicit reward accuracy on the training set (77K rows, Phi-4, 2 epochs). E0 (raw) converges fastest and saturates highest (∼99.5%), consistent with exploiting generative fingerprints. Each regularization step—normalization and length balancing (E2) and paraphrasing (E3)—progressively delays convergence and lowers the saturation ceiling, consistent with fingerprint reduction pushing the model to learn from content-quality differences rather than surface-level distributional shortcuts; reward accuracy is indirect evidence and does not directly measure fingerprint reliance.

from the 77K Phi-4 ablation. All configurations eventually reach near-perfect training accuracy, but they differ substantially in convergence speed and saturation level. E0 (raw, unregularized) reaches 95% within 0.17 epochs and saturates above 99.5%, consistent with the model rapidly exploiting generative fingerprints—surface-level patterns such as formatting, length, and vocabulary that distinguish chosen from rejected responses without requiring semantic understanding.

Each successive regularization step slows convergence: E2 (norm+bal) reaches 95% at 0.20–0.23 epochs, while E3 (para+norm+bal) requires 0.31 epochs and plateaus ∼2pp lower at 97.6%. This is indirect evidence consistent with paraphrasing contracting the fingerprint signal; it is not a direct source-classifier measurement: once local stylistic cues are reduced, the model must rely more strongly on semantic quality differences, which are harder to fit and yield a lower ceiling on training reward accuracy.

Crucially, E3's slower convergence corresponds to the best generalization on held-out evaluation (Section 5.2), consistent with fingerprint reduction leading the model to learn a more robust semantic preference signal that transfers beyond the training distribution; this association supports, but does not prove, the mechanism.

This distinction is important: reduced source-model separability alone would not imply improved preference learning, since semantic information could also be degraded. Our evidence instead comes from the combination of reduced reward overfitting, stable learning dynamics, and improved held-out generalization under the regularized pipeline.

D.2 Cross-Model Transfer to Phi-3-mini (3.8B)

To test whether cross-model DPO transfer extends to smaller models, we trained Phi-3-mini-128k-instruct (3.8B parameters) using the same three data variants from the Qwen ablation study (Section 5.3), generated with the Qwen-2.5-3B rejection model. Training configuration: 2 nodes (16 B200 GPUs), lr=3e-8, 2 epochs, batch size 32, ZeRO-2.

Unlike the Phi-4 (14B) and Qwen-2.5-14B (14B) models, which both benefit from cross-family data (Section 5.3), Phi-3-mini shows no improvement under any pipeline variant. The E0-77K Phi-3-mini model selected by maximum ODev accuracy reaches 35.3%, which is within noise of the 34.4% baseline, and all curves trend *downward* with continued training. The X3 experiment (dashed purple) disfavors the cross-family hypothesis: even when trained on same-family data (Phi-4 E3 pipeline with Phi-3-mini rejections), Phi-3-mini degrades similarly, peaking at 34.6% and declining to ∼31%. This is consistent with a capacity-related failure: at 3.8B the model does not extract a useful preference signal from DPO data at this scale, regardless of whether the rejection model belongs to the same or a different model family. With a single small target, however, we cannot identify the capacity threshold or exclude optimization- and model-specific causes.

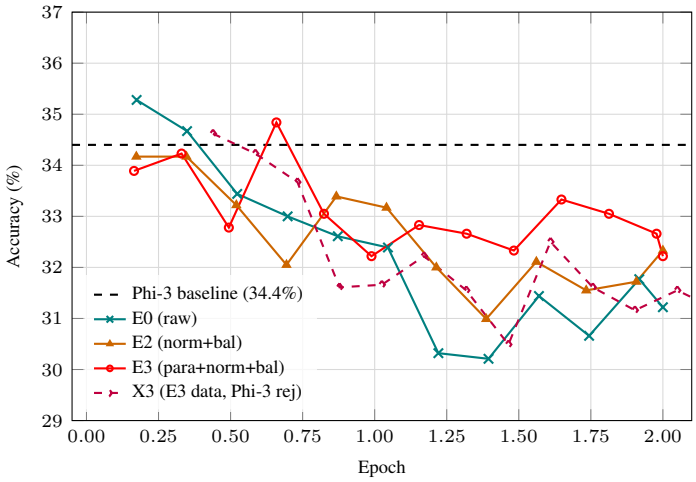


Figure 6: Cross-model transfer to Phi-3-mini (3.8B). Solid lines: Qwen-2.5-3B rejection data (cross-family); dashed: Phi-3-mini rejection data via the Phi-4 E3 pipeline (same-family, X3). All four variants fail to improve over the Phi-3 baseline (34.4%), with accuracy degrading steadily over training. E0 (raw, cross-family) shows the worst collapse, dropping to 30.2% by epoch 1.4. X3 (same-family Phi-3 rejections, E3 pipeline) follows a similar trajectory: starting near baseline (34.6%) and declining to $\sim$31%, consistent with a capacity-related failure rather than a cross-family artifact, although one target size does not identify the threshold. This contrasts sharply with the same data applied to Qwen-2.5-14B (Figure 3a), where E3 reaches 66.0%.

E Judge Robustness and Controls

This appendix evaluates direct source separability, cross-judge robustness, paraphraser dependence, and public-data transfer with additional target policies; each experiment isolates one variable left open by a single-judge, single-paraphraser protocol.

E.1 Direct Source-Separability Probe and Mechanism

Probe. We train a pairwise TF–IDF logistic classifier to predict which member of a (y^+, y^-) pair is the logged answer, with train/test split by dialog so that no dialog contributes to both sides. Trained and tested on E0 pairs it reaches 97.2% on OTrain and 90.2% on WildChat. Trained and tested on the E3 versions of the identical pair IDs it drops to 94.3% and 81.4%, while the Opus-judged quality gap remains large (40.4→37.0 on OTrain, 43.8→35.6 on WildChat; Table 1). The residual cue is diffuse rather than concentrated: in a separate run of the probe (different split; 89.8% baseline), removing the 252 strongest TF–IDF dimensions changes pairwise accuracy only to 86.8%, and removing the top and bottom 1,000 dimensions only to 77.0%.

Why residual accuracy stays high. Chosen answers are judged correct far more often than rejected ones (73% vs. 36%), so a detector of quality alone already predicts the source. High post-E3 accuracy therefore does not measure the remaining fingerprint; a fully quality-controlled probe would need source-orthogonal labels, which do not exist here. What can be measured is the contraction, and on identical pairs a 3–9 point contraction coincides with the change from early saturation (E0) to continued improvement (E3) in Figure 2.

Why not generic regularization. Four facts leave little room for a general-regularization reading of the E3 gain. E3 adds no data and lowers label quality (gap 40.4→37.0, Table 1). E1/E2 normalize and balance the same pairs yet plateau like E0 (Figure 2). SFT on the chosen answers loses (Figure 2b). The gain is paraphraser-independent (Appendix E.3). The one operation unique to E3 is rewriting both answers through the same model so that they no longer differ in style, only in content; removing that style difference is what we mean by reducing the fingerprint; the association is consistent with this mechanism, although it does not prove it (Appendix D.1).

E.2 Cross-Judge Robustness

We re-evaluate the Phi-4 model trained on E3-77K and selected by maximum ODev accuracy (*best-Phi4-E3-77K*; throughout this appendix, "best" means highest ODev accuracy) and the

Judge	Base	E3-77K	Δ	p
Opus 4.6	61.9	66.0	+4.1	$< .001$
gpt-oss-120B	81.4	83.2	+1.8	.0074
Qwen3-Coder-480B	86.2	87.8	+1.6	.0070

Table 5: Prompt-matched cross-judge robustness on OTest (accuracy in %). All three judges find a significant paired improvement for the same ODev-selected E3-77K model. The valid paired samples are 2,963 for gpt-oss and 3,000 for Qwen3-Coder; Opus significance is reported conservatively as $p < .001$.

Phi-4 baseline on the same OTest answer sets with three model families. Opus 4.6, gpt-oss-120B (OpenAI et al., 2025), and Qwen3-Coder-480B (Yang et al., 2025) all receive the exact judge system and user prompts of Appendix C.4. We test each base–E3 difference with a paired McNemar test over per-example binary labels; no judge output is used for model training or selection. Table 5 reports the resulting accuracies and paired significance tests.

Judge calibration changes the absolute acceptance rate substantially, from approximately 62–66% under Opus to 81–88% under the two open-weight judges. Nevertheless, every judge assigns E3 a positive gain and rejects the paired null at the 1% level. Thus, the E3-77K improvement is not specific to the Claude judge, although its measured magnitude remains judge-dependent. Raw cross-judge correlation is less informative than this paired test because the judges use different absolute thresholds.

E.3 Target-Independent Paraphraser Control

To test whether the result depends on Phi-4 serving as both paraphraser and target, we compare two matched E3-77K training conditions while holding fixed all 77,130 pair IDs, their order, prompts, normalization, the Phi-4 target, and the DPO recipe (2 epochs, learning rate 5×10^{-8}, β=0.1, ZeRO-2, and maximum length 4096). The Phi-paraphrased condition uses the batch-80 Phi-paraphrased run's checkpoint 1560, selected by maximum full-ODev accuracy. The Qwen-paraphrased condition uses checkpoint 1040 of the matched Qwen-2.5-14B-paraphrased run (*selected-Phi4-E3-77K-QwenPara*), selected on ODev-eco; ODev-eco is a fixed 500-item subset used only for economical screening, without inspecting OTest. Both paraphraser conditions use an effective batch of 80 preference pairs per optimizer update across all GPUs. Because this differs from the batch-32 77K ablation recipe in Section 5.1, only comparisons within this matched control are interpreted. Both selected models and the base model are evaluated on the same 3,000 OTest items. Table 6 reports the results under the three judges of Appendix E.2.

Judge	Base	Phi para.	Qwen para.	Qwen–base Δ (p)	Qwen–Phi Δ (p)
Opus 4.6	62.2	65.6	65.3	+3.1 (p=1.1×10^{-5})	−0.2 (p=.75)
gpt-oss-120B	81.2	82.3	83.1	+1.9 (p=.0028)	+0.8 (p=.17)
Qwen3-Coder-480B	84.5	87.4	86.9	+2.4 (p=4.9×10^{-5})	−0.4 (p=.41)

Table 6: Controlled paraphraser swap on OTest. Deltas and exact paired McNemar p-values compare Qwen-paraphrased E3 with the base and Phi-paraphrased E3 outputs on identical items. Scores are rounded to one decimal place; deltas are computed from unrounded scores.

Qwen paraphrasing improves significantly over the base under every judge ($p<.003$), while no judge distinguishes Qwen from Phi paraphrasing ($p\geq.17$). Within this matched control, the E3 gain therefore does not require the target model to be its own paraphraser, and the data weigh against Phi-specific self-style distillation; a null difference is evidence against, not a proof of absence. The base and Phi-paraphrased answers in Table 6 come from a separate generation run, so their absolute rates differ slightly from those in Table 5; all comparisons within each table are on identical items.

E.4 Public WildChat Replication and Larger Targets

To test whether the effect depends on the proprietary log distribution, we run the full pipeline on the public WildChat GPT-4 partition (208,187 source dialogs, yielding 169,950 E0 and 101,151

E3 pairs). The E0→E3 hierarchy of Section 5.2 recurs on this corpus, and WildChat E3 improves Phi-4 from 61.9% to 68.4% on OTest (+6.5pp), a corpus with different users, a different source model, and a different collection process. This argues against a production-specific abandonment or external-content explanation of the gain. The E0 controls are identity-matched, not merely row-count matched: E0-101K holds the raw versions of exactly the same 101,151 pair IDs as E3, and E0-170K keeps all 169,950 raw pairs. Both plateau earlier and reach 66.2% and 66.6% on OTest against 68.4% for E3, so neither pair selection nor data volume explains the gain. On these identity-matched pairs the pairwise TF–IDF source classifier of Section 5.2 drops from 90.2% (E0) to 81.4% (E3).

We also train additional target policies, each with ODev-selected checkpoints evaluated by Opus 4.6 on OTest and reported as deltas over the corresponding base model. (i) A Qwen-only pipeline on public data: Qwen-2.5-14B trained on the WildChat-208K partition with Qwen-2.5-3B rejections and a Qwen-2.5-14B paraphraser improves from 60.9% to 65.2% on OTest (+4.3pp). (ii) Larger targets trained on Phi-paraphrased OTrain-1M E3 data: Qwen3-32B improves from 76.8% to 79.6% (+2.8pp) and Qwen-2.5-72B from 69.3% to 75.8% (+6.5pp). (iii) Qwen3-14B improves by +1.9pp, which is within the ±1.7pp evaluation uncertainty and is reported only as marginal. (iv) Conversely, the same-family Phi-3-mini target does not improve under any variant (Appendix D.2). Family matching between paraphraser and target is therefore neither necessary nor sufficient; target capacity and the preserved semantic gap are consistent with the pattern, though the 3.8B result alone does not establish a capacity threshold.

We plan to release the WildChat splits, pair IDs, preprocessing code, prompts, and training configuration so that the public replication can be reproduced end to end; until that release, the replication is not independently reproducible.