Label Words as Reward Anchors: Auditing Model-Visible Preference Labels

Anonymous ACL submission

Abstract

Preference optimization methods such as Direct Preference Optimization (DPO) usually treat labels such as *chosen* and *rejected* as formal indicators of a pairwise preference. In practice, these labels are often serialized as natural-language text in prompts, completions, rubrics, chat templates, or preference-record renderings. This paper studies annotation language as a hidden variable in preference construction. We first state a simple invariance result: if label words remain metadata and never enter the model-visible string or loss, relabeling them cannot affect DPO. We then test the model-visible case with an OpenRouter-served Qwen2.5-7B-Instruct audit over 24 near-tie preference pairs, 10 label conditions, and both A/B orders. With no labels, the model selects the designer-preferred response in 72.9% of comparisons. Aligned labels such as *chosen/preferred* or *evidence-grounded/supported* raise this to 100.0%, while conflict labels that attach the positive wrapper to the rejected response reduce gold-choice rates to 18.8–31.2% and induce 41.7–54.2% flips relative to no-label prompts. These results show that exposed annotation wording can steer model preference judgments even when response pairs are fixed, motivating explicit reporting and auditing of label serialization in preference datasets.

1 Introduction

Preference optimization is a central tool for aligning language models with human judgments. RLHF and related methods train from comparisons between responses (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022). Direct Preference Optimization (DPO) simplifies the pipeline by optimizing a policy directly from preferred and dispreferred completions (Rafailov et al., 2023). In the standard mathematical abstraction, each example is a triple (x, y_w, y_l): prompt x, preferred response y_w, and rejected response y_l. The words used by a dataset or interface to name this relation are usually treated as irrelevant metadata.

That abstraction is only safe when the label words remain outside the text consumed by the model. Real preference pipelines are often text pipelines: records are rendered into prompts, chat messages, JSON-like transcripts, rubrics, critique fields, or templated completions. If a response is serialized as "chosen: ..." or "helpful: ...", the label is no longer a formal bit. It is language with semantics. Work on verbalizers and label words has shown that labels can anchor information flow in in-context learning (Wang et al., 2023a); work on LLM-as-judge evaluation has shown that model preferences are sensitive to presentation details (Zheng et al., 2023; Wang et al., 2023b; Deutsch et al., 2023). This motivates a direct audit question: when response pairs are held fixed, can exposed label words change model preference judgments?

We distinguish two regimes. In **metadata-only DPO**, labels are stored in the dataset schema but do not appear in model inputs, completions, reward-model prompts, or loss arguments. In this regime, label wording is provably irrelevant. In **label-exposed preference construction**, labels are serialized into text seen by the model. This regime is common enough to deserve measurement: even if a final DPO implementation strips labels, intermediate self-labeling, model-judge filtering, preference-rationale generation, or dataset rendering can expose annotation language to a model.

Our contribution is an empirical audit of this exposed-label channel in a real instruction model. We construct 24 near-tie preference pairs across factuality, helpfulness, safety, cultural sensitivity, concision, and general response quality. The underlying responses are fixed across conditions. We vary only the annotation wrapper: no labels, aligned labels, random nonword labels, and conflict labels in which the semantically positive wrapper is intentionally attached to the rejected response. We

evaluate Qwen2.5-7B-Instruct (Yang et al., 2024) through OpenRouter (OpenRouter, 2026) with deterministic decoding and both candidate orders.

The result is a strong label-exposure effect. Aligned labels make the model almost perfectly select the designer-preferred response, while conflict labels cause the model to follow the positive wrapper on most trials. This does not show that every DPO run is label-word sensitive. It shows a narrower and actionable point: if preference annotation language is visible to a model during preference construction, it should be treated as part of the experimental condition and reported accordingly.

2 Related Work

Preference optimization and DPO. Pairwise preference learning has roots in Bradley–Terry models (Bradley and Terry, 1952). Modern RLHF pipelines learn from human comparisons and optimize policies using reward models or related objectives (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022). DPO derives a simple objective over preferred and rejected responses without fitting a separate reward model (Rafailov et al., 2023). We do not alter the DPO objective; we audit when the text used to express the preference relation becomes part of model-visible data.

Label words and verbalizers. Prompt-based learning often maps labels to words. Wang et al. (2023a) show that label words can act as anchors for information flow in in-context learning. Our setting moves the concern from inference-time demonstrations to preference-data serialization: words such as *chosen*, *helpful*, *safe*, or *unsupported* can become visible tokens rather than invisible metadata.

Model judges and presentation bias. LLM judges are widely used for scalable evaluation (Zheng et al., 2023), but they can show position bias, verbosity bias, order effects, and fairness failures (Wang et al., 2023b; Deutsch et al., 2023; Doddapaneni et al., 2024). Our audit studies a related but distinct source of presentation sensitivity: natural-language preference labels attached to candidates.

3 Problem Setup

Let x be a prompt, y_w the preferred response, and y_l the rejected response. A standard DPO dataset is

$$D = \{(x_i, y_{w,i}, y_{l,i})\}_{i=1}^{n}. \tag{1}$$

A label wrapper is a natural-language serialization of the preference relation. We write positive and negative wrappers as z^+ and z^-, for example *chosen/preferred* versus *rejected/less preferred*, or *safe/careful* versus *unsafe/risky*.

Metadata-only labels. Labels are metadata if they never enter the arguments of the policy, reference policy, judge model, reward model, batching weights, sampling logic, or loss. In that case, label wording cannot affect DPO.

Proposition 1. Let $D_z = \{(x_i, z_i, y_{w,i}, y_{l,i})\}$ be a preference dataset with labels z_i that are not used in the arguments of π_θ or π_{ref} and do not affect batching, sampling, or weighting. Then the DPO objective is invariant to any relabeling function f applied to z_i.

Proof. For one example, the DPO loss is

$$\mathcal{L}_{\text{DPO}} = -\log \sigma\left(\beta\left[\Delta_\theta - \Delta_{\text{ref}}\right]\right), \tag{2}$$

where

$$\Delta_\theta = \log \pi_\theta(y_w \mid x) - \log \pi_\theta(y_l \mid x), \tag{3}$$

and Δ_{ref} is defined analogously. If z is absent from x, y_w, y_l, sampling weights, and every loss argument, replacing z by $f(z)$ leaves each term unchanged. Summing over examples preserves equality. □

Label-exposed records. In the exposed case, a record can be rendered as

$$c_w = [z^+; y_w], \qquad c_l = [z^-; y_l], \tag{4}$$

where $[a; b]$ denotes textual concatenation. Once labels are inside c_w, c_l, or a judge prompt, relabeling can change model-visible tokens. The empirical question is not whether the DPO formula changes; it is whether the rendered text changes model behavior during preference construction.

4 Model Audit

4.1 Design

We evaluate a deterministic pairwise-judgment audit rather than a synthetic finite-policy simulation. Each item contains a task and two plausible responses. The designer-preferred response is slightly clearer, safer, more complete, or better calibrated than the rejected response. The pairs are

intentionally near-tie rather than trivially right versus wrong, because preference pipelines most need reliable labels near decision boundaries.

The audit varies only wrapper wording. Conditions are:

- no labels;
- aligned neutral labels: *chosen/preferred* vs. *rejected/less preferred*;
- aligned semantic labels: *helpful/useful*, *safe/careful*, or *evidence-grounded/supported* on the preferred response;
- random nonword labels: *zorp/muv* vs. *tiv/blicket*;
- conflict labels, where the positive wrapper is attached to the rejected response and the negative wrapper to the preferred response.

We run all 24 items under all 10 conditions and both A/B orders, giving 480 model calls. The model is `qwen/qwen-2.5-7b-instruct`, served through OpenRouter on May 26, 2026. Decoding uses temperature 0, top-$p = 1$, and a maximum of 80 output tokens. The prompt asks the model to choose the candidate that should be preferred for training based on task fit, factuality, helpfulness, safety, and clarity, and to return JSON with a single A/B choice.

4.2 Metrics

Gold Choice Rate measures selection of the designer-preferred response:

$$\mathrm{GCR}(z) = \frac{1}{N} \sum_i \mathbb{I}[\hat{y}_{i,z} = y_{w,i}]. \tag{5}$$

Positive-Label Following measures selection of the candidate carrying the semantically positive wrapper:

$$\mathrm{PLF}(z) = \frac{1}{N_z} \sum_i \mathbb{I}[\hat{y}_{i,z} = y^{+}_{i,z}], \tag{6}$$

where $y^{+}_{i,z}$ is undefined for the no-label condition. Flip Rate compares a labeled condition against the same item and A/B order with no labels:

$$\mathrm{Flip}(z) = \frac{1}{N} \sum_i \mathbb{I}[\hat{y}_{i,z} \neq \hat{y}_{i,\mathrm{none}}]. \tag{7}$$

We compute bootstrap confidence intervals over items; raw calls and aggregate CSVs are released with the paper source.

5 Results

The no-label condition is intentionally imperfect: the model selects the designer-preferred answer in 72.9% of comparisons. This confirms that the pairs are not trivial. Aligned labels sharply increase preference agreement. Neutral *chosen/preferred* labels and evidence labels both reach 100.0% gold-choice rate; helpfulness and safety labels reach 95.8% and 93.8%, respectively. These aligned labels also flip 20.8–27.1% of decisions relative to no-label prompts, indicating that they do not merely decorate already stable judgments.

Conflict labels reverse the pattern. When *chosen/preferred* is attached to the rejected response, gold-choice rate falls to 18.8%, and the model follows the positive wrapper in 81.2% of comparisons. Helpfulness and safety conflict labels reduce gold-choice rate to 31.2%; evidence conflict labels reduce it to 22.9%. Flip rates range from 41.7% to 54.2% relative to no-label prompts. The effect is not explained by a simple position bias: position-A choice rates range from 47.9% to 60.4% in conflict conditions.

Random labels behave differently. The random condition reduces gold-choice rate to 62.5%, but its positive-label following rate is also 62.5%, with no interpretable semantic direction. This suggests that arbitrary wrapper tokens can perturb judgments, while meaningful wrappers can steer them in predictable directions.

6 Analysis

Label words act as preference evidence. The strongest result is the contrast between aligned and conflict labels. Because response pairs and A/B orders are fixed, the change must come from presentation. The model treats exposed words such as *chosen*, *safe*, and *evidence-grounded* as evidence about which response should be preferred, even though the response text itself is unchanged.

The effect is strongest near decision boundaries. This audit uses near-tie pairs by design. We do not claim that label words override every obvious factual distinction. In early smoke tests on plainly right-versus-wrong pairs, the same model often ignored conflicting labels. The submission-relevant risk is more practical: many preference pairs are subtle, and those are exactly the cases where model-visible annotation wording can become a hidden decision variable.

Condition	Gold choice	Positive-label choice	Flip vs. no label	Position A
none	72.9	–	–	52.1
neutral	100.0	100.0	27.1	50.0
helpfulness	95.8	95.8	22.9	54.2
safety	93.8	93.8	20.8	56.2
evidence	100.0	100.0	27.1	50.0
random	62.5	62.5	18.8	33.3
neutral_conflict	18.8	81.2	54.2	56.2
helpfulness_conflict	31.2	68.8	41.7	56.2
safety_conflict	31.2	68.8	41.7	47.9
evidence_conflict	22.9	77.1	50.0	60.4

Table 1: OpenRouter model audit on fixed near-tie preference pairs. Gold choice is selection of the response marked preferred by the dataset designer. Positive-label choice is selection of the candidate carrying the semantically positive wrapper; in conflict conditions this wrapper is intentionally attached to the rejected response. Flip rate compares each condition against the same item and order with no labels.

Alignment labels can inflate apparent quality. Aligned labels raise gold-choice rates above the no-label baseline. This might look beneficial, but it is not evidence that the model understands the responses better. The conflict ablations show that the same channel can pull decisions toward the wrong candidate. Therefore, label exposure should be reported and controlled rather than silently relied upon.

Implications for DPO datasets. The invariance proposition says that hidden metadata labels are harmless with respect to the DPO objective. The model audit shows that exposed labels are not hidden metadata. If annotation words appear in completions, prompts, judge inputs, rationales, or rendered preference records, they are part of the treatment. Dataset cards and training reports should specify the exact wording, where it appears, and whether label-removed or conflict-label controls were run.

7 Limitations

The experiment audits preference judgments from one OpenRouter-served instruction model, not full DPO fine-tuning. OpenRouter chat APIs do not expose a controlled fine-tuning interface, so the study targets the preference-construction stage rather than policy-update dynamics. The dataset is small, synthetic, and intentionally near-tie; broader work should test larger item pools, more models, model families with different judge behavior, and actual DPO or LoRA training where labels are serialized into completions. The designer-preferred labels are curated by the authors, not collected from human annotators. Finally, conflict labels are diagnostic stress tests, not normal data-cleaning recommendations.

8 Ethics Statement

This work is intended to improve transparency in preference-data construction. It should not be used to smuggle hidden objectives into training data through annotation wording. Safety and cultural labels require special care because they encode normative assumptions. Our safety examples are benign advice scenarios and do not provide operational harmful instructions. Cultural examples are abstract and should not be read as claims about any specific community. The OpenRouter API key used for the run is not stored in the repository or outputs.

9 Conclusion

Preference labels are neutral only when they remain outside the model-visible computation. If annotation words are serialized into the text a model reads, they can function as reward anchors. In a real-model audit with fixed response pairs, aligned labels substantially increased designer-preferred choices, while conflict labels caused large drops and frequent flips toward the positive wrapper. Preference datasets and training reports should therefore disclose exact label wording and test whether model-visible annotation language changes preference construction.

Reproducibility Checklist

- **Code:** `scripts/run_label_anchor_model_experiment.py`.
- **Raw outputs:** `raw_generations.jsonl` under `results/label_anchor_model/`.

- **Aggregates:** `summary_by_condition.csv` and `summary_by_dimension.csv` in the same output directory.
- **Model:** `qwen/qwen-2.5-7b-instruct` through OpenRouter.
- **Decoding:** temperature 0, top-$p = 1$, max 80 output tokens.
- **Scale:** 24 items, 10 label conditions, two A/B orders, 480 calls.
- **Cost trace:** the recorded OpenRouter usage totals 86,965 tokens and about $0.0044 for this run.

References

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.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30.

Daniel Deutsch, George Foster, and Markus Freitag. 2023. Ties matter: Meta-evaluating modern metrics with pairwise accuracy and tie calibration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12914–12929. Association for Computational Linguistics.

Sumanth Doddapaneni, Mohammed Safi Ur Rahman Khan, Sshubam Verma, and Mitesh M. Khapra. 2024. Finding blind spots in evaluator LLMs with interpretable checklists. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16279–16309. Association for Computational Linguistics.

OpenRouter. 2026. OpenRouter API. https://openrouter.ai/. Accessed 2026-05-26.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023a. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023b. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao 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. *arXiv preprint arXiv:2306.05685*.