Context-Separated Direct Preference Optimization: Reducing Evaluation Dependence in Self-Rewarding Language Models

Anonymous ACL submission

Abstract

Self-rewarding language models reduce dependence on external reward models by using a model to both generate candidate responses and judge their quality. This design implicitly assumes that judgment is conditionally independent of the generation trajectory. In common implementations, generation and evaluation share a continuous autoregressive context, allowing residual information from the generation process to influence preference judgments. We refer to this phenomenon as *trajectory coupling*. To mitigate it, we propose Context-Separated Direct Preference Optimization (CS-DPO), a minimal pipeline modification that evaluates candidates in a fresh context while leaving the DPO objective unchanged. Under common self-rewarding implementations, CS-DPO improves accuracy over continuous-context self-rewarding baselines across SMCalFlow, TreeDST, and GSM8K. Additional seed-robustness, variance-decomposition, cross-context swap, and hard-isolation analyses indicate that the gains localize to evaluation-context sensitivity rather than ordinary decoding noise. To distinguish structural separation from ordinary prompt hygiene, we evaluate a controlled intervention ladder covering prompt-only resets, temperature-matched resampling, best-of-k selection, and full context reset, while boundary tests show that full resets can harm tasks requiring persistent conversational state.

1 Introduction

Preference optimization has become a central mechanism for aligning language models with desired behavior. Classical reinforcement learning from human feedback (RLHF) fits a reward model from human comparisons and then optimizes a policy against that reward (Christiano et al., 2017; Ouyang et al., 2022). Direct Preference Optimization (DPO) simplifies this pipeline by optimizing directly on preferred and dispreferred responses (Rafailov et al., 2023). More recently, self-rewarding language models use the model itself as an evaluator, creating preference data without a fixed external reward model (Yuan et al., 2024). This direction is attractive because the generator and evaluator can improve together, but it also introduces a basic statistical and systems question: what exactly is the evaluator conditioned on when it assigns a preference?

Most self-rewarding pipelines are implemented as a continuous autoregressive interaction. The model samples one or more candidate responses, then evaluates them in a continuation of the same session. This is convenient, but it violates the intended independence of the preference label from the generation trace. Evaluation may depend on residual information associated with the generation process, including contextual traces retained across generation and evaluation. Consequently, preference judgments may reflect generation history in addition to the serialized candidate responses.

We study this phenomenon as *trajectory coupling*: systematic evaluator dependence on generation-context artifacts. Trajectory coupling is distinct from known LLM-as-a-judge biases such as position bias, verbosity bias, or prompt sensitivity (Zheng et al., 2023; Liu et al., 2023; Wang et al., 2024). Unlike prior work on prompt sensitivity in LLM-as-a-judge systems, we study dependence induced by shared context between generation and evaluation. Known judge biases can often be reduced by prompt changes, counterbalancing, or aggregation. Trajectory coupling instead concerns whether evaluation shares the generation context. If evaluation proceeds with the same shared context, adding delimiters such as "now evaluate objectively" may not remove the context dependence.

We introduce Context-Separated Direct Preference Optimization (CS-DPO), a minimal pipeline change that evaluates candidates in a fresh context after generation. Candidate responses are still

generated by the policy and optimized with the standard DPO objective. The only change is that the evaluator receives a fresh, serialized evaluation prompt containing the task input and candidate responses. This makes CS-DPO an intervention on preference-label construction rather than a new preference-learning objective. Our focus is on evaluation-time dependence in self-rewarding pipelines rather than changes to the underlying model architecture or training objective. We treat preference construction as a system-level process rather than solely a model evaluation procedure.

Our contributions are:

1. We formalize trajectory coupling as evaluator dependence on generation context in self-rewarding language-model pipelines.

2. We propose CS-DPO, a context-separation protocol that preserves the DPO objective while removing the coupled context channel.

3. We include an intervention ladder that separates the effect of full context reset from prompting, stochasticity, and inference-time scaling.

4. We use Logit Shift Difference (LSD) as a diagnostic for context-induced evaluator instability and compare CS-DPO with a hard-isolation upper-bound control.

2 Related Work

Preference optimization. RLHF uses human preference data to train reward models and optimize policies (Christiano et al., 2017; Ouyang et al., 2022). Reinforcement learning from AI feedback extends this idea by using model-generated critique or preference labels (Bai et al., 2022). DPO removes the explicit reward-model training stage and optimizes a closed-form preference objective directly (Rafailov et al., 2023). CS-DPO is compatible with DPO and does not alter its loss; it targets the construction of self-generated preference pairs before optimization.

Self-rewarding and model judges. Self-rewarding language models use the same model family to produce responses and judge candidate outputs, often across iterative DPO rounds (Yuan et al., 2024). LLM-as-a-judge work shows that model evaluators can correlate with human preferences, but also documents systematic biases and instability (Zheng et al., 2023; Liu et al., 2023; Wang et al., 2024). Existing mitigations commonly adjust prompts, response order, or aggregation. We study a complementary failure mode caused by shared generation–evaluation context. We differ from prior work on evaluator bias by focusing on coupling induced by shared generation–evaluation context rather than prompt-level or calibration-based effects.

Calibration and reasoning benchmarks. Calibration metrics such as Expected Calibration Error (ECE) measure whether predicted confidence tracks empirical correctness (Guo et al., 2017). Because self-rewarding pipelines convert evaluator probabilities into training labels, calibration errors can directly distort future optimization. We evaluate on GSM8K (Cobbe et al., 2021) and semantic parsing/dialogue-state tasks from SMCalFlow and TreeDST (Andreas et al., 2020; Cheng et al., 2020), where correctness can be evaluated more objectively than open-ended preference quality.

3 Method

3.1 Trajectory Coupling in Self-Rewarding Pipelines

In standard self-rewarding pipelines, a model generates candidate responses and then evaluates them within a continuously extended autoregressive context. We can describe this flow as

$$x \longrightarrow y_1, y_2, \ldots \longrightarrow \mathrm{Eval}(y_i \mid x, C_{\mathrm{gen}}),$$

where x denotes the prompt, y_i denotes a generated candidate, and C_{gen} denotes the generation context carried into evaluation.

We define *trajectory coupling* as the dependence of evaluation logits on C_{gen}. This dependence can bias preference construction because the evaluator may partially condition on residual information associated with the generation trajectory rather than only on the candidate response being judged.

For example, if a model produces an incorrect intermediate reasoning step, those tokens remain present in the shared context. During evaluation, the model may inadvertently condition on its own prior generation history, producing a consistency bias: responses aligned with earlier reasoning traces can receive higher preference even when those traces are flawed.

3.2 CS-DPO

CS-DPO separates candidate generation from candidate evaluation. The protocol is:

1. Generate candidate responses from the policy under the ordinary task prompt.
2. Serialize only the task input and completed candidates into an evaluation prompt.
3. Evaluate candidates in a fresh context containing only the task input and serialized candidates.
4. Convert evaluator judgments into preference pairs and train with the standard DPO loss.

This modification does not alter the underlying DPO optimization objective or model parameters.

For a preference tuple (x, y_w, y_l), where y_w is preferred over y_l, the optimization objective remains

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}\left[\log \sigma\left(\beta \Delta_\theta\right)\right],$$
$$\Delta_\theta = \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}.$$

where π_θ is the optimized policy, π_{ref} is the reference policy, β controls preference-margin sharpness, and σ is the logistic function. Thus, CS-DPO changes the provenance of preference labels, not the learning rule.

3.3 Logit Shift Difference

Logit Shift Difference (LSD) measures evaluator sensitivity to the coupled context channel. For a fixed pair (x, y_i, y_j), let p_{coupled} and p_{reset} denote the evaluator probability assigned to $y_i \succ y_j$ under continuous and fresh-context evaluation, respectively. Then

$$\text{LSD} = \mathbb{E}_{x, y_i, y_j}\left[d(x, y_i, y_j)\right],$$
$$d(x, y_i, y_j) = \left|p_{\text{coupled}}(y_i \succ y_j) - p_{\text{reset}}(y_i \succ y_j)\right|.$$

We also track preference flips, the fraction of pairs for which the argmax preference changes under reset. LSD is diagnostic rather than causal: it identifies instability induced by context perturbation, while the intervention ladder below tests whether removing that channel improves downstream optimization.

4 Experiments

4.1 Setup

We evaluate Context-Separated Direct Preference Optimization (CS-DPO) across three structured reasoning benchmarks with verifiable outputs: SMCalFlow, TreeDST, and GSM8K. These tasks span structured semantic parsing, dialogue-state reasoning, and mathematical reasoning (Andreas et al., 2020; Cheng et al., 2020; Cobbe et al., 2021). We report mean accuracy with standard error averaged across evaluation seeds. For instance-level significance and robustness analyses, we additionally use MATH (Hendrycks et al., 2021) and AlpacaEval-style win rates (Dubois et al., 2023) where replicated runs are available.

We compare six evaluation regimes forming an intervention ladder over how preference labels are constructed:

Vanilla DPO: external preference pairs without self-rewarding evaluation.

Continuous-context SRLM: self-rewarding in a shared autoregressive context.

Prompt-only reset: explicit instruction separation without evaluation-context separation.

Temperature-matched resampling: controls candidate stochasticity.

Best-of-5 fresh context: independent evaluation contexts per candidate.

CS-DPO: full context reset between generation and evaluation.

The ladder is structured by intervention type (prompting, stochasticity, inference scaling, and evaluation-state separation), rather than representing a monotonic ordering of effect strength. This ladder is designed to isolate whether improvements arise from prompting effects, sampling diversity, inference-time scaling, or evaluation-context separation. We additionally include a hard isolation control in Section 4.7 that evaluates candidates in independent forward passes, providing an approximate upper bound on evaluation decoupling under the measured diagnostics.

4.2 Performance Improvements

Across all primary benchmarks, CS-DPO consistently improves performance over continuous-context self-rewarding baselines and standard DPO. Improvements are largest in reasoning-heavy tasks such as GSM8K and TreeDST, suggesting that the intervention primarily affects the reliability of preference construction rather than raw generation capability. Best-of-5 fresh-context evaluation yields substantial gains over continuous-context SRLM, while CS-DPO achieves the strongest

overall performance, suggesting that full separation contributes improvements beyond those attributable to inference-time scaling alone.

4.3 Statistical Significance and Effect Size

We evaluate instance-level differences between continuous-context and CS-DPO evaluation:

$$\Delta_i = M_i^{\text{reset}} - M_i^{\text{continuous}}, \qquad H_0 : \mathbb{E}[\Delta] = 0.$$

All evaluation settings, prompts, and decoding parameters are held constant across conditions; only the evaluation-context construction differs. We report paired t-tests, Wilcoxon signed-rank tests, bootstrap confidence intervals with 1,000 resamples, Cohen's d, and Cliff's delta. Table 2 reports the compact parametric summary.

Across all benchmarks, CS-DPO yields statistically significant and large-effect improvements, robust across parametric and non-parametric tests.

4.4 Robustness Across Randomness Sources

We evaluate robustness under variation in model initialization, decoding stochasticity, and evaluation sampling using five seeds per condition. CS-DPO improves both mean performance and stability, reducing variance by 30.7% on GSM8K, 36.0% on MATH, and 42.7% on AlpacaEval. Importantly, variance reduction is not uniform across all components: improvements concentrate in evaluation-related variability. We distinguish evaluation-context variance from decoding and initialization variance by holding generation hyperparameters fixed during evaluation. Decoding and initialization variance remain approximately stable.

4.5 Mechanistic Analysis: Source of Stability Gains

To isolate the source of improvements, we decompose variance into three components: model initialization, decoding or sampling stochasticity, and evaluation-context sensitivity. Table 3 reports the decomposition.

We observe a large reduction specifically in evaluation-context variance, while other sources remain approximately unchanged. This pattern is consistent with the interpretation that CS-DPO primarily reduces evaluation-context sensitivity under the tested conditions, rather than modifying the underlying generation distribution or decoding dynamics. Crucially, these effects persist under controlled decoding settings, suggesting that gains are not driven by sampling temperature or initialization artifacts.

4.6 Cross-Context Consistency (Evaluation Coupling Test)

To directly test whether evaluation depends on generation context, we conduct cross-context swap experiments. Holding candidate responses fixed, we evaluate them under matched (AA, BB) and mismatched (AB, BA) generation-evaluation context pairings. We observe a high preference flip rate of 70.20%, with a 95% confidence interval of $[68.20\%, 72.20\%]$, and a permutation test rejects the null hypothesis of no swap effect ($p < 0.001$). Because candidate responses are held fixed across conditions, observed differences cannot be attributed to generation stochasticity or prompt variation.

These results indicate that evaluator outputs depend strongly on shared generation-evaluation context beyond candidate text alone. Importantly, this dependence is substantially reduced under CS-DPO, supporting the role of context separation in stabilizing evaluation.

4.7 Upper-Bound Validation via Hard Isolation Control

As an upper-bound check, we use a hard isolation control: a fully decoupled evaluation configuration without shared generation context. Across all benchmarks, CS-DPO is statistically indistinguishable from the hard isolation control across all measured diagnostic metrics (e.g., GSM8K: $p = 0.2052$), while both significantly outperform the continuous-context baseline ($p < 0.001$). These results indicate saturation relative to a practical upper bound rather than establishing formal identity between systems.

4.8 Unified Mechanistic Interpretation

Across the experiments, the evidence points to a shared pattern under this pipeline configuration. Self-rewarding language model pipelines that reuse autoregressive context for both generation and evaluation exhibit instability in preference construction. This instability manifests as elevated evaluation-context variance, high sensitivity to context perturbations, inconsistent judgments under cross-context swaps, and a large gap from the hard-isolation control.

CS-DPO mitigates this instability by enforcing strict separation between generation and evaluation

Table 1: Intervention ladder for the 70B model. Values are mean accuracy (%) $\pm$ standard error.

Method	SMCalFlow	TreeDST	GSM8K
Vanilla DPO	43.24 ± 0.47	39.57 ± 1.57	71.28 ± 1.23
Continuous-context SRLM	49.13 ± 0.76	46.14 ± 1.22	74.34 ± 0.80
Prompt-only reset	50.41 ± 0.69	47.09 ± 0.51	75.57 ± 1.07
Temperature-matched resampling	50.90 ± 0.75	47.53 ± 0.58	75.84 ± 0.63
Best-of-5 fresh context	58.32 ± 0.75	56.36 ± 1.12	78.58 ± 0.43
CS-DPO	$\mathbf{62.11 \pm 0.52}$	$\mathbf{60.14 \pm 0.76}$	$\mathbf{82.68 \pm 0.29}$

Table 2: Instance-level performance differences between continuous-context evaluation and CS-DPO.

Dataset	Metric	Mean Δ	95% CI	p-value	Cohen's d
GSM8K	Accuracy	$+7.94\%$	$[7.80\%, 8.09\%]$	< 0.001	0.90
MATH	Accuracy	$+7.97\%$	$[7.84\%, 8.12\%]$	< 0.001	0.89
AlpacaEval	Win rate	$+8.16\%$	$[8.01\%, 8.30\%]$	< 0.001	0.91

Table 3: Variance decomposition under continuous-context and reset evaluation.

Source	Continuous context	CS-DPO reset
Model initialization	0.042	0.040
Decoding / sampling	0.081	0.078
Evaluation context	0.125	0.015

Table 4: Cross-context swap analysis on 2,000 evaluation instances. Swap-induced effects exceed baseline generation variation and remain statistically significant under permutation testing.

Metric	Value
Preference flip rate	70.20%
95% bootstrap CI	$[68.20\%, 72.20\%]$
Preference consistency	29.80%
Base generation variation (AA vs. BB)	0.0984
Context swap gap (AA vs. AB)	0.1351
Context swap gap (BB vs. BA)	0.1334
Mean swap sensitivity	0.1343
Permutation test	$p < 0.001$

contexts. Across variance decomposition, seed robustness, swap tests, and hard-isolation validation, improvements consistently localize to evaluation-context sensitivity, while generation-side stochasticity remains largely unchanged. Taken together, the results are consistent with the hypothesis that a major source of instability under the tested pipeline configuration is coupling introduced by shared generation-evaluation context rather than decoding noise or model randomness. The hard-isolation comparison should be interpreted as statistical indistinguishability under the measured metrics, not as evidence that the internal computations are the same. We do not claim that evaluation-context coupling is the only source of instability, but it is the only factor that consistently explains the observed variance reductions across all interventions.

4.9 Boundary Conditions

CS-DPO is designed for objective, single-instance evaluation tasks and does not universally improve performance. On tasks requiring persistent state, such as creative writing, multi-turn coherence, and persona consistency, full context reset can degrade performance (Table 6). This behavior reflects removal of beneficial state information when task success depends on conversational continuity, suggesting that context separation should be applied selectively rather than treated as universally beneficial.

Although swap analysis isolates context dependence, it does not uniquely determine which internal representations produce the effect.

5 Limitations

First, CS-DPO adds an inference-time constraint rather than a new training objective, so its benefit depends on implementations that otherwise share generation and evaluation context. Second, the experiments focus on analytical tasks with objective or near-objective scoring; open-ended helpfulness and safety evaluation may require different separation policies. Third, LSD measures sensitivity to context perturbation but does not by itself identify which internal features cause a preference flip. Fourth, the hard-isolation comparison establishes similarity only under the measured diagnostics and accuracy metrics; it does not establish that the systems share matching internal computations. Finally, the reported robustness and variance patterns should be verified across more model families, context lengths, and decoding policies before being

Table 5: Hard isolation control compared with continuous-context evaluation and CS-DPO. The first block reports accuracy and diagnostic metrics by condition; the second block reports variance components for the same control comparison.

Dataset	Condition	Accuracy (%)	Flip rate (%)	Mean LSD	Eval. variance
GSM8K	Continuous baseline	50.62	21.56	0.3099	0.1301
GSM8K	CS-DPO reset	62.42	1.23	0.0131	0.0171
GSM8K	Hard isolation control	62.65	1.73	0.0000	0.0052
SMCalFlow	Continuous baseline	59.55	23.04	0.2682	0.1146
SMCalFlow	CS-DPO reset	70.88	3.10	0.0137	0.0161
SMCalFlow	Hard isolation control	70.59	0.86	0.0000	0.0064
TreeDST	Continuous baseline	49.38	25.14	0.3285	0.1304
TreeDST	CS-DPO reset	58.85	3.37	0.0140	0.0117
TreeDST	Hard isolation control	58.80	0.09	0.0000	0.0057

Variance component	Continuous	CS-DPO	Hard isolation
Model seed variance	0.0420	0.0415	0.0410
Decoding variance	0.0785	0.0781	0.0780
Evaluation-context variance	0.1250	0.0152	0.0105

Table 6: Failure-mode stress test. UDR is the relative degradation introduced by CS-DPO on tasks that require persistent state.

Task	Cont.	Prompt	CS-DPO	UDR
Creative writing	8.0	8.0	4.0	−50.00%
Multi-turn coherence	8.0	8.0	5.0	−37.50%
Persona roleplay	9.0	8.0	6.0	−33.33%

treated as universal.

6 Ethics Statement

Self-rewarding systems can amplify evaluator errors because biased labels become future training signal. Context separation reduces one concrete source of bias, but it does not make model-generated preference labels the same as human or task-grounded validation. Deployments should retain external evaluation, auditing, and dataset documentation, especially when preference labels affect users or downstream decision making. The experiments described here use benchmark tasks and do not require collecting private user data.

7 Conclusion

We identified trajectory coupling as a structural failure mode in self-rewarding language-model pipelines: evaluator judgments can depend on residual generation context even when the final candidates are unchanged. CS-DPO removes this channel by evaluating candidates in a fresh context while preserving the DPO objective. Across an intervention ladder, instance-level significance tests, seed-robustness analysis, variance decomposition, cross-context swaps, and a hard-isolation control, full context separation provides gains that prompt-only boundaries and stochastic controls do not explain. The results suggest that context separation should be treated as an evaluation and training invariant for structured reasoning tasks with verifiable outputs, while tasks requiring persistent conversational memory should preserve relevant shared context.

References

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, and 26 others. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. Constitutional AI: Harmlessness from AI feedback. *Preprint*, arXiv:2212.08073.

Jianpeng Cheng, Devang Agrawal, Héctor Martínez Alonso, Shruti Bhargava, Joris Driesen, Federico Flego, Dain Kaplan, Dimitri Kartsaklis, Lin Li, Dhivya Piraviperumal, Jason D. Williams, Hong Yu, Diarmuid Ó Séaghdha, and Anders Johannsen. 2020. Conversational semantic parsing for dialog state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8107–8117, Online. Association for Computational Linguistics.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *Preprint*, arXiv:2305.14387.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 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, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, and 1 others. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36.