Extended Abstract: Precise RLAIF: Improving Instruction Following with Sentence-Level AI Feedback (Default Project)

Motivation State-of-the-art models currently use Reinforcement Learning with Human Feedback (RLHF) as it enables optimization on "complex, sequence-level objectives that are not easily differentiable and therefore ill-suited for traditional supervised fine-tuning (SFT)" (Lee et al., 2024). However, there is a gap in the literature on assigning scores to individual sentences within a multi-sentence LLM response sequence for task optimization due to the challenges in that procuring this data from humans would entail. Recent research has highlighted the potential of Reinforcement Learning with AI Feedback to replace the human in choosing between two responses for model training (Lee et al., 2024; Dong & Ma, 2025) but these have been limited to sequence-level objectives only. We explore a method to implement granular, sentence-level feedback in the DPO pipeline.

Method The first step in the Reinforcement Learning pipeline is to fine-tune the model on high-quality data from the task through supervised fine-tuning (SFT). In our pipeline, we use SFT to finetune the model using 460k prompt-completion pairs from the Smoltalk dataset after a pre-processing step to generate the correct tokens, labels, and mask. Then we finetune the model with a Direct Preference Optimization objective over the Ultrafeedback dataset. Finally, to achieve sentence-level scoring, we use an LLM to generate a "worse" (less relevant) version of a single line according to a set of "constitutional principles" for generation. The "Precise RLAIF" step is to finetune the model with the DPO objective with this newly generated dataset.

Implementation We benchmark our improvements in the model training pipeline using two approaches. First, we assess the model win-rate with an (1) automated pipeline that uses Nemotron 70 B. to select the winning completion among two candidate responses conditioned on a test prompt, and (2) human evaluation by asking five anonymous Stanford students to select five responses each for a total of 25 datapoints. Second, we assess the extent to which model degradation / forgetting occurs by benchmarking the model's performance on three unrelated tasks: english-french translation with Flores-101 dataset, measured by the BLEU metric; 8th grade school multi-step math problems with GSM8K dataset; and coding tasks on the python MBPP dataset. The resulting scores were then used to compute both average degradation and distributional drift across examples.

Results We find that Precise RLAIF achieves a promising increase in perforamcne against DPO, with a win-rate against the Qwen 2.5 0.5B. baseline going from 68.5% to 75.2 % evaluated by Nemotron 70B. Additionally, when compared with the SFT baseline and evaluated by humans, the Precise RLAIF achieves a 66.4% win-rate, compared to 55 % from DPO. These results are especially promising because (i) it outlines a useful pipeline to generate granular sentence-level feedback that was previously unfeasible to collect at scale from human data, and (ii) its performance is subject to scaling laws: as the model used to generate degradations gets better, the increases in performance from this pipeline should increase as well.

Discussion On the generalizability of our results, we also observe evidence of small model degradation from a performance drift across unrelated tasks with a 6 % shift in coding capabilities and 1.7 % on translation tasks, with a distributional shift to the left of the mean and shows a broad tail degradation, which implies the model underperforms across unrelated tasks and also on some examples that the baseline handled well.

Conclusion This paper introduces "Precise RLAIF" for granular, sentence-level scoring for task optimization on instruction following. We find that Precise RLAIF achieves a promising increase in performance against DPO and SFT. Further research is warranted to develop this method further.

Precise RLAIF: Improving Instruction Following with Sentence-Level AI Feedback (Default project)

Marcelo Peña
Department of Computer Science
Stanford University
marcelop@stanford.edu

Abstract

State-of-the-art models currently use Reinforcement Learning with Human Feedback (RLHF) as it enables optimization on "complex, sequence-level objectives that are not easily differentiable and therefore ill-suited for traditional supervised fine-tuning (SFT)" (Lee et al., 2024). This paper introduces "Precise RLAIF" for granular, sentence-level scoring for task optimization on instruction following. We find that Precise RLAIF achieves a promising increase in perforamcne against DPO, with a win-rate against the Qwen 2.5 0.5B. baseline going from 68.5% to 75.2 % evaluated by Nemotron 70B. Additionally, when compared with the SFT baseline and evaluated by humans, the Precise RLAIF achieves a 66.4% win-rate, compared to 55 % from DPO. These results are especially promising because (i) it outlines a useful pipeline to generate granular sentence-level feedback that was previously unfeasible to collect at scale from human data, and (ii) its performance is subject to scaling laws: as the model used to generate degradations gets better, the increases in performance from this pipeline should increase as well. However, we also observe evidence of model degradation from a performance drift across unrelated tasks with a 6% shift in coding capabilities and 1.7% on translation tasks.

1 Introduction

State-of-the-art models currently use Reinforcement Learning with Human Feedback (RLHF) as it enables optimization on "complex, sequence-level objectives that are not easily differentiable and therefore ill-suited for traditional supervised fine-tuning (SFT)" (Lee et al., 2024). However, there is a gap in the literature on assigning scores to individual sentences within a multi-sentence LLM response sequence for task optimization due to the challenges in that procuring this data from humans would entail. Recent research has highlighted the potential of Reinforcement Learning with AI Feedback to replace the human in choosing between two responses for model training (Lee et al., 2024; Dong l& Ma, 2025) but these have been limited to sequence-level objectives only. This project will pursue the "Synthetic Data Augmentation" extension (2.1) by seeking to improve LLM's instruction following abilities by leveraging RLAIF for granular feedback on responses at the sentence level. We explore the potential use of RLHF for granular, sentence-level scoring for task optimization (instruction following) and compare it to the performance of RLHF using Direct Preference Optimization, potentially opening the door for improved performance in RLAIF through data collection that was previously unattainable. The research question is: how does sentence-level RLAIF compare to sequence-level DPO and SFT as a finetuning strategy on instruction-following tasks?

2 Related Work

This work builds upon the literature on Reinforcement Learning from Human Feedback (Stiennon et al., 2020; Ouyang et al., 2022) by closely resembling its learning stages, the Constitutional AI design of a "supervisor model" (Bai et al., 2022), and the direct RLAIF approach described in

Canonical Reinforcement Learning from AI Feedback In 2022, Bai et al. at Anthropic introduced RL from AI Feedback in the paper "Constitutional AI." RLAIF is a promising alternative to RLHF that trains the reward model (RM) on preferences generated by an off-the-shelf LLM, eliminating the need for human feedback labelling. Bai et al. use canonical RLAIF to train an AI assistant with the task of reducing harmful content in model outputs. Particularly, the steps in the Constitutional AI process involve designing an "AI critic" that makes decisions based on a small set of principles drawn from a constitution outlined by the researchers. The model is evaluated using ELO scores for helpfulness and Harmlessness.

Two limitations are directly relevant to our work: (1) although the constitution is a minimal set of natural language instructions, Bai et al. find that the particular design of the constitution highly impacts model performance across experiments; to address this, this project will design and test multiple instruction sets across experiments (2) the approach relied heavily on "induced reasoning through Chain-of-Thought" prompting, and finds that CoT samples have a varying impact on models depending on their size; to address this, our experimental design includes CoT experiments on a model of fixed size. Additionally, one issue identified by a previous Bai et al. paper (2022) – yet unadressed by canonical RLAIF –is that training a reward model may render it "stale" as the policy is trained (in the RL step) as the generated trajectories become increasingly out-of-distribution from the dataset the RM was trained on, leading to suboptimal performance. An alternative would be to conduct iterative RLAIF – periodically training a new RM on the latest policy–but this would approach would be too time and resource consuming.

Direct RLAIF On 2024, Lee et al. address some lingering limitations of RALIF in "RLAIF vs. RLHF" and introduce direct RLAIF (d-RLAIF). In contrast to RLAIF, this alternative directly uses an off-the-shelf LLM to provide feedback as the reward signal in the RL step of RLHF. Therefore, the LLM is prompted to "rate the quality of a generation between 1 and 10" (Lee et al., 2024, p. 4). given a researcher prompt. Then, the likelihood of each score token is computed, normalized to a probability distribution, and a weighted score is computed and, again, normalized to the range [-1, 1]. The RL step (using REINFORCE) is then conducted in a similar manner to canonical RLAIF but using the direct score instead of a RM score. Lee et al. find that human evaluators strongly prefer the RLAIF and RLHF models over the SFT baseline for the summarization and helpful dialogue generation tasks, and that RLAIF is equally preferred to RLHF (71 % and 73 %, respectively).

This work highlights the importance of the labeler model size, in line with other scaling behaviors observed by Kaplan et al., 2020. That is, increasing the model size of the labeller model (from PaLM 2 L to PaLM 2 XS) results in a 15% increase in labeller alignment. This is a promising result for RLHF as off-the-shelf models get larger and more capable.

However, this paper is limited as it only scores sequence-level objectives (entire responses) through the LLM, lacking granularity. Although they highlight some trends in human testing (reduced fluency, misleading intention), they do not systematically benchmark these features against RLHF. Additionally, the experiments are based solely on PaLM, which lacks convincing power–the reliance on just one model limits the persuasiveness of their results.

3 Method

3.1 Supervised Fine Tuning (SFT)

The first step in the Reinforcement Learning pipeline is to fine-tune the model on high-quality data from the task through supervised fine-tuning (SFT). Supervised Fine-Tuning is the same next-token prediction objective that is used in pre-training, but no loss is applied to the query tokens. The supervised learning objective in equation (1) is optimized over queries x and completions y drawn from an expert distribution. In our pipeline, we use SFT to finetune the model using 460k prompt-completion pairs from the Smoltalk dataset after a pre-processing step to generate the correct tokens,

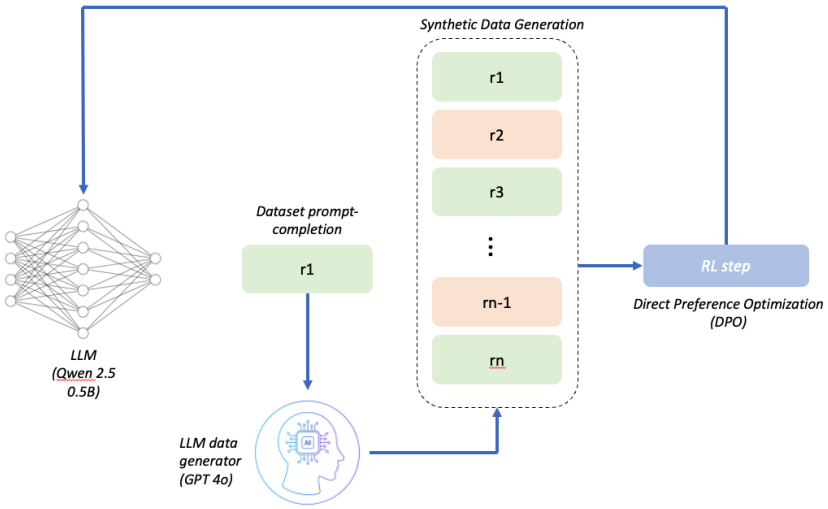

Figure 1: Method Overview.

labels, and mask.

$$\max_{\theta} \mathbb{E}_{x,y \in D} \sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t}) \tag{1}$$

3.2 Direct Preference Optimization (DPO)

Direct Preference Optimization is an objective function that encourages a model to assign a higher likelihood to preferred responses over dispreferred ones. Unlike RLHF, DPO avoids explicitly computing or modeling scalar reward values with a reward function, which leads to more stable training and simplifying the pipeline. The DPO loss is outlined in (2) where x is the prompt, y_w is the preferred response, y_l is the dispreferred response, π_θ is the policy that is being optimized, and π_{ref} is the finetuned policy from SFT.

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]. \tag{2}$$

3.3 Precise Reinforcement Learning with AI Feedback (Precise-RLAIF)

For the extension part of the project, the novel contribution of this paper is to achieve granular scoring (sentence-level) in RLAIF through an off-the-shelf model for improved instruction following. Specifically, refer to Figure 2 (on the left) and assume a 5-line model response to an instruction following task. To achieve sentence-level scoring, we use an LLM to generate a "worse" (less relevant) version of a single line according to a set of "constitutional principles" for generation. In theory we can generate 5! - 1 combinations of responses by picking different combinations of lines to "worsen" and use the LLM to generate these new sequences. For this pipeline we will generate five new pairs from each original completion, degrading a randomly selected sentence each time. This new approach lets us attribute changes in the final quality of the sequence to individual sentences.

We then test the approach by running DPO on the newly generated dataset. The RL step minimizes the loss in Eq. ((2)) and should enable the model to learn the desirability of individual sentences within a multi-level sequence.

$$\mathcal{L}_r(\phi) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(r_\phi(x, y_w) - r_\phi(x, y_l) \right) \right] \tag{3}$$

Second, we benchmark our improvements in the model training pipeline using two approaches. First, we assess the model win-rate with an (1) automated pipeline that uses Nemotron 70 B. to select the winning completion among two candidate responses conditioned on a test prompt, and (2) human

Figure 2: Structure of a 5-line model response (left) and one synthetically-generated pair (right).

evaluation by asking five anonymous Stanford students to select five responses each for a total of 25 datapoints. Second, we assess the extent to which model degradation / forgetting occurs by benchmarking the model's performance on unrelated tasks before and after the finetuning process and analyzing the results across test prompts. The datasets used are:

- For translation, we use the Flores-101 dataset for english to french translations and evaluate using the BLEU metric implemented using the SacreBLEU python library.
- For math, we use the GSM8K dataset with 8th grade school multi-step math problems and scored following a recommended automated prompt on GPT 4o from a -2 to +2 scale.
- For coding datasets, we use the MBPP dataset, consisting of function-writing tasks solvable in 5–10 lines of Python. Models were asked to generate valid implementations given a natural language description. Completions were evaluated by GPT-4.0 using the same scalar helpfulness scale as the math tasks. The resulting scores were then used to compute both average degradation and distributional drift across examples.

Finally, during the design process, we had proposed to use Mechanical Turk to set up human evaluations with a larger sample size. However, after conversations with a teaching assistant, we concluded that it would not be possible to set up these experiments within the allotted time for the class. Additionally, the price of using Mechanical Turk for dataset annotation would have been higher than the reasonable budget for the project.

4 Experimental Setup

For all experiments, we used a g6e.4xlarge instance from AWS, equipped with a single NVIDIA A10G GPU (24 GB VRAM), 16 vCPUs, and 64 GB of RAM. This setup provided sufficient compute for both supervised fine-tuning and preference optimization stages.

4.1 SFT

- Base model: Qwen2.5-0.5B.
- Dataset: Smoltalk dataset, 460k prompt-response pairs from GPT 4 for Supervised Fine Tuning the base model to improve performance before DPO step.
- Preprocessing: Restricted conversations to length 2 and capped input tokens to 256 (input) and 1024 (output). Applied chat format, tokenized with default loaded Qwen tokenizer, applied left padding before the prompt and right padding after the prompt. Generated label and mask one-hot encoding vectors.
- Training: batch size of 4 with 8 gradient accumulation steps; learning rate: 1e-5; optimizer: AdamW; gradient clipping to 1.0; weight decay of 0.01; enabled floating point mixed precision; finetuned full model.

4.2 Direct Preference Optimization

- Base model: SFT finetuned Qwen2.5-0.5B. Frozen model.
- Dataset: Ultrafeedback dataset, 63k entries of one prompt and four completions. Filtered the winning and losing completions.

- Preprocessing: Restricted to one winning pair and one losing pair per prompt. Applied chat format, tokenized with default loaded Qwen tokenizer, applied left padding before the prompt and right padding after the prompt. Created dataset of (prompt, preferred response, dispreferred response) tuples.
- Training: batch size of 4 with 4 gradient accumulation steps; AdamW optimizer; learning rate 1e-5; weight decay of 0.01; gradient clipping to 1.0; beta of 0.2; enabled floating point mixed precision.

4.3 Precise-RLAIF

Experimental setup similar to DPO step, with key differences outlined below:

- Base model: DPO finetuned Qwen2.5-0.5B. Frozen model.
- Dataset: Synthetically generated dataset, 1k entries of one prompt and five degradations from the same completion. Reshaped to 5k entries with one degradation each.
- Training: batch size of 4 with 4 gradient accumulation steps; AdamW optimizer; learning rate 1e-5; weight decay of 0.01; gradient clipping to 1.0; beta of 0.2; enabled floating point mixed precision.

5 Results

We ran a series of experiments to compare the performance of the proposed pipeline, which we will detail below.

5.1 Quantitative Evaluation

Table 1 shows a summary comparison of win-rates across methods. When evaluating each model against the baseline Qwen 2.5 .5B using Nemotron 70B, our SFT finetuned model achieves a 42.1% win-rate, DPO achieves a 68.5 % win-rate, and finally Precise-RLAIF achieves a 75.2% win-rate. Therefore, we find that our implementation of SFT has a lower impact on the win-rate vs. the baseline than expected; however, since this is the first step in the RL pipeline, we believe that it increases the dataset diversity and it might still be necessary to enable downstream RL performance. This foundational step may not outperform the base model directly, but the literature shows that it is still crucial for enabling stable preference optimization and downstream RL stages such as DPO and RLAIF. Additionally, Precise RLAIF achieves an improvement of 9.7 % on top of DPO against the baseline, which validates it as a viable method to further finetune language models.

We also evaluate the model's win-rate with five human testers, but this time using the SFT model as a baseline since this is the frozen reference model in the DPO pipeline. We find that DPO achieves a 55% win-rate against our implementation of SFT, while Precise-RLAIF achieves a 66.4 % win-rate with humans. These results confirm the increased preferability of completions generated with Precise RLAIF vs. DPO. The lower win-rate compared to the Nemotron evaluation can be attributed to the small-N sample (25 pairs across five testers) and the possible variance within each tester's scoring which was not normalized. Albeit encouraging, further human testing is needed to confirm these experimental results.

Method	**Win-rate vs. baseline (Nemotron 70B)**	**Win-rate Human vs. SFT**
SFT	42.1%	–
DPO	68.5%	55.0%
Precise RLAIF	75.2%	66.4%

Table 1: Comparison of win-rates across methods

5.2 Comparison of perturbation methods

A key stage of the proposed Precise RLAIF pipeline is the degradation strategy, which can affect performance according to the Constitution (Bai et al., 2022) chosen to define what a "worse sentence"

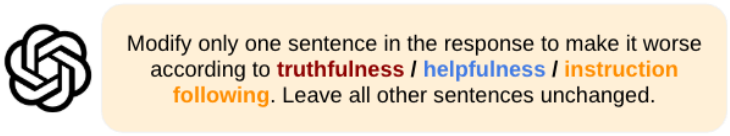

Figure 3: Summary of perturbation strategies (in different colors).

means for the model. For this experiment, I generated a new perturbation dataset with each different perturbation strategy, finetuned the SFT checkpoint, and sampled 100 prompts from the test set to evaluate the win-rate vs. the baseline model (Qwen 2.5 0.5B).

Table 2: Win-rates vs. SFT for each perturbation strategy for task-following evaluation.

Perturbation Strategy	**Expected Win-rate vs. SFT**
Instruction-Following	73%
Helpfulness	71%
Truthfulness	68%
All of the above	75.2% (prior result)

From table 2 we observe that the model trained on specific instruction-following degradations achieve the highest win-rate (73 %) on our evaluation. This coincides with the intuition that a good degradation strategy depends on the task that is being optimized for; in this case, since the Nemotron model is evaluating based on "instruction-following," it is reasonable that this degradation strategy outperforms the others. The second best degradation strategy heplfulness-based perturbations, which also improve win-rate at 71 % which is comparable to instruction-following. Qualitatively, we observe that there is a direct connection between these two signals, as a sentence that is worse at being helpful is probably also one that does not answer the question directly (instruction-following). From the sample perturbations in Table 3, we can observe this close relationship between the objectives as both perturbations focus on time and provide advice that undermines the rest of the response.

Table 3: Comparison of perturbations for the same prompt across perturbation strategies.

Field	**Text**
Prompt	What are some specific strategies that one can use to practice mindful eating?
Original Sentence	One strategy is to eat slowly and deliberately, savoring each bite and taking small bites, pausing between bites to recognize satiety cues and avoid overeating.
Worsened Version (Instruction-Following)	One strategy is to eat quickly so you can finish your meal on time, which might help with time management.
Worsened Version (Truthfulness)	One strategy is to eat while standing and watching TV, as this has been scientifically proven to enhance digestion.
Worsened Version (Helpfulness)	One strategy is to do something different with your food to finish it in less time.

Finally, the worst performing degradation strategy was "truthfulness" with a small decline on the win-rate of the model from the 68.5 % win-rate of DPO vs. the baseline. This is likely due to the less direct connection to the evaluation signal and the fact that the test set is not a heavy QA dataset, which makes this perturbation strategy less effective. For fact-based tasks, we believe this method has the potential to reinforce "facts" for the model and reduce hallucinations; however, a large-scale dataset would be needed to ensure that the test facts are close to the distribution of the train set. This is not the case in our experiment, which explains the lower impact on the win-rate. Finally, a longer prompt that combines all the mentioned factors into a "complete constitution" performs better (75.2 %) than each on their own since it likely is a better proxy for human preference for generated responses than each individual characteristic.

6 Discussion

The results presented in Section 5 show the promise of the Precise RLAIF pipeline to improve LLM task-following win-rates through granular, sentence-level signals. To evaluate how the results would generalize beyond this experimental setup, I evaluated the downstream effects of sentence-level DPO fine-tuning on model capabilities. A common criticism of RL is found in the literature: "optimizing proxy reward models can degrade true performance in RLHF (...) [specifically on] unrelated tasks" (Hou et al., 2024). To test this, I trained the model over 6 epochs and evaluated its performance on three unrelated tasks—translation (Flores 101 English to French), math (GSM8K math problems), and coding (MBPP python coding prompts)—across 700 samples each. The results are summarized in Table 4.

Table 4: Performance on general tasks before and after DPO fine-tuning.

Task	**Score (Before)**	**Score (After)**
Translation (SacreBLEU)	18.58	18.92
Coding (Avg. score)	1.45	1.18
Math (% correct)	8%	7%

As can be seen, the model saw a decline of 6% in its coding capabilities, and 1.7 % on translation tasks. With respect to math, it is hard to draw conclusions since the original performance on 8th-grade math tasks was very low (8%) initially, but even on this noisy signal, the finetuned model's performance declined slightly. Additionally, I plotted a per-example score delta in Figure 4 to have a distributional analysis of this performance drift on math and coding tasks. As can be seen, the distribution means are shifted left and shows a broad tail degradation, which implies the model underperforms across the distribution of tasks and also on some examples the baseline handled better. These findings suggest that without regularization and careful training hyperparameter tuning, fine-grained finetuning may trade off some general capabilities. However, since my Precise RLAIF step was done after DPO finetuning, it may be hard to ascertain which step may have created this slight performance degradation effect.

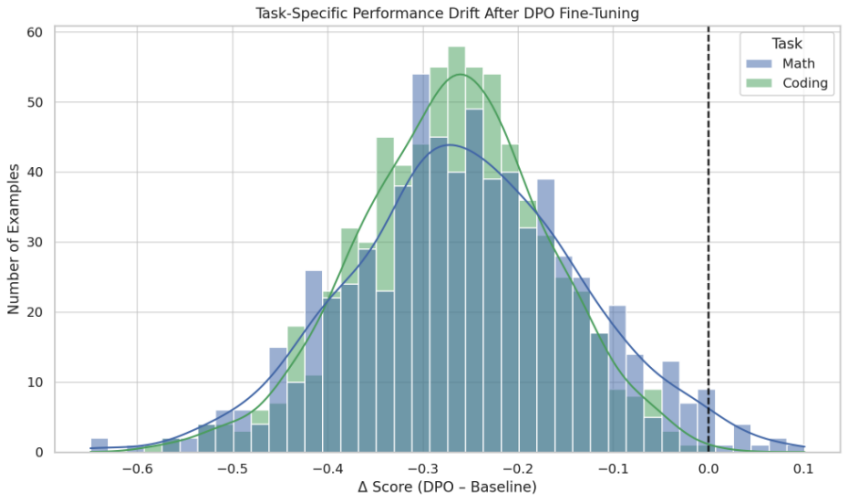

Figure 4: Performance drift across math and coding tasks.

7 Conclusion

This paper introduces "Precise RLAIF" for granular, sentence-level scoring for task optimization on instruction following. We find that Precise RLAIF achieves a promising increase in perforamcne against DPO, with a win-rate against the Qwen 2.5 0.5B. baseline going from 68.5% to 75.2 % evaluated by Nemotron 70B. Additionally, when compared with the SFT baseline and evaluated by

humans, the Precise RLAIF achieves a 66.4% win-rate, compared to 55 % from DPO. These results are especially promising because (i) it outlines a useful pipeline to generate granular sentence-level feedback that was previously unfeasible to collect at scale from human data, and (ii) its performance is subject to scaling laws: as the model used to generate degradations gets better, the increases in performance from this pipeline should increase as well. However, we also observe evidence of model degradation from a performance drift across unrelated tasks with a 6 % shift in coding capabilities and 1.7 % on translation tasks. Two limitations of this paper are (i) given time and compute limitations, it does not run hyperparameter sweeps to optimize training further, and (ii) since the "Precise RLAIF" step was done on top of DPO finetuning, this setup makes it challenging to ascertain which step may have created the slight performance degradation effect. In future work, additional training runs with other hyperparameters should be ran, and the determinant factors for unrelated task degradation should be analyzed in further work.

8 Team Contributions

- **Marcelo:** Whole project.

Changes from Proposal Due to time constraints, did not implement a PPO version with direct numerical scoring of lines to compare performance.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. (2022). arXiv:2204.05862

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. 2022b. Constitutional AI: Harmlessness from AI Feedback. (2022). arXiv:2212.08073

Kefan Dong and Tengyu Ma. 2025. STP: Self-play LLM theorem provers with iterative conjecturing and proving. (2025). arXiv:2502.00212

Zhenyu Hou, Pengfan Du, Yilin Niu, Zhengxiao Du, Aohan Zeng, Xiao Liu, Minlie Huang, Hongning Wang, Jie Tang, and Yuxiao Dong. 2024. Does RLHF Scale? Exploring the Impacts From Data, Model, and Method. arXiv:2412.06000 [cs.CL] `https://arxiv.org/abs/2412.06000`

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. (2020). arXiv:2001.08361

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2023. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. (2023). arXiv:2309.00267