RAZORSFT: ON-POLICY SUPERVISED FINE-TUNING WITH KL-MINIMAL TARGET SELECTION FOR CONTINUAL LEARNING

FARS
Analemma
fars@analemma.ai

ABSTRACT

Continual fine-tuning of large language models (LLMs) on sequential tasks leads to catastrophic forgetting, where previously learned capabilities degrade. While reinforcement learning (RL) methods like GRPO mitigate forgetting, they introduce substantial complexity. We propose RazorSFT, a simple on-policy supervised fine-tuning method that achieves strong forgetting mitigation without RL's complexity. RazorSFT samples candidate responses from the current model, filters them through a task verifier, and selects the highest log-probability correct response as the training target. This KL-minimal selection ensures training targets remain close to the current policy distribution. On a 3-stage continual learning benchmark, RazorSFT reduces forgetting by 60.4 percentage points compared to offline SFT (FM -0.039 vs. -0.643) while outperforming GRPO on average accuracy (0.616 vs. 0.515) and task adaptation (Countdown: 0.628 vs. 0.261). Ablation studies reveal that on-policy data accounts for 76% of the forgetting improvement, demonstrating that the benefits of on-policy learning can be obtained within a simple SFT framework.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed in settings that require continual adaptation to new tasks and domains. However, sequential fine-tuning on new tasks leads to *catastrophic forgetting*—the degradation of previously learned capabilities (van de Ven et al., 2024; Kirkpatrick et al., 2016). This phenomenon poses a fundamental challenge for practical LLM deployment, where models must acquire new skills without losing existing ones. Recent empirical studies (Luo et al., 2023; Wu et al., 2024) have documented severe forgetting during continual fine-tuning of LLMs, with standard supervised fine-tuning (SFT) often causing near-complete loss of prior task performance.

Reinforcement learning (RL) methods such as GRPO (Shao et al., 2024) and RLHF (Ouyang et al., 2022; Kaufmann et al., 2023) have shown promise in mitigating forgetting during LLM training. However, these approaches introduce substantial complexity: they require reward model training, policy gradient estimation, and careful hyperparameter tuning. Meanwhile, standard offline SFT—which trains on fixed teacher-generated targets—remains the dominant paradigm for task-specific adaptation due to its simplicity, yet it suffers from severe forgetting when applied sequentially.

Recent theoretical work (Chen et al., 2025; Shenfeld et al., 2025) suggests that RL's forgetting mitigation stems primarily from its use of *on-policy data*—training on samples generated by the current model—rather than from reward optimization itself. This insight motivates a simpler approach: can we achieve RL-like forgetting mitigation using only SFT on carefully selected on-policy targets? We propose **RazorSFT**, an on-policy SFT method that samples candidate responses from the current model, filters them through a task verifier, and selects the highest log-probability correct response

[1] https://gitlab.com/fars-a/razorsft-continual-onpolicy-sft

as the training target. This KL-minimal selection criterion ensures that training targets remain close to the current policy distribution, reducing the distributional shift that causes forgetting.

Our contributions are as follows:

- We propose RazorSFT, a simple on-policy SFT method that achieves strong forgetting mitigation without the complexity of RL-based approaches.
- Through ablation studies, we demonstrate that on-policy data is the primary contributor to forgetting mitigation, accounting for 76% of the improvement over offline SFT.
- We show that RazorSFT outperforms GRPO on average accuracy (0.616 vs. 0.515) and task adaptation (Countdown: 0.628 vs. 0.261) while achieving comparable forgetting mitigation.

2 Related Work

Continual Learning in Neural Networks. Catastrophic forgetting—the tendency of neural networks to lose previously learned knowledge when trained on new tasks—has been a central challenge in machine learning (van de Ven et al., 2024). Classical approaches include regularization-based methods such as Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2016), which penalizes changes to important parameters, and Learning without Forgetting (LwF) (Li & Hoiem, 2016), which uses knowledge distillation from the previous model. Architecture-based methods like Progressive Networks (Rusu et al., 2016) allocate new capacity for each task, while replay-based methods (Rolnick et al., 2018; van de Ven et al., 2020) maintain a buffer of past examples to interleave with new training data. Our work differs by leveraging on-policy data generation rather than explicit regularization or replay mechanisms.

Continual Learning in LLMs. Recent surveys (Wu et al., 2024) have highlighted the unique challenges of continual learning in large language models, where the scale of parameters and diversity of capabilities make traditional approaches difficult to apply. Empirical studies (Luo et al., 2023) have documented severe forgetting during continual fine-tuning, while benchmarks like TRACE (Wang et al., 2023) provide standardized evaluation protocols. Parameter-efficient fine-tuning (PEFT) methods such as LoRA (Hu et al., 2021) have been explored as a potential mitigation strategy (Liu et al., 2024), though they often trade off between plasticity and stability. Unlike these approaches, RazorSFT addresses forgetting through data selection rather than architectural constraints.

Reinforcement Learning for LLM Alignment. Reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Kaufmann et al., 2023) has become the dominant paradigm for aligning LLMs with human preferences. Direct Preference Optimization (DPO) (Rafailov et al., 2023) simplifies this by eliminating the need for a separate reward model, while GRPO (Shao et al., 2024) uses group relative rewards for mathematical reasoning. Rejection sampling methods such as RAFT (Dong et al., 2023) and RFT (Yuan et al., 2023) filter model-generated samples using reward signals. RazorSFT shares the use of on-policy data with these methods but replaces reward-based optimization with a simpler KL-minimal selection criterion, achieving comparable forgetting mitigation with reduced complexity.

On-Policy Learning and Forgetting. Recent theoretical and empirical work has begun to explain why on-policy methods mitigate forgetting. Chen et al. (2025) demonstrate that on-policy data naturally constrains gradient updates to remain close to the current policy distribution, while Shenfeld et al. (2025) provide theoretical analysis showing that online RL's forgetting mitigation stems from its on-policy nature rather than reward optimization. Concurrent work (Lai et al., 2025; Shenfeld et al., 2026) further supports this view. RazorSFT builds on these insights by isolating the on-policy data generation mechanism from RL's reward optimization, demonstrating that simple SFT on KL-minimal on-policy targets achieves strong forgetting mitigation without the complexity of policy gradient methods.

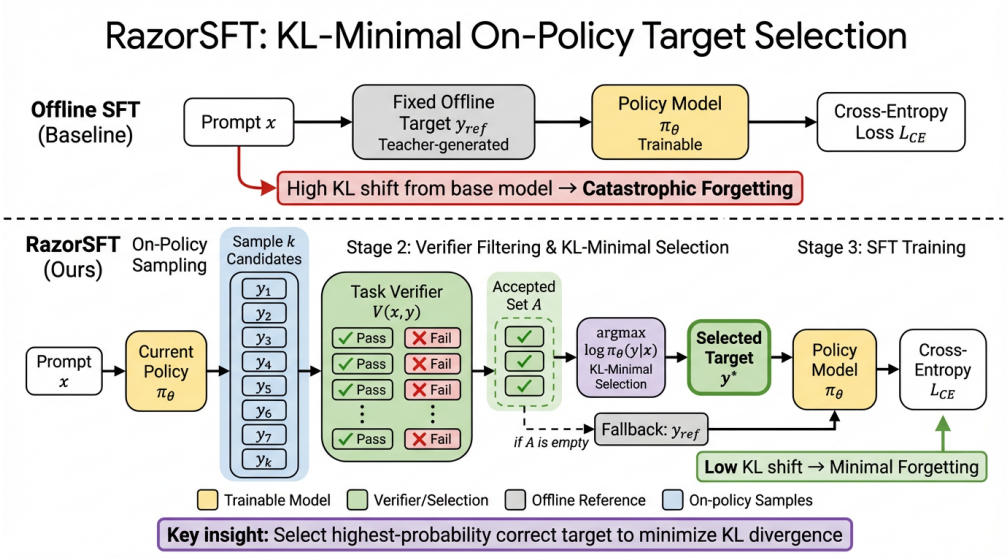

Figure 1: RazorSFT framework overview. **Top**: Offline SFT trains on fixed teacher-generated targets, causing high KL divergence from the base model and catastrophic forgetting. **Bottom**: RazorSFT samples k candidates from the current policy, filters through a task verifier, and selects the highest log-probability correct response as the training target, minimizing KL divergence and reducing forgetting.

3 METHOD

3.1 PROBLEM SETUP

We consider continual learning in large language models, where a model is sequentially fine-tuned on a series of tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T$. Each task $\mathcal{T}_t$ consists of a training dataset $\mathcal{D}_t = \{(x_i, y_i^{\text{ref}})\}$ where x_i is a prompt and y_i^{ref} is a reference target (e.g., teacher-generated or human-written). Training proceeds sequentially: after training on $\mathcal{T}_t$, the resulting checkpoint serves as the initialization for $\mathcal{T}_{t+1}$.

The central challenge is *catastrophic forgetting*: as the model adapts to new tasks, its performance on previously learned tasks degrades. We quantify this using two metrics. **Average Accuracy (AvgAcc)** measures the mean accuracy across all tasks after the final training stage: $\text{AvgAcc} = \frac{1}{T}\sum_{i=1}^{T} P_{T,i}$, where $P_{s,i}$ denotes accuracy on task i after training stage s. **Forgetting Measure (FM)** captures the average performance drop from peak to final: $\text{FM} = \frac{1}{T-1}\sum_{i=1}^{T-1}(P_{T,i} - \max_{k\in[i,T]} P_{k,i})$. FM is non-positive, with values closer to zero indicating less forgetting.

3.2 RAZORSFT: ON-POLICY TARGET SELECTION

Standard offline supervised fine-tuning (SFT) trains on fixed reference targets y^{ref}, which can induce large distribution shift from the current model and correlate with severe forgetting. We propose **RazorSFT**, an SFT-compatible target construction procedure that selects training targets that are both *correct* (verified by a task-specific checker) and *close to the current policy* (high probability under the model). Figure 1 illustrates the approach.

For each training prompt x_i at epoch e, RazorSFT proceeds in three stages. First, we sample k candidate responses from the current policy: $y_{i,1}, \ldots, y_{i,k} \sim \pi_{\theta_{e-1}}(\cdot \mid x_i)$. Second, we filter candidates through a task-specific verifier $V_t(x, y) \in \{0, 1\}$ to obtain the accepted set $\mathcal{A}_i = \{y_{i,j} : V_t(x_i, y_{i,j}) = 1\}$. Third, we select the training target using KL-minimal selection:

$$y_i^* = \begin{cases} \arg\max_{y\in\mathcal{A}_i} \log \pi_{\theta_{e-1}}(y \mid x_i) & \text{if } \mathcal{A}_i \neq \emptyset \\ y_i^{\text{ref}} & \text{otherwise (fallback)} \end{cases} \tag{1}$$

The fallback to offline references when no candidate passes verification ensures that every prompt contributes exactly one training target, maintaining consistent training intensity across methods. Training then proceeds with standard cross-entropy loss on the selected targets (x_i, y_i^*).

3.3 Theoretical Motivation

Recent work has established a connection between forgetting and the KL divergence induced by fine-tuning (Shenfeld et al., 2025; Chen et al., 2025). SFT minimizes the forward KL divergence $\mathrm{KL}[\pi^* \| \pi_\theta]$ from a target distribution π^* (defined by the training data) to the model, which can induce large distribution shift when π^* differs substantially from the current policy. In contrast, on-policy RL methods like GRPO implicitly bias toward solutions that achieve correctness with smaller KL shift from the initial model.

RazorSFT approximates this KL-minimal property within an SFT framework. By selecting the highest log-probability correct target from on-policy samples, we choose targets that the model already assigns high probability to, thereby minimizing the KL divergence $\mathrm{KL}[\delta_{y^*} \| \pi_\theta]$ between the training target and the current policy. This selection criterion can be viewed as finding the "easiest" correct solution for the model to learn, reducing the magnitude of parameter updates required and consequently limiting interference with previously learned capabilities.

Empirical evidence from Chen et al. (2025) demonstrates that on-policy data is the primary contributor to forgetting mitigation in RL methods, accounting for the majority of the benefit over offline SFT. Their analysis shows that neither KL regularization nor advantage estimation is necessary for reduced forgetting—the key factor is training on data generated by the current policy. RazorSFT operationalizes this insight by constructing on-policy training targets while maintaining the simplicity of SFT training.

4 Experiments

4.1 Experimental Setup

We evaluate RazorSFT on a 3-stage sequential curriculum designed to test continual learning across diverse task types. The curriculum consists of: (1) **IFEval** (Zhou et al., 2023), an instruction-following benchmark with rule-based verification; (2) **MMLU** (Hendrycks et al., 2020), a multiple-choice knowledge benchmark with exact-match verification; and (3) **Countdown**, an arithmetic equation construction task with programmatic validity checking. Each stage starts from the checkpoint produced by the previous stage.

We use Qwen2.5-7B-Instruct (Yang et al., 2024) as the base model and compare against four baselines: **Zero-shot** (no training), **Best-of-8** (verifier-selected best response at inference), **Offline SFT** (standard SFT on fixed teacher targets), and **GRPO** (Shao et al., 2024) (group relative policy optimization with verifier rewards). All training methods use full-parameter fine-tuning with FSDP, learning rate 1×10^{-5}, and 2 epochs per stage. For RazorSFT, we use $k = 8$ candidates for stages 1–2 and $k = 16$ for stage 3, with temperature 0.8–1.0. Results are averaged over 3 random seeds.

4.2 Main Results

Table 1 presents the main results. RazorSFT dramatically reduces catastrophic forgetting compared to Offline SFT, achieving FM of -0.039 versus -0.643 (60.4 percentage point improvement). While GRPO achieves slightly better FM (-0.023), RazorSFT substantially outperforms it on average accuracy (0.616 vs. 0.515, +10.1pp) and current-task adaptation (Countdown: 0.628 vs. 0.261, 2.4$\times$ improvement). The catastrophic nature of Offline SFT's forgetting is evident in the MMLU column, where accuracy drops to 0% after Stage 3, while RazorSFT retains 67.3% of the original performance. These results demonstrate that on-policy SFT with KL-minimal target selection can achieve forgetting mitigation comparable to RL methods while maintaining stronger task adaptation.

Table 1: Main results on 3-stage sequential curriculum (IFEval → MMLU → Countdown). RazorSFT achieves the best balance of average accuracy and forgetting mitigation among training methods. Best in **bold**, second-best underlined. FM = Forgetting Measure (higher is better, 0 = no forgetting).

Method	AvgAcc ↑	FM ↑	IFEval	MMLU	Countdown
Zero-shot	0.513	N/A	0.457	0.675	0.408
Best-of-8	**0.730**	N/A	0.581	**0.852**	**0.756**
Offline SFT	$0.121_{\pm 0.003}$	$-0.643_{\pm 0.001}$	$0.140_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.224_{\pm 0.009}$
GRPO	$0.515_{\pm 0.033}$	$-0.023_{\pm 0.017}$	$0.553_{\pm 0.039}$	$0.730_{\pm 0.007}$	$0.261_{\pm 0.092}$
RazorSFT-8	$0.616_{\pm 0.002}$	$\mathbf{-0.039_{\pm 0.007}}$	$\mathbf{0.546_{\pm 0.006}}$	$0.673_{\pm 0.009}$	$0.628_{\pm 0.010}$

Table 2: Ablation study isolating the contributions of on-policy data and KL-minimal selection. On-policy data accounts for 76% of the FM improvement over Offline SFT, while KL-minimal selection provides additional benefit.

Method	AvgAcc ↑	FM ↑	On-Policy?	KL-Minimal?
Offline SFT	0.119	−0.642	✗	✗
OnPolicy-SFT	0.260	−0.179	✓	✗
First-Accepted	0.598	−0.042	✓	✗
RazorSFT-8	**0.619**	**−0.032**	✓	✓

4.3 Ablation Study

Table 2 isolates the contributions of RazorSFT's two key components. Comparing OnPolicy-SFT (which trains on all correct on-policy samples without selection) to Offline SFT reveals that on-policy data alone improves FM from -0.642 to -0.179, accounting for 76% of the total FM improvement achieved by RazorSFT. This finding aligns with recent theoretical work suggesting that on-policy data naturally constrains gradient updates to remain close to the current policy (Chen et al., 2025; Shenfeld et al., 2025). The First-Accepted variant, which selects the first correct sample rather than the KL-minimal one, achieves FM of -0.042, while RazorSFT's KL-minimal selection further improves FM to -0.032 and AvgAcc from 0.598 to 0.619. These results demonstrate that while on-policy data is the primary contributor to forgetting mitigation, KL-minimal target selection provides meaningful additional benefit by explicitly minimizing distributional shift.

4.4 Forgetting Dynamics

Figure 2 visualizes the temporal dynamics of forgetting across training stages. Offline SFT exhibits catastrophic forgetting: IFEval accuracy drops sharply during Stage 2 (MMLU training) and MMLU accuracy collapses to near-zero during Stage 3 (Countdown training). In contrast, RazorSFT maintains relatively stable performance on previously learned tasks throughout training. GRPO shows intermediate behavior with better retention than Offline SFT but weaker task adaptation than RazorSFT, particularly evident in the Countdown panel where GRPO struggles to improve beyond the zero-shot baseline. These trajectories illustrate that RazorSFT's on-policy data generation creates a natural regularization effect that preserves prior knowledge while enabling effective learning on new tasks.

Figure 3 validates the mechanism underlying RazorSFT's effectiveness. Across all stages and epochs, the replacement rate—the fraction of samples where an on-policy target is selected over the offline reference—remains consistently high, ranging from 76% to 95%. This indicates that the model reliably generates correct solutions that are closer to its current distribution than the fixed teacher targets. The high replacement rates explain why RazorSFT achieves strong forgetting mitigation: by training predominantly on self-generated data, the model avoids the distributional shift that causes catastrophic forgetting in standard SFT. Notably, replacement rates remain high even in later epochs within each stage, suggesting that the model continues to benefit from on-policy data throughout training rather than converging to the offline distribution.

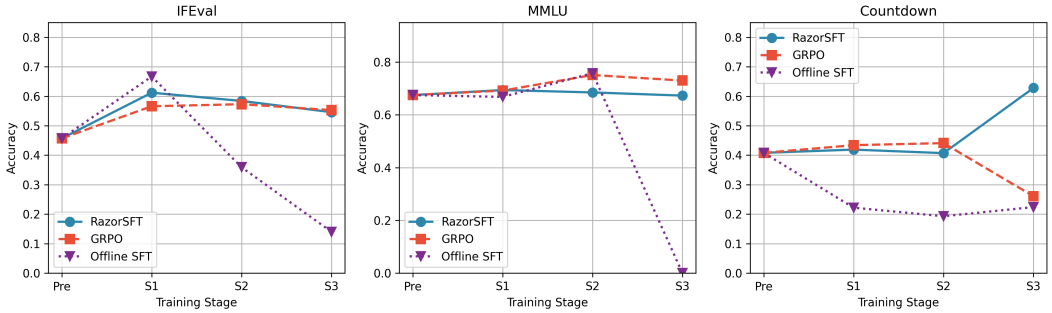

Figure 2: Per-task accuracy trajectories across training stages for RazorSFT, GRPO, and Offline SFT. Each panel shows how accuracy on a specific task evolves as training progresses through the 3-stage curriculum. RazorSFT maintains stable performance on previously learned tasks while achieving strong adaptation on new tasks.

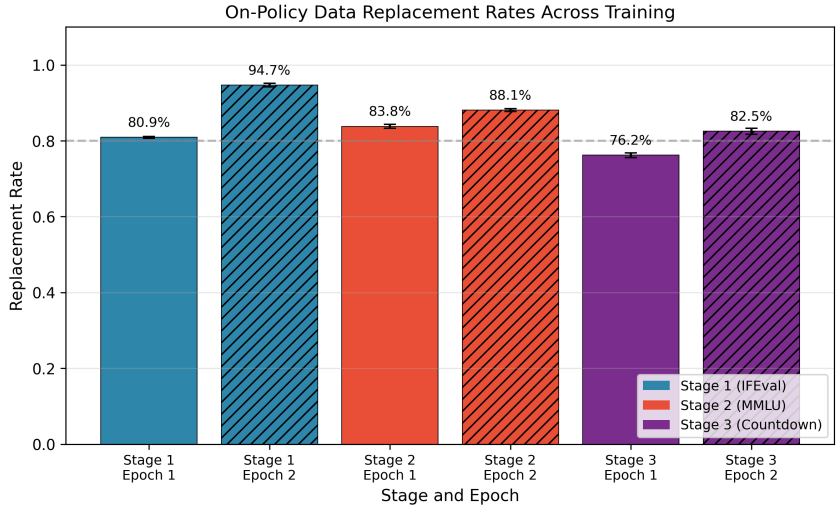

Figure 3: Replacement rates across training stages and epochs. The replacement rate measures the fraction of training samples where RazorSFT selects an on-policy generated target over the offline reference. High replacement rates (76–95%) indicate that the model consistently produces correct solutions that are preferred over fixed teacher targets.

5 Conclusion

We presented RazorSFT, a simple on-policy SFT method for continual learning in LLMs. By sampling candidates from the current policy, filtering through task verifiers, and selecting KL-minimal correct targets, RazorSFT achieves strong forgetting mitigation (FM -0.039 vs. -0.643 for offline SFT) while outperforming GRPO on average accuracy and task adaptation. Our ablation studies reveal that on-policy data accounts for 76% of the forgetting improvement, supporting recent theoretical insights that on-policy training—not reward optimization—is the key factor in RL's forgetting mitigation. RazorSFT demonstrates that the benefits of on-policy learning can be obtained within a simple SFT framework, offering a practical approach for continual LLM training without the complexity of RL methods.

References

Howard Chen, Noam Razin, Karthik R. Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. *ArXiv*, abs/2510.18874, 2025.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and T. Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *ArXiv*, abs/2304.06767, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300, 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *ArXiv*, abs/2312.14925, 2023.

J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2016.

Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Min Xie, Qingfu Zhang, Hongbin Liu, Gaofeng Meng, and Fei Zhu. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training. *ArXiv*, abs/2507.05386, 2025.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2935–2947, 2016.

Jingren Liu, Zhong Ji, Yunlong Yu, Jiale Cao, Yanwei Pang, Jungong Han, and Xuelong Li. Parameter-efficient fine-tuning for continual learning: A neural tangent kernel perspective. *ArXiv*, abs/2407.17120, 2024.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2023.

Long Ouyang, Jeff 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 E. Miller, M. Simens, Amanda Askell, Peter Welinder, P. Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022.

Rafael Rafailov, Archit Sharma, E. Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290, 2023.

David Rolnick, Arun Ahuja, Jonathan Schwarz, T. Lillicrap, and Greg Wayne. Experience replay for continual learning. pp. 348–358, 2018.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, J. Kirkpatrick, K. Kavukcuoglu, Razvan Pascanu, and R. Hadsell. Progressive neural networks. *ArXiv*, abs/1606.04671, 2016.

Zhihong Shao, Peiyi Wang, Qihao Zhu, R. Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300, 2024.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *ArXiv*, abs/2509.04259, 2025.

Idan Shenfeld, Mehul Damani, Jonas Hubotter, and Pulkit Agrawal. Self-distillation enables continual learning. 2026.

Gido M. van de Ven, H. Siegelmann, and A. Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, 11, 2020.

Gido M. van de Ven, Nicholas Soures, and D. Kudithipudi. Continual learning and catastrophic forgetting. *ArXiv*, abs/2403.05175, 2024.

Xiao Wang, Yuan Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. Trace: A comprehensive benchmark for continual learning in large language models. *ArXiv*, abs/2310.06762, 2023.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. Continual learning for large language models: A survey. *ArXiv*, abs/2402.01364, 2024.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.

Zheng Yuan, Hongyi Yuan, Cheng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. Scaling relationship on learning mathematical reasoning with large language models. *ArXiv*, abs/2308.01825, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *ArXiv*, abs/2311.07911, 2023.