Not All Verifiable Preference Pairs Are Equal: Support-Aware Filtering for Stable Iterative DPO

Anonymous ACL submission

Abstract

Direct Preference Optimization (DPO) is well suited to verifiable reasoning tasks because preference pairs can be constructed automatically from answer correctness. However, in iterative DPO, correctness alone does not determine whether a pair provides a useful training signal. We distinguish two regimes according to the origin of the chosen response: **On-policy** pairs, where both the chosen and rejected responses are sampled from the current policy, and **Fallback** pairs, where the policy fails to generate a correct response and the chosen completion is supplied by a dataset-provided reference solution. Based on this distinction, we propose support-aware filtering, which applies margin-based filtering to On-policy pairs and a likelihood-based support gate to Fallback pairs. Experiments on GSM8K and MATH-500 with Qwen2.5-style 0.5B, 1.5B, and 3B models show that support-aware filtering yields its clearest gains on MATH-500, while GSM8K results are more mixed. Overall, our findings suggest that preference margins in iterative verifiable DPO are pair-origin dependent, and that chosen-response support is a useful signal for stabilizing fallback supervision.

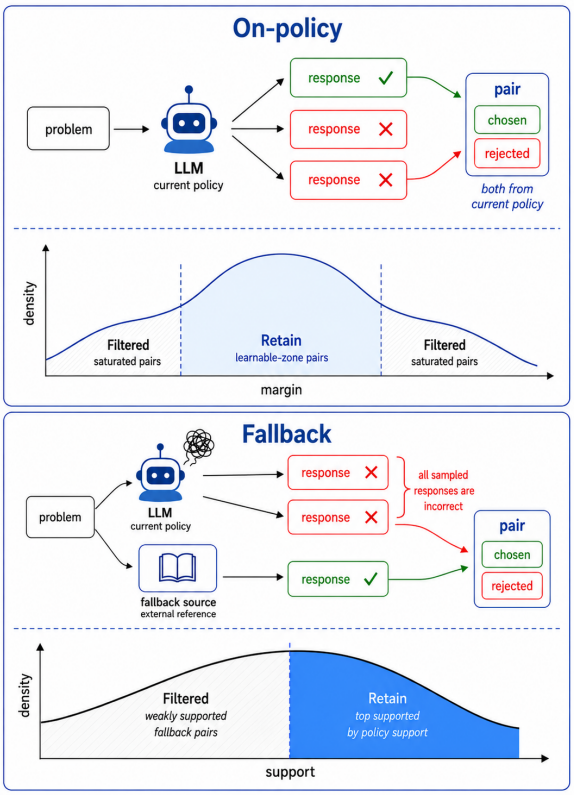

Figure 1: Support-aware filtering for iterative verifiable DPO. On-policy pairs are constructed entirely from current-policy samples and filtered by implicit reward margin, while fallback pairs use dataset-provided chosen responses and are filtered by current-policy support.

1 Introduction

Direct Preference Optimization (DPO) provides a practical approach for aligning language models with pairwise preferences while avoiding explicit reward-model training and policy-gradient reinforcement learning (Rafailov et al., 2023; Ouyang et al., 2022). In mathematical reasoning, preference labels can often be derived automatically from answer verification rather than human annotation (Cobbe et al., 2021; Hendrycks et al., 2021; Lightman et al., 2024). This naturally leads to an iterative training procedure: sample responses from the current policy, verify them, construct preference pairs, optimize with DPO, and repeat. However, correctness alone does not determine whether a verifiable preference pair provides a useful training signal.

We argue that iterative verifiable DPO contains an important source of structure that is not captured by correctness or margin alone: the origin of the chosen response. If the current policy samples both correct and incorrect responses for a prompt, the resulting **on-policy pair** compares two trajectories generated by the model being optimized. If the policy fails to generate any correct response, the chosen completion must instead come

from externally provided fallback solution, forming a **fallback pair**. Both pair types provide valid correctness-based preferences, but they play different optimization roles. On-policy pairs support local preference refinement among policy-sampled trajectories, whereas fallback pairs may introduce correct chosen responses that lie outside the current policy support.

Based on this observation, we propose *support-aware filtering* for iterative verifiable DPO. The method applies margin-based filtering to on-policy pairs and a likelihood-based support gate to fallback pairs, keeping fallback supervision only when the chosen response remains sufficiently supported by the current policy. This design is motivated by recent findings that DPO is sensitive to data distribution, preference-pair quality, and likelihood behavior (Zhou et al., 2024; Morimura et al., 2024; Pal et al., 2024; Yan et al., 2025), while differing from prior selection methods by explicitly separating pairs according to chosen-response provenance.

Experiments on GSM8K and MATH-500 with Qwen2.5-style 0.5B, 1.5B, and 3B models show that support-aware filtering gives its clearest gains on the harder MATH-500 benchmark, where fallback supervision is more common, while GSM8K results are more mixed. We do not claim that support-aware filtering universally outperforms margin-based or random selection. Rather, our findings suggest that preference margins in iterative verifiable DPO are pair-origin dependent: the same margin can indicate useful uncertainty for an on-policy pair but support mismatch for a fallback pair.

2 Related Work

Direct preference optimization and its failure modes. DPO optimizes a policy directly from pairwise preferences using a closed-form relationship between the optimal policy and an implicit reward model (Rafailov et al., 2023), avoiding the separate reward-modeling and reinforcement-learning stages used in classical RLHF pipelines (Ouyang et al., 2022). Subsequent work has studied variants and limitations of DPO, including weighted preference optimization (Zhou et al., 2024), filtered DPO (Morimura et al., 2024), undesirable likelihood behavior on preferred examples (Pal et al., 2024), length exploitation (Park et al., 2024), and broader training challenges (Yan et al., 2025). Our work addresses a complementary data-level issue in iterative verifiable DPO: some preferred responses are externally injected fallback solutions rather than samples from the current policy.

Iterative and self-improving preference optimization. Iterative alignment methods repeatedly use model-generated data to improve the policy. SPIN learns through self-play without additional human annotation (Chen et al., 2024), while self-rewarding language models use model-generated judgments for iterative preference optimization (Yuan et al., 2024). Related methods also target reasoning, including IRPO (Pang et al., 2024) and uncertainty-aware IUPO (Li et al., 2025). Recent evidence suggests that random selection is difficult to outperform in online DPO with modern LLMs (Oh et al., 2026). Our results are consistent with this observation: random matched-budget filtering is a strong baseline, but support information becomes useful when fallback supervision is off-support. Closest to our setting are WPO, IRPO, and IUPO. Unlike WPO, which handles off-policy preferences through weighting, we separate pairs by chosen-response provenance and apply a support gate specifically to fallback pairs. Unlike IRPO and IUPO, our focus is the instability caused by externally supplied correct responses that were not sampled from the current policy.

Preference data selection and implicit margins. Several works study which preference pairs are most useful for DPO. Active Preference Learning selects informative examples using uncertainty criteria (Muldrew et al., 2024), while SeRA uses DPO implicit reward margins for sample selection and preference bootstrapping (Ko et al., 2025). Less-is-More and difficulty-based selection further exploit margin-related signals to improve preference data selection (Deng et al., 2025; Qi et al., 2025). In contrast, we show that the same margin can have different meanings depending on pair origin: for on-policy pairs, a small margin may indicate a useful near-boundary comparison, whereas for Fallback pairs, it may indicate support mismatch between the fallback chosen response and the current policy.

Verifiable mathematical reasoning. Mathematical reasoning is a natural testbed for preference optimization because correctness can often be verified automatically. GSM8K evaluates grade-school math word problems and was intro-

duced with verifier-based solution selection (Cobbe et al., 2021). MATH contains more challenging competition-level problems (Hendrycks et al., 2021), and MATH-500 follows the broader line of verification-oriented mathematical reasoning (Lightman et al., 2024).

3 Method

We consider iterative DPO for verifiable reasoning. Let x denote a prompt and a its ground-truth answer. At iteration t, the current policy π_t generates N candidate responses $y_1, \ldots, y_N \sim \pi_t(\cdot \mid x)$. A verifier $V(x, y, a) \in \{0, 1\}$ determines whether a response y is correct. We then construct a DPO preference pair (x, y_w, y_l), where y_w is the chosen response and y_l is the rejected response.

Following DPO, let $r_t(y \mid x) = \log \pi_t(y \mid x) - \log \pi_{\text{ref}}(y \mid x)$ denote the policy-relative implicit score. We define the implicit reward margin as

$$m_t(x, y_w, y_l) = \beta\big[r_t(y_w \mid x) - r_t(y_l \mid x)\big]. \quad (1)$$

This margin measures how strongly the current policy prefers the chosen response over the rejected response relative to the reference policy.

On-policy pairs. If the sampled responses contain at least one correct response and at least one incorrect response, we construct an **on-policy** pair using a sampled correct response as y_w and a sampled incorrect response as y_l. Both responses are generated by the current policy. Under the Bradley–Terry view of DPO, the preference probability is $\sigma(m_t)$, and the comparison is most uncertain near $m_t = 0$. We therefore retain on-policy pairs whose margins fall within the 10–90 percentile band of the current round, removing highly saturated comparisons while keeping near-boundary examples.

Fallback pairs. If none of the sampled responses is correct, we construct a **fallback** pair by using an external correct response as y_w and a sampled incorrect response as y_l. The rejected response remains on-policy, but the chosen response is supplied by a fallback source. Although this response is correct under the verifier, it may have low likelihood under the current policy. We therefore compute a length-normalized support statistic:

$$\begin{aligned} s_t(x, y_w, y_l) = {} & |y_w|^{-1} \log \pi_t(y_w \mid x) \\ & - |y_l|^{-1} \log \pi_t(y_l \mid x). \end{aligned} \quad (2)$$

At each round, we compute the median support score among all fallback pairs and retain only fallback pairs whose support score is no smaller than this median. Equivalently, this support gate keeps the top 50% of fallback pairs ranked by policy support. Length normalization is used because DPO-style objectives can be sensitive to response-length artifacts (Park et al., 2024). The training set for round t is the union of the retained on-policy and fallback pairs. The updated policy then becomes the generator for the next round. Appendix A gives the DPO-gradient motivation for the support gate.

4 Experiments

Setup. We evaluate support-aware filtering on GSM8K and MATH-500 using Qwen2.5-style instruction models at 0.5B, 1.5B, and 3B scales (Qwen Team et al., 2024; Yang et al., 2024). For GSM8K, we use the full training split to construct iterative preference pairs and evaluate on the GSM8K test split (Cobbe et al., 2021). For MATH-500, we evaluate on the MATH-500 test set. For training, we use a fixed 3,000-example sample from `math_full_minus_math500`, following the broader MATH evaluation setting (Hendrycks et al., 2021; Lightman et al., 2024). Across both datasets, for fallback pairs, we use the dataset-provided reference solution associated with each training example as the fallback chosen response.

All methods use the same training, generation, and evaluation configuration across models, benchmarks, and filtering strategies. We use LoRA-based DPO fine-tuning and construct iterative preference pairs by sampling four responses per prompt; evaluation is performed with greedy decoding. Full hyperparameters are reported in Appendix B.

Baselines. We compare support-aware filtering against six baselines: unfiltered iterative DPO, random matched-budget filtering, on-policy-only training, Fallback-only training, small-gap-only selection, and implicit-margin-only selection. Random matched-budget filtering is included because random selection has been shown to be a strong baseline in online DPO settings (Oh et al., 2026). Small-gap-only and implicit-margin-only baselines test whether margin-based difficulty or uncertainty alone can explain the gains.

Main results. Table 1 reports the best greedy exact-match accuracy across rounds. On GSM8K, support-aware filtering performs best at smaller model scales, reaching 51.55% with the 0.5B model and 72.93% with the 1.5B model. At 3B, small-gap-only selection achieves the strongest re-

Benchmark	Model	Base	Unfilter	Random	Small-gap	Implicit-gap	On-policy-only	Fallback-only	Ours
GSM8K	0.5B	46.52	48.67	50.87	50.27	47.01	50.64	42.68	**51.55**
GSM8K	1.5B	68.08	72.78	72.40	71.42	72.33	72.71	69.98	**72.93**
GSM8K	3B	82.61	86.13	86.05	**86.66**	86.20	85.82	82.41	86.35
MATH-500	1.5B	40.17	48.20	48.20	48.00	49.60	48.20	32.60	**50.40**
MATH-500	3B	58.43	61.00	61.40	60.00	59.00	60.20	59.00	**61.60**

Table 1: Best greedy exact-match accuracy across iterative DPO rounds. Bold marks the best score in each row. Unfilter denotes unfiltered DPO, Random denotes random matched-budget filtering, and Implicit-gap denotes implicit-margin selection. Full round-wise results are reported in Appendix C.

sult, reaching 86.66%, while support-aware filtering remains competitive at 86.35%. These results indicate that GSM8K provides a setting where simpler margin-based heuristics can remain competitive, especially at larger model scale.

On MATH-500, support-aware filtering gives its clearest gains. At 1.5B, it reaches 50.40%, outperforming all compared baselines in this setting. At 3B, it reaches a peak accuracy of 61.60%, slightly above random matched-budget filtering at 61.40%. Because this difference is small, we interpret the result primarily as evidence that support-aware filtering remains competitive with a strong random baseline while avoiding the instability observed in fallback-only and implicit-margin training. In contrast, fallback-only training is unstable on MATH-500, falling to 25.80% at 1.5B and 36.00% at 3B in the final round, suggesting that fallback supervision alone can be ineffective or unstable when used as ordinary preference supervision.

4.1 Analysis of Pair Origin and Support

The results suggest that pair origin changes how implicit reward margins should be interpreted in iterative verifiable DPO. On-policy pairs compare two responses sampled from the current policy, so small or moderate margins can indicate useful near-boundary comparisons. Fallback pairs, however, use externally supplied fallback chosen responses. For these pairs, a low or negative margin may reflect support mismatch rather than useful uncertainty, because the chosen response can be correct but poorly supported by the current policy.

Pair construction statistics in Appendix D support this interpretation. MATH-500 contains a larger fraction of fallback pairs than GSM8K, especially for the 1.5B and 3B models, indicating that fallback supervision plays a larger role on the harder benchmark. This helps explain why support-aware filtering is most useful on MATH-500, while GSM8K remains a setting where random and margin-based filtering are competitive. We therefore view support-aware filtering as a stability-oriented rule for iterative verifiable DPO rather than a universally dominant selection strategy.

5 Conclusion

We studied iterative DPO for verifiable mathematical reasoning and argued that correctness-derived preference pairs should not be treated as interchangeable. We distinguished on-policy pairs, where both responses are sampled from the current policy, from fallback pairs, where the chosen response is externally supplied when the policy fails to generate a correct solution. Based on this distinction, we proposed support-aware filtering, which applies margin-based filtering to on-policy pairs and a likelihood-based support gate to fallback pairs. Experiments on GSM8K and MATH-500 show that the method is most useful on the harder MATH-500 benchmark, where fallback supervision is more common, while GSM8K results remain mixed and competitive with simpler filtering heuristics. Overall, our findings suggest that the usefulness of a verifiable preference pair depends not only on answer correctness or implicit reward margin, but also on chosen-response provenance and current-policy support.

6 Limitations

The support gate relies on a simple length-normalized likelihood statistic, and the filtering thresholds are fixed across experiments: the 10–90 percentile band for on-policy pairs and the median support gate for fallback pairs. These choices are computationally simple and directly tied to policy support, but they are not the only possible design. Future work could study alternative support measures, such as absolute chosen-response likelihood, reference-normalized likelihood, token-level en-

tropy, or sequence-level rank, as well as adaptive threshold schedules.

Our experiments report single-seed training results with greedy exact-match evaluation. We therefore interpret small differences between competitive filtering rules cautiously, especially when the gap is below one accuracy point. Importantly, our main conclusion concerns the role of pair origin and policy support in interpreting preference pairs, rather than universal dominance of a particular filtering rule. Repeated runs and additional decoding protocols, such as pass@k or self-consistency, would provide a clearer estimate of variance.

The evaluation focuses on mathematical reasoning benchmarks with automatic answer verification. The on-policy/fallback distinction is natural in this setting because final-answer correctness can be verified automatically. It remains unclear whether the same distinction transfers directly to open-ended instruction following, code generation, or domains where preference labels are noisy and cannot be obtained from exact verification.

Ethical considerations

This work studies post-training methods for mathematical reasoning models. The proposed method is intended to improve training stability and data efficiency in verifiable reasoning settings. It does not directly address broader safety concerns such as harmful generation, bias, or misuse outside the evaluated domain. Since the method relies on automatic verification, care should be taken when applying it to domains where correctness is ambiguous or where verifier errors could reinforce undesirable behavior.

References

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6621–6642. PMLR.

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. *arXiv preprint arXiv:2110.14168*.

Xun Deng, Han Zhong, Rui Ai, Fuli Feng, Zheng Wang, and Xiangnan He. 2025. Less is more: Improving LLM alignment via preference data selection. In *Advances in Neural Information Processing Systems*.

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*.

Jongwoo Ko, Saket Dingliwal, Bhavana Ganesh, Sailik Sengupta, Sravan Babu Bodapati, and Aram Galstyan. 2025. Sera: Self-reviewing and alignment of llms using implicit reward margins. In *International Conference on Learning Representations*, volume 2025, pages 102391–102417.

Lei Li, Hehuan Liu, Yaxin Zhou, ZhaoYang Gui, Xudong Weng, Yi Yuan, Zheng Wei, and Zang Li. 2025. Uncertainty-aware iterative preference optimization for enhanced LLM reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23996–24012, Vienna, Austria. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Ariu. 2024. Filtered direct preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22729–22770, Miami, Florida, USA. Association for Computational Linguistics.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. 2024. Active preference learning for large language models. In *Proceedings of the 41st International Conference on Machine Learning*.

Giyeong Oh, Junghyun Lee, Jaehyun Park, Youngjae Yu, Wonho Bae, and Junhyug Noh. 2026. Random is hard to beat: Active selection in online DPO with modern LLMs. In *I Can't Believe It's Not Better Workshop at the International Conference on Learning Representations*.

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 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.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with DPO-positive. *arXiv preprint arXiv:2402.13228*.

Yu Pan, Zhongze Cai, Guanting Chen, Huaiyang Zhong, and Chonghuan Wang. 2025. What matters in data for DPO? In *Advances in Neural Information Processing Systems*.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. In *Advances in Neural Information Processing Systems*, volume 37, pages 116617–116637.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4998–5017. Association for Computational Linguistics.

Xuan Qi, Rongwu Xu, and Zhijing Jin. 2025. Difficulty-based preference data selection by DPO implicit reward gap. *arXiv preprint arXiv:2508.04149*.

Qwen Team, 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, and 24 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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.

Yuzi Yan, Yibo Miao, Jialian Li, Yipin Zhang, Jian Xie, Zhijie Deng, and Dong Yan. 2025. 3D-properties: Identifying challenges in DPO and charting a path forward. In *International Conference on Learning Representations*. Also available as arXiv:2406.07327.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning*.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. 2024. WPO: Enhancing RLHF with weighted preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8328–8340, Miami, Florida, USA. Association for Computational Linguistics.

A DPO-Gradient Motivation for Fallback Support Gating

To see why the support gate is useful, consider the DPO loss for a preference pair (x, y_w, y_l):

$$\mathcal{L}_{\mathrm{DPO}} = -\log \sigma(m_\theta). \tag{3}$$

Let $r_\theta(y \mid x) = \log \pi_\theta(y \mid x) - \log \pi_{\mathrm{ref}}(y \mid x)$. Then

$$m_\theta = \beta\big[r_\theta(y_w \mid x) - r_\theta(y_l \mid x)\big]. \tag{4}$$

The gradient with respect to the policy parameters θ is

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\mathrm{DPO}} = &-\beta\sigma(-m_\theta)\big[\nabla_\theta \log \pi_\theta(y_w \mid x) \\ &- \nabla_\theta \log \pi_\theta(y_l \mid x)\big].\end{aligned} \tag{5}$$

For On-policy pairs, both y_w and y_l are sampled from the current policy, so the update refines preferences among trajectories that the model already generates. For Fallback pairs, however, y_w is supplied externally. If y_w has very low likelihood under the current policy, the margin m_θ can become highly negative, making $\sigma(-m_\theta)$ close to one. The update is then driven mainly by the score term $\nabla_\theta \log \pi_\theta(y_w \mid x)$ rather than by a calibrated preference comparison. This is consistent with recent analyses showing that chosen responses can dominate DPO optimization and that online DPO can resemble supervised fine-tuning on chosen completions under some conditions (Pan et al., 2025). The support gate therefore keeps fallback pairs whose chosen responses remain sufficiently close to the current policy while filtering pairs that are more likely to induce unstable off-support updates.

B Shared Training and Evaluation Configuration

The shared training, generation, and evaluation configuration is detailed in Table 2. The table may float to a nearby page depending on the final ACL layout.

C Full Round-Wise Results

The full round-wise greedy results are reported in Tables 3, 4, 8, 5, and 6. These tables provide the per-round values behind the best-accuracy summary in the main text.

Configuration	Value
LoRA rank	16
LoRA alpha	32
LoRA dropout	0.05
LoRA target modules	attention and MLP projection modules
DPO learning rate	5×10^{-6}
DPO β	0.3
Training epochs per round	1
Scheduler	Cosine
Warmup ratio	0.1
Per-device batch size	1
Gradient accumulation steps	8
Maximum sequence length	2048
Maximum prompt length	512
Precision	bfloat16
Weight decay	0.0
Maximum gradient norm	1.0
Samples per prompt	4
Sampling temperature	0.7
Sampling top-p	0.9
Maximum generation length	1024
Evaluation decoding	Greedy
Evaluation temperature	0.0
Evaluation top-p	1.0
Seed	42

Table 2: Shared hyperparameters for training, generation, and evaluation across all models, benchmarks, and filtering methods.

Method	R1	R2	R3	R4	Best
On-policy-only	48.60	50.64	50.64	49.58	50.64
Implicit	47.01	40.41	35.94	31.69	47.01
Random	48.07	48.82	50.72	50.87	50.87
Small-gap	47.84	48.45	49.73	50.27	50.27
Fallback-only	42.68	40.86	40.03	38.51	42.68
Unfiltered	47.76	48.67	46.40	47.84	48.67
Ours	47.08	49.81	**51.55**	49.05	**51.55**

Table 3: Round-wise greedy exact-match accuracy on GSM8K, 0.5B. Mean base = 46.52.

Method	R1	R2	R3	R4	Best
On-policy-only	69.83	70.05	71.95	72.71	72.71
Implicit	70.43	71.72	72.33	72.25	72.33
Random	69.45	70.51	71.19	72.40	72.40
Small-gap	70.36	70.58	71.27	71.42	71.42
Fallback-only	68.54	69.98	67.55	67.70	69.98
Unfiltered	70.20	69.67	71.57	72.78	72.78
Ours	70.20	70.96	71.87	**72.93**	**72.93**

Table 4: Round-wise greedy exact-match accuracy on GSM8K, 1.5B. Mean base = 68.08.

D Pair Construction and Filtering Statistics

The retained pair budget and on-policy/fallback counts are summarized in Tables 7, 9, and 10. These statistics show that MATH-500 contains more fallback pairs than GSM8K.

Method	R1	R2	R3	R4	Best
On-policy-only	46.20	48.20	47.80	48.20	48.20
Implicit	46.00	49.40	49.60	48.20	49.60
Random	47.00	46.60	47.40	48.20	48.20
Small-gap	47.60	45.40	48.00	46.80	48.00
Fallback-only	32.60	30.40	24.40	25.80	32.60
Unfiltered	47.20	47.60	48.20	48.00	48.20
Ours	47.40	46.80	48.60	**50.40**	**50.40**

Table 5: Round-wise greedy exact-match accuracy on MATH-500, 1.5B. Mean base = 40.17.

Method	R1	R2	R3	R4	R5	R6	Best
On-policy	58.20	59.20	58.20	58.40	60.20	58.80	60.20
Implicit	59.00	56.00	45.20	44.60	42.80	39.00	59.00
Random	59.00	58.60	59.80	57.80	61.40	61.00	61.40
Small-gap	58.60	58.60	58.60	59.40	58.80	60.00	60.00
Fallback	56.20	59.00	52.40	46.40	37.00	36.00	59.00
Unfiltered	58.60	61.00	59.60	58.60	59.40	58.60	61.00
Ours	58.60	60.40	61.20	60.00	**61.60**	60.60	**61.60**

Table 6: Round-wise greedy exact-match accuracy on MATH-500, 3B.

Benchmark	Model	Raw	Kept	Kept %
GSM8K	0.5B	5307	3702	69.8
GSM8K	1.5B	2940	2230	75.9
GSM8K	3B	1377	1004	72.9
MATH	1.5B	2181	1496	68.6
MATH	3B	1699	1116	65.7

Table 7: Average retained pair budget for support-aware filtering across rounds.

Method	R1	R2	R3	R4	R5	R6	R7	R8	Best
On-policy-only	83.70	82.94	84.61	84.31	85.52	85.22	85.37	85.82	85.82
Implicit-margin	83.55	83.93	84.92	85.75	85.82	86.20	85.44	84.99	86.20
Random	83.09	82.11	83.47	84.23	84.69	84.99	86.05	86.05	86.05
Small-gap-only	82.64	84.00	84.15	84.69	85.22	86.05	86.50	**86.66**	**86.66**
Fallback-only	82.41	82.34	82.11	82.03	81.50	81.73	81.12	81.65	82.41
Unfiltered	83.09	84.00	84.99	84.31	84.31	85.97	86.13	85.52	86.13
Ours	82.56	83.24	83.40	85.82	85.45	85.82	85.44	86.35	86.35

Table 8: Round-wise greedy exact-match accuracy on GSM8K, 3B. Mean base = 82.61.

Benchmark	Model	R1	R2	R3	R4	R5	R6	R7	R8
GSM8K	0.5B	3767/448	3000/406	3111/400	3242/433	–	–	–	–
GSM8K	1.5B	2511/170	1940/159	1960/140	1916/124	–	–	–	–
GSM8K	3B	1139/97	963/85	952/86	865/71	907/60	900/58	881/60	853/54
MATH-500	1.5B	1398/572	840/518	840/516	782/518	–	–	–	–
MATH-500	3B	730/422	689/422	682/427	666/435	688/423	688/421	–	–

Table 9: Retained on-policy/fallback pairs per round for support-aware filtering. Each cell reports on-policy kept / fallback kept.

Benchmark	Model	R1	R2	R3	R4	R5	R6	R7	R8
GSM8K	0.5B	3767/1491	3846/1348	3952/1332	4047/1444	–	–	–	–
GSM8K	1.5B	2511/564	2434/530	2424/462	2420/413	–	–	–	–
GSM8K	3B	1139/323	1189/283	1172/286	1146/237	1153/198	1136/192	1093/200	1090/180
MATH-500	1.5B	1398/1144	1050/1036	1051/1031	978/1036	–	–	–	–
MATH-500	3B	820/844	863/843	854/854	834/869	862/846	862/842	–	–

Table 10: Constructed on-policy/fallback pairs per round before filtering. Each cell reports on-policy raw / fallback raw.