Large Language Model Training and Reinforcement Learning

Miquel Noguer i Alonso
Artificial Intelligence Finance Institute

March 26, 2025

Abstract

Large Language Models (LLMs) have revolutionized natural language processing, but optimizing these models to align with human preferences remains challenging. This paper provides a comprehensive review of reinforcement learning (RL) techniques used in LLM training pipelines with a focus on policy optimization methods. We survey the evolution from traditional supervised fine-tuning to more sophisticated approaches including Reinforcement Learning from Human Feedback (RLHF), Proximal Policy Optimization (PPO), Direct Preference Optimization (DPO), and Group Relative Policy Optimization (GRPO). We analyze the theoretical foundations, implementation challenges, and empirical performance of each method while highlighting their respective trade-offs in terms of computational efficiency, sample complexity, and alignment quality. Our analysis reveals that while RLHF with PPO remains the industry standard, newer methods like DPO and GRPO offer promising alternatives with reduced computational requirements and comparable performance. We conclude with directions for future research, emphasizing the importance of developing more efficient alignment techniques as models continue to scale in size and capability.

1 Introduction

Large language models (LLMs) have transformed natural language processing, demonstrating remarkable capabilities across diverse tasks Brown et al. [2020], Wei et al. [2022]. The development of these models follows a two-phase approach: pre-training on vast text corpora followed by post-training refinement to align with human preferences and improve task performance Ouyang et al. [2022]. While pre-training methodologies have received considerable attention Vaswani et al. [2017], Radford et al. [2019], post-training techniques—particularly those leveraging reinforcement learning—have emerged as crucial for addressing alignment challenges Christiano et al. [2017], Bai et al. [2023].

This paper provides a systematic review of reinforcement learning approaches for LLM policy optimization, analyzing the progression from supervised fine-tuning (SFT) to more sophisticated techniques including reinforcement learning from human feedback (RLHF) Ouyang et al. [2022], proximal policy optimization (PPO) Schulman et al. [2017], direct preference optimization (DPO) Rafailov et al. [2023], and group relative policy optimization (GRPO) AI [2024]. For each method, we examine theoretical foundations, practical implementation challenges, and empirical performance, highlighting trade-offs in computational efficiency, sample complexity, and alignment quality.

Our contribution is threefold: (1) a comprehensive review of state-of-the-art RL methods for LLM alignment, (2) a comparative analysis of their theoretical properties and practical trade-offs, and (3) directions for future research in efficient alignment techniques as models continue to scale in size and capability.

2 Modern LLM Training Pipeline

The training pipeline for modern LLMs splits into two phases:

1. **Pre-training:** Learning general language patterns from vast, unlabeled text Radford et al. [2019], Brown et al. [2020].
2. **Post-training (Fine-tuning):** Specializing the model for tasks or preferences Ouyang et al. [2022], Touvron et al. [2023].

Current development trends emphasize post-training methodologies over pre-training innovations, with scaling laws guiding the efficient allocation of computational resources Kaplan et al. [2020], Hoffmann et al. [2022]. This shift reflects the increasing focus on making LLMs more helpful, harmless, and honest rather than simply more capable at next-token prediction Bai et al. [2022].

2.1 Pre-training

Pre-training exposes an LLM to a massive, unlabeled dataset (e.g., books, articles, websites) to build a foundational understanding of language Radford et al. [2019]. The objective is *next-token prediction*:

$$\mathcal{L}_{\text{pre-train}} = -\sum_{t=1}^{T} \log P(w_t | w_{1:t-1}; \theta),$$

where w_t is the t-th token, $w_{1:t-1}$ is the preceding sequence, and θ are model parameters. The goal is to maximize the likelihood of predicting the next token, enabling the model to capture linguistic patterns and semantics Brown et al. [2020].

Pre-training has evolved significantly since the introduction of transformer architectures Vaswani et al. [2017], with innovations in scaling laws Kaplan et al. [2020], mixture-of-experts approaches Xu et al. [2023], and efficient attention mechanisms enhancing model capabilities while managing computational costs.

Example: Given the sequence "The cat sat on the," the model predicts "mat" with high probability based on patterns in the training corpus.

2.2 Post-training

Post-training refines the model for specific tasks or human preferences using techniques like instruction tuning and RLHF Ouyang et al. [2022], Bai et al. [2022]. Recent methods include SFT and RLHF, though innovations like DeepSeek R1 challenge the need for extensive human-labeled data AI [2024].

The post-training landscape has diversified to include approaches like:

- **Instruction Tuning:** Training models to follow natural language instructions Zhou et al. [2024].
- **Constitutional AI:** Using principle-based guidance for model behavior Bai et al. [2023].
- **Reinforcement Learning from AI Feedback (RLAIF):** Using more capable AI systems to provide training signals Lee et al. [2023].

Example: For the prompt "Is 2 + 2 = 5?", the pre-trained model might output a probabilistic response, but post-training ensures it answers "No" with reasoning: "2 + 2 equals 4, not 5."

3 OpenAI's Post-training Framework

OpenAI's influential RLHF framework Ouyang et al. [2022] established a three-step process that has become an industry template:

1. **SFT:** Train on human demonstrations. Prompt: "Explain the moon landing to a six-year-old." Human response: "A big rocket took people to the moon, and they walked on it!" This labeled data fine-tunes the model Zhou et al. [2024].
2. **Reward Model Training:** Generate multiple outputs for a prompt (e.g., "A: It's when astronauts landed..." vs. "B: Moon's a rock..."), rank them (A ¿ B), and train a reward model $r_\phi(y|x)$ to predict human preferences Zheng et al. [2023].
3. **Policy Optimization:** Optimize the policy π_θ using RL (e.g., PPO) against the reward model Schulman et al. [2017].

This approach has produced leading models like GPT-4 Gao et al. [2023] but comes with significant computational costs and implementation complexity, motivating the development of more efficient alternatives.

Algorithm 1 OpenAI RLHF Pipeline

procedure RLHF($\pi_{\text{pretrained}}$, D_{SFT}, $D_{\text{preferences}}$)
    $\pi_{\text{SFT}} \leftarrow \text{SupervisedFineTune}(\pi_{\text{pretrained}}, D_{\text{SFT}})$
    $r_\phi \leftarrow \text{TrainRewardModel}(\pi_{\text{SFT}}, D_{\text{preferences}})$
    $\pi_{\text{RLHF}} \leftarrow \text{PPO}(\pi_{\text{SFT}}, r_\phi)$
    return π_{RLHF}
end procedure

4 Supervised Fine-tuning (SFT)

SFT adapts a pre-trained LLM to a specific task using a labeled dataset Zhou et al. [2024]. The objective remains next-token prediction:

$$\mathcal{L}_{\text{SFT}} = -\sum_{i=1}^{N} \log P(y_i|x_i;\theta),$$

where (x_i, y_i) are prompt-response pairs, and N is the dataset size.

Differences from Pre-training:

- **Labeled Data:** SFT uses curated input-output pairs Kopf et al. [2023].
- **Task-specificity:** Tailors the model for particular applications Ding et al. [2023].
- **Learning Rate:** Typically uses smaller learning rates to avoid catastrophic forgetting.

Recent research suggests that high-quality SFT with a few thousand carefully selected examples can achieve strong performance without extensive RLHF Zhou et al. [2024], challenging assumptions about the necessity of complex alignment techniques.

Example: Prompt: "Write a poem about rain." Human response: "Drops fall soft, skies weep, / Puddles dance beneath the deep." The model learns to mimic this style.

5 Reinforcement Learning (RL)

RL trains an agent to maximize cumulative reward by interacting with an environment Schulman et al. [2015a]. Unlike supervised learning, it explores optimal actions via trial and error, learning policies that generalize to new situations.

5.1 Key RL Concepts

- **Policy** $\pi(a|s)$: Maps states to actions (e.g., a neural network).
- **State** s: Describes the environment (e.g., in text generation, the prompt plus tokens generated so far).
- **Action** a: Chosen stochastically from $\pi(a|s)$ (e.g., the next token).
- **Reward** $r(s, a)$: Feedback (e.g., alignment with human preferences).
- **Value** $V^\pi(s)$: Expected cumulative reward from state s under policy π:

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s \right],$$

  where $\gamma \in [0, 1)$ is the discount factor Schulman et al. [2015b].
- **Advantage** $A(s, a) = Q(s, a) - V(s)$: The relative benefit of taking action a in state s compared to the average action.

Example (LLM Context): State s: Prompt "Write a story about a brave knight" plus tokens generated so far. Action a: Generate the next token "dragon". Reward r: High if this leads to an engaging narrative, low if it creates inconsistencies.

6 Policy Optimization Algorithms

6.1 Policy Gradients

Policy gradients maximize expected reward $J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)]$ over trajectories:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) R(\tau)\right].$$

The gradient is approximated via sampling:

$$\nabla_\theta J(\theta) \approx \frac{1}{|\mathcal{D}|}\sum_{\tau\in\mathcal{D}}\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) R(\tau).$$

While simple policy gradients provide a fundamental approach, they suffer from high variance and sample inefficiency, leading to the development of more sophisticated methods Schulman et al. [2015a].

6.2 Proximal Policy Optimization (PPO)

PPO Schulman et al. [2017] balances stability and efficiency in policy updates. Components:

- **Policy Model** π_θ: Generates text; in LLMs, this is the language model being optimized.
- **Value Model** $V_\phi(s)$: Critic estimating state value.
- **Reward Model** $r_\psi(y|x)$: Frozen model that scores outputs based on learned human preferences.
- **Reference Model** π_{ref}: Frozen SFT model that enforces stability via KL-divergence penalty.

Objective:

$$L^{\text{PPO}}(\theta) = \mathbb{E}\left[\min\left(\frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}A(s,a), \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}, 1-\epsilon, 1+\epsilon\right)A(s,a)\right)\right] - \beta D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}) + \eta H(\pi_\theta),$$

where $A(s,a)$ is the advantage, ϵ clips updates to prevent extreme policy changes, D_{KL} penalizes deviation from the reference model, and H is entropy regularization.

The clipping term ensures that the ratio $\frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}$ cannot move outside the range $[1-\epsilon, 1+\epsilon]$, providing stability while allowing learning.

PPO implementation in LLMs faces several challenges:

- **Reward Hacking:** Models may find unexpected ways to maximize reward that don't align with human intent.
- **KL Collapse:** Without proper regularization, the policy can collapse to a narrow distribution.
- **Compute Requirements:** RLHF with PPO demands significant computational resources.

Example: Prompt: "Write a frog story." Output: "Frogs hopped happily." Reward: +0.8. PPO updates the policy to increase the probability of generating similar high-reward outputs while maintaining linguistic quality through KL regularization.

6.3 Group Relative Policy Optimization (GRPO)

GRPO AI [2024] simplifies PPO by eliminating the value model. It estimates average rewards via group sampling (e.g., 16 trajectories) and optimizes:

$$L^{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}(r(y|x) - \bar{r})\right],$$

where $\bar{r}$ is the sampled mean reward.

This approach reduces computational overhead by:

- Removing the need to train a separate value function

- Using batch statistics as a baseline instead of learned value estimates
- Simplifying implementation while maintaining performance

DeepSeek R1 demonstrated that this simplified approach achieves comparable or better results than full PPO implementation with significantly reduced computational requirements.

Example: For the prompt "Summarize DeepSeek's approach," GRPO samples 16 summaries, computes rewards, and updates π_θ based on the relative performance of each summary.

6.4 Direct Preference Optimization (DPO)

DPO Rafailov et al. [2023] bypasses reward model training, optimizing directly from preference pairs (y^+, y^-):

$$L^{\text{DPO}}(\theta) = -\mathbb{E}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \beta \log \frac{\pi_\theta(y^-|x)}{\pi_{\text{ref}}(y^-|x)}\right)\right],$$

where σ is the sigmoid function, and β controls preference strength.

DPO's elegance comes from a theoretical connection to reward modeling: it implicitly assumes a reward model of the form:

$$r_\beta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)},$$

and then optimizes the policy to maximize the probability of preferred outputs under this implicit reward.

Advantages of DPO include:

- No need to train a separate reward model
- Single-stage training from preference data
- No complex RL optimization
- Reduced computational requirements

DPO has inspired several variants, including:

- **Identity Preference Optimization (IPO)** Liu et al. [2023a]: Using the identity function instead of the sigmoid.
- **Robust DPO** Liu et al. [2023b]: Adapting DPO to handle noisy preference data.
- **Best-of-N DPO** Tunstall et al. [2023]: Extending DPO to work with ranking data beyond simple pairs.

Example: Prompt: "Explain RL." Preferred y^+: "RL is learning from rewards." Dispreferred y^-: "RL is random." DPO increases $P(y^+)$ and decreases $P(y^-)$.

7 Empirical Comparisons

Method	Compute	Data Efficiency	Implementation	Performance
SFT	Low	Medium	Simple	Baseline
RLHF (PPO)	Very High	Low	Complex	Strong
GRPO	High	Medium	Moderate	Strong
DPO	Medium	High	Simple	Strong

Table 1: Comparative analysis of policy optimization methods.

Recent benchmarks Zheng et al. [2023], Cobbe et al. [2021] suggest that while RLHF with PPO remains the industry standard for top-performing models, newer methods like DPO and GRPO offer compelling alternatives with comparable performance at reduced computational cost. Notably, DPO has demonstrated strong performance in direct comparisons Rafailov et al. [2023], while GRPO offers a middle ground between PPO's complexity and DPO's simplicity AI [2024].

The choice of method depends on several factors:

- **Available Compute:** Organizations with limited resources may prefer DPO or GRPO.
- **Data Quality:** High-quality preference data favors DPO, while noisy data might benefit from PPO's robust optimization.
- **Model Size:** Larger models increase the computational advantages of simplified methods.

8 Why Use RL in Post-training?

SFT struggles with complex, dynamic tasks. RL excels because:

- Limited SFT data is insufficient for nuanced behavior Ouyang et al. [2022].
- RL adapts to preferences and environments efficiently Christiano et al. [2017].
- RL incorporates negative feedback and supports online learning Bai et al. [2022].

However, recent work suggests that with sufficient high-quality data, methods like LIMA Zhou et al. [2024] can achieve strong performance with SFT alone, raising questions about the necessity of complex RL pipelines for all applications.

9 End-to-End Post-training Example

To illustrate the full pipeline:

1. **SFT:** Pre-trained GPT-3. Prompt: "What is RL?" Human: "RL is learning from rewards." Fine-tune with $\mathcal{L}_{\text{SFT}}$.
2. **Reward Model:** Prompt: "Explain SFT." Outputs: (1) "SFT adapts models with labeled data..." (best), (2) "SFT is training..." (okay), (3) "SFT is about dogs" (worst). Train r_ψ with rankings.
3. **PPO:** Prompt: "Summarize DeepSeek's advantage." Policy output: "DeepSeek uses GRPO for efficiency." Reward: +0.9. Update with PPO objective.

10 Future Directions

Several promising research directions are emerging:

- **Efficient Alignment:** Methods like Forward-Forward Wu et al. [2023] aim to reduce the computational burden of alignment.
- **Combined Approaches:** Hybrid methods leveraging the strengths of both RLHF and DPO Liu et al. [2023b].
- **Synthetic Feedback:** Using AI-generated feedback to scale preference data Lee et al. [2023], Hu et al. [2023].
- **Causal Alignment:** Incorporating causal reasoning to improve robustness to distribution shifts.
- **Online Learning:** Continuous adaptation to evolving user preferences and requirements.

11 Conclusion

This paper has surveyed the landscape of reinforcement learning approaches for LLM policy optimization, tracing the evolution from SFT to sophisticated methods like PPO, DPO, and GRPO. Our analysis highlights the trade-offs between computational efficiency, sample complexity, and alignment quality across these methods.

The field is witnessing a rapid evolution in alignment techniques, with recent innovations challenging the necessity of computationally intensive RLHF pipelines. Direct preference methods like DPO

and simplified RL approaches like GRPO offer promising alternatives that maintain performance while reducing implementation complexity and computational requirements.

As language models continue to scale in size and capability, the development of more efficient alignment techniques becomes increasingly important. Future research will likely focus on reducing the computational and data requirements of alignment while improving the robustness and generalizability of aligned behaviors.

Ultimately, the goal remains consistent: to develop LLMs that not only demonstrate impressive capabilities but also reliably align with human values and intentions across diverse contexts. The methods reviewed in this paper represent significant progress toward this goal, but substantial challenges remain in achieving robust, efficient, and scalable alignment.

References

DeepSeek AI. Deepseek r1: Leveraging group relative policy optimization for efficient alignment. *arXiv preprint arXiv:2401.10020*, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ning Ding, Yulin Chen, Bokai Xu, Shengding Hao, Ziyue Liu, Zhiyuan Liu, Minlie Wang, Weizhu Shi, Maosong Sun, and Ming Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

Leo Gao, John Schulman, and Jacob Hilton. Scaling supervised learning for natural language generation: From gpt-3 to chatgpt and claude. *Communications of the ACM*, 66(12):66–79, 2023.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Edward J Hu, Mikel Artetxe, Loïc Barrault, Swaroop Bhat, Clare Arrington Chang, Clark Chen, Marta Clinciu, Vedanuj East, David Grangier, Siddharth Karamcheti, et al. Aligning large language models through synthetic feedback. *arXiv preprint arXiv:2305.13735*, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Clark, Christopher Berner, Jack Hesse, Heewoo Chen, et al. Scaling laws for neural language models. In *International Conference on Machine Learning*, pages 5185–5194. PMLR, 2020.

Andreas Kopf, Yannic Kilcher, Leandro von Werra, Nelson Elhage, Dominic Jones, Papa Ndiour, Thomas Wolf, Gasper Beguš, Johannes von Oswald, Nora Hollmann, et al. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*, 2023.

Harrison Lee, Xiuyu Gu, Mikayel Samvelyan, Stephen James, Yuqing Hu, Nicolas Usunier, Angeliki Lazaridou, Daniel Strouse, Jose Hernandez-Orallo, Shane Legg, et al. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Xiao Liu, Dan Hendrycks, Marvin Mazeika, Zhengbao Zhou, Mohit Bansal, Jared Kaplan, Dan Roth, and Aditya Grover. Reward rationalisation: Aligning large language models through reflection and reward maximisation. In *International Conference on Learning Representations*, 2023a.

Zhengxuan Liu, Michael Ruan, Jiawei Shang, Xinrui Li, Harikrishna Balakrishnan, William Wang, Xuezhi Cao, Joseph Zheng, Francisco JR Ruiz, Thomas Scialom, et al. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023b.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. *International conference on machine learning*, pages 1889–1897, 2015a.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Lewis Tunstall, Nathan Lambert, Nazneen Rajani, Tahmid Mohiuddin, Ellie Pavlick, Tim Dettmers, Yonatan Bisk, Ari Cooper Stickland, and Vladimir Karpukhin. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Jerry Tworek Colin Raffel Wu, Nova DasSarma, Alethea Geiger, Vineet Kosaraju, Romi Fang, Sanjana Gopinath, Sunil Mistele, Kamilė Krishnan, Loic Barrault, and Robin Jia. Fine-tuning language models with just forward passes. *arXiv preprint arXiv:2312.11508*, 2023.

Yiming Xu, Xin Sun, Cheng Cheng, Yihang Xiong, Wenhao Wang, Yuxin Ding, Wenhui Cao, Lizhi Wang, Fei Xu, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Li, Dacheng Li, Joseph Gonzalez, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yu Meng, Hannaneh Hajishirzi, Xuezhe Lee, and Mike Lewis. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*, 2024.