

Extended Abstract

Motivation Aligning large language models (LLMs) with human preferences is critical for producing helpful, truthful, and safe outputs. While supervised fine-tuning (SFT) establishes a strong foundation, it often fails to optimize for nuanced preferences present in open-ended tasks. Reinforcement learning from human feedback (RLHF) addresses this gap but is costly and complex. Recent alternatives like Direct Preference Optimization (DPO) offer a scalable middle ground. However, scaling DPO is still bottlenecked by the availability of high-quality preference data. We investigate whether synthetic supervision from frozen reward models can bridge this gap.

Method We construct a three-stage alignment pipeline. First, we fine-tune a pretrained language model on the SmolTalk dataset to encourage instruction-following behavior. Next, we apply DPO using UltraFeedback, which is a large collection of human-annotated preference pairs, to align the model with user preferences. Finally, we propose a synthetic extension: generating model completions, scoring them using Nemotron-70B (a frozen reward model), and using these scores to construct new preference tuples for further fine-tuning. We evaluate each stage quantitatively and qualitatively.

Implementation We use parameter-efficient fine-tuning (LoRA) for both DPO and the synthetic extension stages to reduce memory footprint and support modular experimentation. All training was performed using three NVIDIA RTX 4090 GPUs with mixed-precision and gradient checkpointing. The DPO model was trained on binarized UltraFeedback preference tuples, while the synthetic model used 400 bootstrapped tuples derived from Nemotron scoring. Each fine-tuning stage used consistent hyperparameters for comparability.

Results Each stage in the pipeline improves model alignment. Pairwise evaluation using Nemotron reward scores shows DPO outperforming SFT (52.5% win rate), and the synthetic extension outperforming DPO (53.7%). Iterative bootstrapping further improves results over longer training on static data. Qualitative evaluation reveals better logical structure and factual correctness in models trained with synthetic preferences, correcting major reasoning failures present in earlier stages.

Discussion Our findings show that synthetic feedback, when sourced from strong frozen reward models, can meaningfully improve alignment at low cost. Unlike traditional RLHF pipelines, our approach avoids explicit reward model training and leverages modular LoRA adapters to enable lightweight preference refinement. Iterative reward-space bootstrapping yields additional gains, demonstrating that new model generations over fixed prompts can continue to explore improved response regions.

Conclusion We demonstrate that alignment through synthetic supervision is feasible, scalable, and effective. By combining SFT, DPO, and synthetic bootstrapping via a frozen reward model, we achieve consistent improvements in preference alignment. This modular pipeline offers a practical alternative to full RLHF and sets the stage for further work in reward-model-driven self-improvement.

RL fine-tuning of Language Models with Synthetic Data

Daniel Sorvisto

Institute of Computational and Mathematical Engineering
Stanford University
sorvisto@stanford.edu

Abstract

We explore a multi-stage pipeline for aligning language models to human preferences using both supervised and synthetic data. Starting with supervised fine-tuning (SFT) on the SmolTalk dataset, we build a strong instruction-following base model. We then apply Direct Preference Optimization (DPO) using the UltraFeedback dataset to align the model with human-annotated preferences. Finally, we introduce a synthetic extension method that uses a frozen reward model (Nemotron-70B) to score model-generated responses, creating new preference data without human labeling. This synthetic bootstrapping approach yields consistent improvements over SFT and DPO baselines, both in win rate metrics and qualitative output quality. Our findings demonstrate that reward-model-driven supervision can scale preference alignment efficiently without reinforcement learning or additional human feedback.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance across a range of natural language understanding and generation tasks. While base models trained on massive corpora exhibit impressive zero-shot capabilities, fine-tuning remains essential for adapting models to downstream applications and for aligning their outputs with human preferences. Supervised fine-tuning (SFT), in particular, provides an effective foundation by training models on high-quality input-output pairs, often derived from human-written demonstrations or instruction-following datasets.

However, SFT alone does not guarantee alignment with human preferences. Models trained purely via SFT tend to mimic behavior from their training set and may not always produce responses that are preferred by users. This limitation has motivated the development of reinforcement learning-based fine-tuning methods, such as Reinforcement Learning from Human Feedback (RLHF) Ouyang et al. (2022), which further refine model behavior using preference data. While RL methods can improve alignment, they are computationally intensive and complex to implement. Direct Preference Optimization (DPO) Rafailov et al. (2023), a recently proposed method, simplifies this process by leveraging pairwise human preferences in a supervised learning framework, avoiding the need for full RL.

In this project, we first perform SFT using the **SmolTalk** Mu et al. (2023) dataset, a collection of synthetic dialogues designed to teach conversational skills to language models in a structured, lightweight manner. This serves as our base model for further experimentation. We then explore alignment using the **UltraFeedback** Yuan et al. (2023) dataset, which consists of preference comparisons over long-form model completions on a wide range of queries. Using this data, we train a DPO model to better capture human preferences.

Finally, we propose an extension based on **synthetic preference data** generation. Specifically, we use our existing DPO model to generate candidate responses to prompts, and then score these responses

using the *Nemotron-70B Reward Model* NVIDIA (2024). This approach allows us to automatically generate high-quality preference data without relying on costly human annotations. We fine-tune a second layer of LoRA adapters Hu et al. (2021) on top of the DPO model using this synthetic supervision, and compare its performance against the original models.

Our experiments aim to understand the tradeoffs and potential of preference-based alignment through both human-annotated and synthetic signals.

2 Related Work

Supervised Fine-Tuning (SFT) remains the foundational method for adapting large language models (LLMs) to downstream tasks. Datasets like *SmolTalk* Mu et al. (2023) provide structured dialogues for instruction tuning and serve as a lightweight alternative to massive instruction datasets. While SFT enables models to follow task instructions and produce syntactically fluent outputs, it does not directly optimize for human preference or utility, leading to suboptimal alignment.

Direct Preference Optimization (DPO) Rafailov et al. (2023) builds on SFT by incorporating pairwise human preference data in a contrastive loss framework. This allows models to learn not just what to say, but which responses are preferred when multiple valid outputs exist. DPO sidesteps the instability of reinforcement learning algorithms like PPO, offering a simpler and more scalable alignment method. Our baseline uses SFT followed by DPO, forming a strong aligned model trained on the UltraFeedback dataset Yuan et al. (2023).

Synthetic Preference Supervision. A key challenge in scaling DPO-style alignment methods is the high cost of collecting large volumes of preference data. Inspired by recent work such as Reinforcement Learning from AI Feedback (RLAIF) Lee et al. (2024), we explore a different path: using frozen reward models to evaluate and generate new preference data. Unlike RLAIF, which trains a reward model from AI preferences, our method directly applies a strong frozen reward model (Nemotron-70B NVIDIA (2024)) to score completions. This enables a single-stage synthetic reward pipeline that bypasses the need to fit a new reward function. Similar in spirit to Constitutional AI Bai et al. (2022), which uses model self-critiques as supervision, our approach uses model generations evaluated by a reward model to synthesize learning signals except we skip the critique step entirely in favor of scalar feedback.

Low-Rank Adaptation. Our extension fine-tunes a second layer of LoRA adapters Hu et al. (2021) on top of the DPO-aligned model using this synthetic reward data. This allows for a lightweight and modular approach to preference refinement, making it possible to stack alignment stages without retraining the full base model or discarding previous training.

Our extension complements previous alignment methods by offering a scalable way to bootstrap further preference training without additional human supervision or reward model retraining.

3 Datasets

3.1 SmolTalk Dataset

SmolTalk is a large-scale, synthetic instruction-tuning dataset designed to improve language model performance, particularly for smaller models. Curated and released by the SmolLM2 team at Hugging Face, it includes over 1 million samples generated via GPT-4 and other open data sources. SmolTalk reflects a data-centric approach where synthetic high-quality samples can rival curated human-created data, making it a strong base for SFT in downstream reinforcement learning pipelines.

3.2 UltraFeedback

UltraFeedback Yuan et al. (2023) is a dataset for training and evaluating language models using AI-generated preference signals. It includes approximately 64,000 diverse instruction-following prompts drawn from sources like ShareGPT, FLAN, TruthfulQA, and Evol-Instruct. For each prompt, four responses are generated using different language models, and GPT-4 is used to annotate each response along multiple axes, including helpfulness, factuality, honesty, and instruction adherence. These annotations often include chain-of-thought justifications, providing rich feedback beyond simple scalar scores.

The dataset was specifically curated to support preference-based learning. A binarized version of UltraFeedback is constructed by selecting, for each prompt, the highest-rated response (based on the GPT-4 aggregate score) as the “chosen” sample and pairing it with a randomly selected lower-rated “rejected” response. This results in a large number of high-quality $(x, y_{\text{chosen}}, y_{\text{rejected}})$ tuples, making UltraFeedback ideal for training with Direct Preference Optimization (DPO). The dataset’s consistent formatting and reliability make it a strong benchmark for evaluating preference-aligned models.

4 Methods

4.1 Supervised Fine-Tuning (SFT)

Supervised Fine-Tuning (SFT) is the foundational step in our alignment pipeline. Starting from a pretrained language model, we fine-tune it on curated input-output pairs to induce task-following behavior and improve response quality. In our case, we use the **SmolTalk** dataset Mu et al. (2023), which consists of synthetic dialogues covering a range of conversational intents. The dataset emphasizes clean instruction-following examples without requiring large-scale human annotations.

Given a prompt-response pair (x, y) , the model is trained to maximize the log-likelihood of the target response y conditioned on the input x . This corresponds to minimizing the standard cross-entropy loss:

$$\mathcal{L}_{\text{SFT}} = - \sum_{t=1}^T \log p_{\theta}(y_t \mid y_{<t}, x)$$

where p_{θ} is the model’s conditional probability distribution over tokens, and T is the length of the response.

We initialize from a base pretrained model and fine-tune using the Hugging Face Transformers library. The model parameters are updated using AdamW, and we use standard token-level teacher forcing, feeding the ground-truth tokens at each time step. This stage provides a strong initialization for subsequent preference-based alignment.

4.2 Direct Preference Optimization (DPO)

After SFT, we align the model further using Direct Preference Optimization (DPO) Rafailov et al. (2023), which trains on pairwise human preference data. The key idea in DPO is to bypass reinforcement learning algorithms like PPO and instead use a contrastive loss that directly encourages the model to assign higher likelihoods to preferred completions.

Given a prompt x and two responses y_{chosen} and y_{rejected} , DPO trains the model to increase the log-probability of y_{chosen} relative to y_{rejected} . The loss is defined as:

$$\mathcal{L}_{\text{DPO}} = - \log \sigma \left(\beta \left[(\log \pi_{\theta}(y_{\text{chosen}}|x) - \log \pi_{\text{ref}}(y_{\text{chosen}}|x)) - (\log \pi_{\theta}(y_{\text{rejected}}|x) - \log \pi_{\text{ref}}(y_{\text{rejected}}|x)) \right] \right)$$

where π_{θ} is the fine-tuned model, σ is the sigmoid function, and β is a temperature parameter controlling the sharpness of the preference contrast. We use $\beta = 0.1$ in our experiments, following the original DPO paper.

Unlike RL-based methods, DPO avoids the need for explicit reward modeling or advantage estimation. Instead, it relies on implicit feedback encoded in the preference pairs. We apply DPO on top of our SFT model using the UltraFeedback dataset Yuan et al. (2023), which contains high-quality human preferences over alternative LLM responses across a range of open-ended queries.

Training is done using LoRA adapters on top of the SFT model, allowing us to retain the original weights and make preference alignment modular. This stage results in a DPO-aligned model that performs better on preference-sensitive tasks while retaining fluency and generality from the SFT base.

4.3 Synthetic Data Extension via Frozen Reward Model

To further improve alignment without additional human supervision, we introduce a synthetic data extension stage. The core idea is to use a strong, frozen reward model (Nemotron-70B-Reward NVIDIA (2024)) to score candidate completions generated by our DPO-aligned model, and use these scores as supervision signals for a new round of preference learning. This approach builds on ideas from RLAIIF Lee et al. (2024), but bypasses the reward model training step entirely.

We generate responses from the current model using a fixed set of prompts (sampled from UltraFeedback) and sample multiple outputs per prompt using stochastic decoding (temperature sampling). Each generated response is scored by the Nemotron reward model, and the highest-scoring sample is treated as a pseudo-label for training. The resulting dataset consists of new (x, y_{chosen}) pairs, where x is the prompt and y_{chosen} is the model’s own high-reward completion.

We then train a new set of LoRA adapters on top of the DPO model. Importantly, this procedure is modular: the extension adapters are trained on top of (but separate from) the DPO adapters, preserving the previous training and allowing easy ablation or swapping of components.

Iterative Bootstrapping. A key advantage of this method is that it enables iterative self-improvement. Once an extended model is trained on synthetic data, it can generate higher-quality responses than its predecessor. These improved completions can be scored again with the same reward model to produce new, even better pseudo-labels. This process can be repeated multiple times with the same prompt set, leading to higher diversity and potentially richer supervision than simply training on the same data for multiple epochs.

Training for more epochs on the same DPO preference pairs leads to diminishing returns and potential overfitting. In contrast, synthetic bootstrapping introduces new and varied responses that can better explore the reward landscape. It effectively increases both the size and diversity of the preference dataset without any new human labeling.

This extension can be viewed as a scalable alternative to collecting new preferences or training new reward models, leveraging large frozen LLMs as reward functions and using the model’s own generations as a source of feedback-aligned supervision.

5 Experimental Setup

5.1 SFT Setup

We fine-tuned a pretrained causal language model on the SmolTalk dataset using full-parameter supervised learning (i.e., no parameter-efficient adapters like LoRA were used in this stage). The model was trained on a setup with three NVIDIA RTX 4090 GPUs, using mixed-precision (FP16) training and data parallelism.

The training used the AdamW optimizer with a learning rate of 5×10^{-6} and weight decay of 0.01. A linear learning rate scheduler with warm-up was applied, along with gradient clipping set to a maximum norm of 1.0. We trained for half an epoch, which was sufficient.

Inputs were batched using a custom data loader with the following configuration: a batch size of 2, maximum prompt length of 256 tokens, and maximum response length of 1024 tokens, as well as shuffled the data before training.

This setup produced a strong base model capable of following instructions and generating coherent responses, which was then used as the initialization for the DPO alignment stage.

5.2 DPO Setup

We fine-tuned our SFT model using Direct Preference Optimization (DPO) on the binarized UltraFeedback dataset. DPO requires pairs of model completions $(y_{\text{chosen}}, y_{\text{rejected}})$ given a shared prompt x . The model is trained to prefer the chosen response by optimizing the following objective:

$$\mathcal{L}_{\text{DPO}} = -\log \sigma(\beta [(\log \pi_{\theta}(y_{\text{chosen}}|x) - \log \pi_{\text{ref}}(y_{\text{chosen}}|x)) - (\log \pi_{\theta}(y_{\text{rejected}}|x) - \log \pi_{\text{ref}}(y_{\text{rejected}}|x))])$$

where π_θ is the trainable policy, π_{ref} is the frozen reference model (our SFT model), and β is a temperature parameter controlling the sharpness of the preference distinction.

We used the same tokenizer and prompt formatting setup as in SFT, with a maximum of 256 prompt tokens and 1024 response tokens. For DPO, we froze the base model and trained a LoRA adapter on top of it using the PEFT library. The LoRA configuration used rank $r = 8$, $\alpha = 16$, dropout of 0.05, and targeted the attention and MLP projection layers (e.g., `q_proj`, `k_proj`, `fc1`, etc.).

Training was conducted with a batch size of 2 and gradient accumulation of 8 steps (effective batch size 16). We used the AdamW optimizer with a learning rate of 1×10^{-6} and a StepLR scheduler with decay factor 0.5 and step size equal to 20% of the steps per epoch. The DPO model was trained for 3 epochs, and checkpoints were saved at 1500 steps and after each epoch. Validation loss was monitored every 500 steps using a held-out test split. The checkpoint after 1 epoch was selected for usage.

All training was done on three RTX 4090 GPUs with mixed-precision (FP16) and gradient checkpointing for memory efficiency.

5.3 Synthetic Extension Setup

To extend the capabilities of the DPO-finetuned model without requiring additional human-labeled preferences, we implemented a bootstrapped preference learning approach based on synthetic reward signals. Specifically, we reused 400 prompts from the UltraFeedback dataset and generated two candidate completions for each using the DPO model trained in the previous stage. These completions were scored using Nemotron 70B, a strong LLM capable of providing reward estimates aligned with human preferences.

Each prompt-response pair was scored independently, and the higher-scoring completion was designated the "chosen" sample while the lower was treated as "rejected." This allowed us to construct synthetic preference tuples $(x, y_{\text{chosen}}, y_{\text{rejected}})$ without requiring manual annotation or training a separate reward model. We refer to this approach as reward-space bootstrapping, since it keeps the input space fixed and synthesizes new data purely through model generation and reward estimation.

We then trained a second-stage LoRA adapter on top of the DPO model using these 400 synthetic preference samples and the DPO loss. Generation used a temperature of 0.7 and nucleus sampling with top-p = 0.95. Hyperparameters such as learning rate, LoRA configuration, and prompt/response length limits were kept consistent with the original DPO training.

Notably, this bootstrapped extension could be iterated: once the extended model is trained, it can be used to generate higher-quality completions for the same prompts (or new prompts), which can again be scored and used to train an even better model. This cycle allows for continual refinement using a static prompt set and a frozen reward LLM, providing a scalable alternative to traditional multi-epoch training over the same data.

6 Results

6.1 Evaluation Strategy

To assess the impact of supervised fine-tuning (SFT), Direct Preference Optimization (DPO), and our synthetic data extension, we evaluate the resulting models using two complementary methods:

1. Pairwise Preference Evaluation using Nemotron. For each pair of models, we sample a set of prompts and have both models generate a response. We then score each response using Nemotron 70B, a large language model that produces scalar reward estimates aligned with human preferences. The model whose response receives the higher reward is considered the winner for that prompt. We compute the win rate of each model across three pairwise matchups: SFT vs. DPO, DPO vs. Extension, and Extension vs. SFT. Ties where both models receive the same score are excluded from win rate calculations unless otherwise noted.

2. External Benchmark via CS224N Leaderboard. We additionally submit responses from each model on 400 prompts from the UltraFeedback test set to the CS224N project leaderboard, which

uses a proprietary evaluation protocol based on GPT-4 ranking judgments. This serves as an external, standardized benchmark of instruction following quality, providing a consistent point of comparison across teams and modeling approaches.

3. Qualitative Comparison of Model Outputs. In addition to quantitative metrics, we manually inspect sample completions from each model across a diverse subset of prompts. This allows us to analyze not just which model scores higher, but also how their outputs differ, e.g., in verbosity, formatting, factual accuracy, safety, or helpfulness. These qualitative comparisons highlight stylistic and behavioral shifts induced by DPO and by the synthetic extension procedure, offering insight into what kinds of improvements (or regressions) the training stages produce.

6.2 Iterative Extension Comparison

To further analyze the value of our reward-space bootstrapping approach, we also compare different versions of the extension model across multiple iterations of training. Each iteration consists of: (1) generating new completions using the current model, (2) scoring them with the frozen reward model (Nemotron), and (3) training a new LoRA adapter on the resulting synthetic preference tuples.

We aim to quantify whether performing multiple rounds of such extension training yields meaningful gains in model alignment and response quality, even when reusing the same prompts. This helps determine whether our synthetic feedback loop offers a viable alternative to traditional multi-epoch training over static preference datasets.

6.3 Quantitative Evaluation

We report pairwise win rates between models based on Nemotron-scored completions over 400 prompts from the UltraFeedback test set. The win rate reflects the fraction of prompts for which one model’s response was rated higher than the other’s. Ties were excluded from the calculations.

Model A	Model B	Win Rate (A > B)	95% CI
Extension	DPO	0.537	[0.506, 0.568]
Extension	SFT	0.564	[0.533, 0.595]
DPO	SFT	0.525	[0.494, 0.556]

Table 1: Pairwise win rates using Nemotron reward scores with 95% confidence intervals.

These results show consistent improvements across training stages. The DPO model outperforms the SFT baseline, and the synthetic extension further improves over DPO, despite being trained on only 400 additional synthetic preferences.

Unfortunately, we were not able to get the leaderboard scores back in time, so they will be excluded from the report.

Comparison	Win Rate (A > B)	95% CI
Iter2-Epoch1 vs. Iter1-Epoch1	0.570	[0.539, 0.601]
Iter2-Epoch1 vs. Iter1-Epoch3	0.519	[0.488, 0.550]
Iter1-Epoch3 vs. Iter1-Epoch1	0.517	[0.486, 0.548]

Table 2: Pairwise win rates between different synthetic training iterations.

In Table 2, we observe that models trained through additional synthetic iterations outperform those trained merely for more epochs on the same synthetic dataset. Here, a synthetic iteration refers to the process of generating new responses using the current model, scoring them with a reward model (in our case, Nemotron 70B), and using the resulting preferences to fine-tune the model further. This iterative bootstrapping approach consistently yields greater improvements than simply training longer on static synthetic data.

Prompt: Arrange the numbers 34, 7, 89, 12, and 45 in ascending order. What is the last number in the new arrangement?

DPO Model Response:

To arrange the numbers in ascending order, we simply need to arrange them from smallest to largest. Starting from the smallest number, 34, we add 1, 12, and 13 to reach $34 + 1 + 12 + 13 = 60$, and then add 7 to get 67. Then, adding 45 to get 112. Finally, adding 89 to get 191.

So, the final order is 34, 7, 89, 12, and 45 arranged in ascending order. The last number in the new arrangement is 191.

191 is indeed the last number in the new arrangement.

Iter1-Epoch1 Model Response:

To arrange the numbers in ascending order, we need to compare them and arrange them from smallest to largest.

The smallest number is 34, so it will be the first number in the new arrangement.

Next, we compare 7 and 89, and since 89 is larger, it will be placed after 7.

Finally, we compare 12 and 45, and since 45 is larger, it will be placed after 12.

So, the final arrangement is 34, 7, 12, 45, 89.

The last number in the new arrangement is 89.

Table 3: Comparison of model responses to a sorting prompt.

6.4 Qualitative Analysis

Both models attempt to solve a simple sorting task, but their performance diverges significantly. The DPO model produces a highly incoherent response, blending incorrect arithmetic with a faulty sort and ultimately arriving at a nonsensical answer (191). It appears to misunderstand the prompt and attempts additive reasoning instead of sorting.

In contrast, the synthetic model from Iteration 1 Epoch 1 provides a mostly structured explanation that, while slightly verbose and redundant in reasoning, reaches the correct sorted sequence and final answer (89). This highlights a key advantage of using Nemotron-labeled synthetic feedback: it encourages more precise and verifiable reasoning chains in fine-tuning. The synthetic model shows better logical structure and factual correctness, suggesting that reinforcement using reliable reward signals, even from another LLM, can meaningfully correct major failure modes.

7 Discussion

Our experiments demonstrate that preference-based fine-tuning with human-annotated and synthetic supervision can significantly improve LLM behavior over standard supervised fine-tuning. The progression from SFT to DPO yielded consistent gains, validating prior findings that contrastive preference modeling improves alignment. However, the most notable result came from our synthetic extension: a relatively small set of Nemotron-labeled preferences was sufficient to outperform the original DPO model.

This finding supports the hypothesis that reward-space bootstrapping can substitute, at least partially, for costly human labeling. By scoring model-generated responses with a strong, frozen reward model, we obtained feedback that improved alignment without modifying the reward function itself. Unlike traditional multi-epoch training, which yielded diminishing returns, iterated bootstrapping produced additive gains, showing that new generations even over fixed prompts can continue to explore higher reward regions in output space.

Qualitative results further confirm that synthetic supervision improves logical consistency and factual accuracy. In particular, the corrected sorting task highlights how a reward model can steer generation toward interpretable and correct outputs, even when the base model produces confident but erroneous responses.

Nevertheless, some limitations remain. While reward models like Nemotron offer scalable supervision, they may also encode biases or failure modes that propagate into the trained model. Additionally, the

performance plateau observed after two iterations suggests diminishing marginal value, motivating future work in prompt variation or reward model ensembling.

In sum, our approach offers a lightweight, modular path to alignment that is scalable and model-agnostic. It bridges supervised and reinforcement-style training while avoiding most of the complexity of full RLHF pipelines.

8 Conclusion

In this project, we explored a multi-stage pipeline for aligning language models to human preferences using both supervised and synthetic methods. Beginning with SFT on the SmolTalk dataset, we established a strong instruction-following base. We then applied Direct Preference Optimization (DPO) on the UltraFeedback dataset to refine model behavior based on pairwise human preferences. Finally, we introduced a novel synthetic extension method that leverages a frozen reward model (Nemotron-70B) to generate and score new completions, enabling preference learning without additional human supervision.

Our results show that each stage in this pipeline improves alignment, with synthetic bootstrapping offering meaningful gains beyond traditional training. The iterative nature of this method allows for continued improvement by simply generating, scoring, and fine-tuning on model-labeled data. This makes it a scalable and practical alternative to full reinforcement learning approaches.

Overall, our work highlights the potential of modular, reward-model driven alignment strategies and provides a foundation for future research into scalable preference tuning with minimal human oversight.

9 Team Contributions

- **Daniel Sorvisto:** Sole member, did everything.

References

- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv:2212.08073 [cs.CL]* <https://arxiv.org/abs/2212.08073>
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Weizhu Wang, and Zichao Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685* (2021).
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. RLAIIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. *arXiv:2309.00267 [cs.CL]* <https://arxiv.org/abs/2309.00267>
- Joshua Mu, Lauren Weinberger, Xuechen Zhang, Shixiang Shane Wu, Colin Raffel, Rohan Taori, Tatsunori B Hashimoto, Percy Liang, and Eric Zhang. 2023. SmolTalk: Lightweight LLM Instruction Tuning with Synthetic Dialogues. *arXiv preprint arXiv:2310.03052* (2023).
- NVIDIA. 2024. Nemotron-3 70B Reward Model. <https://huggingface.co/NVIDIA/Nemotron-3-70B-Reward>. Accessed 2025-06-09.
- 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 Miller, Maddie Simens, Amanda Askill, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. [arXiv:2203.02155 \[cs.CL\]](#)

Raphael Rafailov, Yi Wang, Xuechen Li, Eric Zhang, and Tatsunori Hashimoto. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290* (2023).

Daniel Yuan, Steven Lin, Zhengbao Jiang, Aojun Zeng, Andy Zou, Kenny Yang, Yichi Qin, Frank F Liu, Christopher Potts, Tatsunori Hashimoto, et al. 2023. UltraFeedback: Boosting Language Models with High-Quality Feedback. *arXiv preprint arXiv:2310.01377* (2023).