Extended Abstract

Motivation Instruction-following language models have become central to human-AI interaction, yet reliably aligning their outputs with user intent—especially for complex, long-form prompts—remains a challenge. These models often suffer from poor generalization across inputs of varying length and complexity, and further training is limited by the high cost of diverse, high-quality data, leading to overfitting. To address these limitations, we combine curriculum learning to structure model exposure and improve generalization, with teacher-guided self-revision to synthetically expand training, offering a scalable framework for enhancing alignment in compact open-weight models.

Method We fine-tune Qwen 2.5–0.5B using SFT, DPO, and a modified self-revision framework inspired by CITING. While the original CITING algorithm generates unstructured revision data from a teacher LLM, our approach is novel in introducing a length-based curriculum: we partition prompts into short, medium, and long tiers and train the model in stages. This structured curriculum directly addresses our DPO model's failure to generalize to longer, more complex inputs. Our method uniquely combines synthetic teacher-guided revision with curriculum learning, enabling the model to gradually improve revision ability and instruction-following performance across prompt complexities—something not explored in the original CITING setup.

Implementation We implemented our method on the 500M-parameter Qwen 2.5-0.5B model using a three-stage pipeline: supervised fine-tuning (SFT), Direct Preference Optimization (DPO), and CITING with a length-based curriculum. For SFT, we trained on 455k Smol-Smoltalk samples for one epoch using AdamW (learning rate: 1e-6, batch size: 32, gradient accumulation: 32). DPO was applied using 60k UltraFeedback preference pairs. We then fine-tuned with CITING on 6k prompts grouped into short, medium, and long tiers to support curriculum pacing. GPT-4o-mini served as the teacher LLM, generating revision targets via batched prompt completions. Training employed 8 gradient accumulation steps and cross-entropy loss on student outputs aligned to teacher revisions. Additional rounds of self-revision and inference-time recursion were explored to study autonomous improvement.

Results Our approach yielded consistent improvements across training stages. Supervised fine-tuning (SFT) achieved a 40% win rate over the Qwen 2.5-0.5B Instruct baseline, while Direct Preference Optimization (DPO) improved this to 67% and reduced validation loss to 0.630. Building on this, our CITING-based self-revision model—trained with a length-based curriculum and teacher-guided edits—achieved a 71% win rate over DPO and the highest Nemotron score (-23.89). Training and validation loss curves show steady improvement within each curriculum tier (short, medium, long), highlighting the effectiveness of staged instruction tuning. Qualitative comparisons confirmed that DPO enhanced coherence and relevance over SFT, though sometimes at the cost of directness. The self-revision model addressed these limitations, offering more stable and context-aware outputs across instruction types.

Discussion While our approach showed strong improvements over SFT and DPO, several challenges and limitations emerged. The use of GPT-4o-mini as a teacher model, while efficient, introduced variability in revision quality and required repeated inference, making the self-revision process resource-intensive. DPO improved alignment but occasionally sacrificed clarity by inserting unnecessary follow-up questions. Evaluation via the Nemotron-70B reward model provided scalable benchmarking but may not fully reflect human judgment. Managing CITING at scale was technically challenging due to API rate limits and resource constraints, and implementing the curriculum required careful prompt categorization and propagation. Nonetheless, the modular structure of our pipeline enabled us to explore the effects of revision and instruction complexity in a scalable way, revealing that even small models can benefit from structured, teacher-guided self-improvement.

Conclusion We present a multi-stage alignment framework for the Qwen 2.5-0.5B model that combines SFT, DPO, and a length-based, teacher-guided self-revision extension using the CITING algorithm. Our results show that lightweight models can achieve meaningful instruction-following improvements through scalable preference optimization and structured curriculum learning. This approach offers a practical path toward fine-grained alignment without relying on large proprietary models or heavy compute.

Instruction Following via Self-Revision: Fine-Tuning Qwen with Teacher Feedback and Staged Curriculum

Landon Choy
Stanford University
landonc@stanford.edu

Tracy Li
Stanford University
tracyqli@stanford.edu

Abstract

We propose a scalable, multi-stage framework for aligning compact language models with human preferences by integrating supervised fine-tuning (SFT), Direct Preference Optimization (DPO), and a batched curriculum extension of the CITING algorithm. Our method introduces a length-based curriculum that structures training across short, medium, and long prompts, addressing generalization challenges in standard alignment pipelines. Using the 500M-parameter Qwen 2.5-0.5B model, we show that structured teacher-guided revision significantly enhances instruction-following performance beyond SFT and DPO baselines. Experimental results demonstrate improved win rates and reward model scores, while analysis of training dynamics highlights the benefits of staged curriculum learning. This approach enables low-cost, modular alignment improvements for open-weight models without reliance on proprietary systems or massive compute.

1 Introduction

Instruction-following language models have become essential tools for aligning AI systems with human intent. Yet, two major bottlenecks continue to impede progress in reinforcement learning (RL)-based optimization: poor policy initialization and diminishing data diversity as training progresses. These challenges are particularly evident in long-form, complex prompts where models often veer off-topic, produce shallow reasoning, or fail to meet task-specific criteria. Improving generalization and faithfulness in such settings remains a central challenge for scalable alignment.

To address these issues, we enhance the performance of Qwen 2.5-0.5B—an open-weight, resource-efficient model—using a hybrid fine-tuning framework that combines Direct Preference Optimization (DPO) with the CITING algorithm (Feng et al.). Our approach begins with Supervised Fine-Tuning (SFT) followed by DPO, providing a strong initialization baseline. We then apply CITING, a teacher-student framework in which GPT-4o-mini serves as the "teacher" to revise the model's initial responses according to explicit criteria (e.g., accuracy, relevance, completeness). The Qwen student model is subsequently trained to imitate these improved revisions, effectively learning to self-correct based on teacher feedback.

To further boost coherence and instruction fidelity, we incorporate a curriculum learning strategy, organizing training batches by increasing prompt complexity and length. This staged progression enables the model to gradually build capacity for long-horizon reasoning and structured generation under more demanding instructions.

As part of the original CITING algorithm, we implement recursive self-revision—where the student iteratively refines its own outputs—but introduce a novel twist by integrating this process between length-based curriculum tiers, enabling progressive, self-guided improvement.

Our results show that the combined use of teacher feedback, curriculum design, and recursive self-revision significantly improves the model's ability to stay on-topic, follow complex instructions, and produce higher-quality outputs, improving over the DPO baseline with a winrate of 71%. This work contributes a lightweight yet effective recipe for enhancing instruction-following performance in smaller models, bridging the gap between open-weight LLMs and state-of-the-art proprietary systems. It also explores how to improve generalization across diverse inputs—particularly when models like our DPO-tuned baseline overfit quickly and could not be trained further. By integrating teacher-guided self-revision and curriculum learning, we show that LLMs can continue learning in a structured way, even when direct fine-tuning plateaus.

2 Related Work

Our work draws upon and integrates advances in preference optimization, curriculum learning, teacher–student frameworks, instruction tuning via AI-generated feedback, and self-revision in language models. We review each area below and position our contributions in relation to prior work.

Direct Preference Optimization (DPO). Rafailov et al. (2023) introduced Direct Preference Optimization (DPO), a reinforcement learning–free method for aligning language models with human preferences using pairwise comparison data. DPO simplifies and stabilizes alignment by replacing reward models and policy gradients with a contrastive objective that encourages the model to increase the likelihood of preferred responses over rejected ones. We adopt DPO as the backbone of our optimization pipeline, applying it to filtered preference pairs from the UltraFeedback dataset. In our work, DPO serves as a strong post-SFT baseline and sets the stage for further refinement through curriculum-guided, teacher-supervised revision.

Curriculum Learning. The idea of curriculum learning, introduced by Bengio et al. (2009), suggests that training models on easier examples first and gradually increasing complexity improves convergence and generalization. Soviany et al. (2021) provide a detailed survey of curriculum learning strategies and their applications in NLP and vision tasks. Our implementation of CITING incorporates this principle by organizing training prompts into short, medium, and long categories, progressing through them in a staged fashion to build robustness on increasingly complex instructions.

Teacher–Student and AI-Guided Revision Frameworks. Wei et al. (2023) demonstrated how LLMs can act as reasoning teachers, providing rationales and demonstrations to help smaller models learn more effectively. Bang and et al. (2022) extended this with the Demonstrate–Practice–Review (TPD) pipeline, where a teacher provides feedback to guide the student's improvement. Inspired by these approaches, our CITING framework leverages a teacher model (GPT-4o-mini) to revise student outputs based on prompt-specific criteria. The student is then fine-tuned to imitate these improvements, enabling scalable quality enhancement without human-in-the-loop editing.

Instruction Tuning with AI-Created Feedback. Our work builds directly on the CITING framework, which introduces a novel teacher-student instruction tuning paradigm where a teacher LLM provides rubrics and revisions to help a student model iteratively improve its outputs Feng et al. (2023). CITING demonstrates that learning from teacher-generated feedback can outperform both RLHF and ranking-based alignment approaches in articulation, depth, and comprehensiveness.

Our approach. We build CITING directly on DPO as a lightweight, stable preference optimization method and extends it with teacher-guided revision and curriculum design. Unlike prior work, we show that self-revision can be made more effective when embedded into a length-aware curriculum, allowing the model to generalize across instruction types and complexities more robustly. Furthermore, our use of synthetic revisions as a scalable data augmentation strategy is tailored to the model's current stage in the curriculum—something not explored in the original CITING or other teacher-student paradigms. This combination yields a lightweight, extensible, and cost-efficient recipe for enhancing instruction-following alignment in compact open-weight models. By integrating insights from teacher–student learning and curriculum-based training, we propose a scalable and practical framework for improving instruction-following in compact open-weight language models.

3 Method

3.1 Supervised Fine-Tuning (SFT)

We begin with a standard supervised learning phase on Qwen 2.5-0.5B language model using high-quality instruction-response pairs derived from the `Smol-Smoltalk` subset of UltraFeedback. The objective is to train a base model that provides a reasonable initialization for subsequent preference optimization and revision-based fine-tuning.

We treat the problem as a next-token prediction task and apply the cross-entropy loss only over the *completion portion* of each sample. The objective is defined as:

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

The instruction is x, the ground-truth response is y, and π_θ is the student policy parameterized by θ.

3.2 Direct Preference Optimization (DPO)

After SFT, we fine-tune the model using *Direct Preference Optimization* with binarized preference pairs from UltraFeedback. The DPO objective compares the likelihoods of preferred (y_w) and less-preferred (y_l) responses, using the SFT model π_{ref} as a reference policy:

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

This method promotes generation of responses that align with human preferences while maintaining stability relative to the reference model.

3.3 Curriculum Learning with Self-Revision

To further enhance alignment and coherence, we integrate the self-revision framework CITING (Feng et al.) with staged curriculum learning and inference-time self-revision. CITING introduces a teacher-student revision loop, in which an external teacher model (GPT-4o-mini) provides expert self-revisions to student-generated outputs using task-specific prompt criteria over many iterations. We combine this approach with a length-based curriculum and recursive self-revisions at inference time, not only during training.

3.3.1 Length-based Curriculum Design

We divide a randomly selected dataset of $\sim$6,000 UltraFeedback samples into three tiers based on prompt length: **short**, **medium**, and **long**. These tiers reflect increasing instruction complexity and are used to structure training in a staged curriculum. The model trains on each tier sequentially, progressing to the next tier only after validation loss plateaus on the current one.

Unlike traditional curricula based on task type or domain, prompt length serves as a simple but effective proxy for instruction complexity—shorter prompts often requiring more direct responses, and longer ones demanding nuanced, multi-step reasoning. Our DPO model often rambled or failed to stop on short prompts and struggled to handle the reasoning demands of longer ones. These behaviors suggested that a one-size-fits-all training strategy might be insufficient for building coherent, aligned behavior across instruction types.

We treat prompt length as a proxy for difficulty—short prompts typically require concise, direct responses, while long prompts demand complex reasoning and multi-step generation. By training from short to long, we align the model's learning with task complexity, helping it first master basic behaviors (e.g., stopping) before tackling harder inputs, leading to more stable training and better generalization across instruction lengths.

3.3.2 Prompt Criteria Generation

Following the CITING algorithm, we rely on explicit revision criteria for each instruction. We generate these criteria via two complementary strategies:

Teacher-Based Summarization

- We use GPT-4o-mini to manually inspect a representative subset of prompts.
- Each prompt is assigned to an *instruction category* (e.g., classification, summarization, reasoning).
- For each category, we define a set of *revision guidelines* that capture what constitutes a "good" response (e.g., answer format, factuality, structure).

Similarity-Based Propagation

- We encode all unmatched prompts using `sentence-BERT`, generating dense vector representations.
- Each new prompt is compared against the category exemplars using cosine similarity.
- Prompts are automatically assigned to the closest instruction category, and thus inherit its revision criteria.

We use CITING's two-step strategy to assign meaningful, context-aware revision criteria to each prompt without costly manual annotation. Thus, we create consistent supervision that helps the model revise effectively across diverse tasks in our curriculum-based CITING framework.

3.3.3 Curriculum-Based Instruction Tuning

During the main training loop, we use a staged iterative process that gradually builds the model's ability to handle more complex instructions by combining the CITING algorithm with our length-based curriculum.

Training Phase For each curriculum tier (short, medium, long), and training stage $k = 0$ to N, we do the following:

1. **Student Generation:** The student model $\pi^{(k)}$ generates a set of initial responses $r^{(k)}$ for each instruction x.
2. **Teacher Revision:** GPT-4o-mini revises these outputs using the revision prompt:

   $$\text{prompt}(x, c, r^{(k)})$$

   where c is the instruction-specific criterion. We use the following prompt format to elicit teacher revisions during CITING:

   ```
   Below is an instruction and my initial response.
   A criteria for evaluating the response is also provided.

   Instruction:
   {x}

   My Initial Response:
   {r_k}

   Criteria:
   {c}

   My initial response may be incorrect and may not follow the criteria.
   Please revise it using the ideal response as a guide and the criteria
   for improvement. Return only the revised answer, without any additional
   comments or explanation.
   ```

From this, we use GPT's revised response output, $r^{(k+1)}$, as the ground truth for training.

3. **Student Fine-Tuning:** The model is fine-tuned on the prompt $(x, c, r^{(k)}, r^{(k+1)})$ to minimize cross-entropy loss over the revised output, updating the model to $\pi^{(k+1)}$.

Each fine-tuning round runs multiple epochs on fixed student/teacher pairs before regenerating new ones, allowing the model to learn self-revision. Repeated regeneration can improve performance because it exposes the student model to fresh, higher-quality feedback as it evolves—allowing it to correct new types of errors and better mimic the teacher's revisions. Each regeneration round adapts to the student's current outputs, offering more relevant supervision than static training data.

Although repeated regeneration can improve performance, the original CITING paper found diminishing returns after just a few rounds, suggesting a ceiling on benefits from flat, undifferentiated training. To address this, our length-based curriulum aimed to improve generalization, stabilize learning, and help the model incrementally acquire the skills needed to revise responses across a spectrum of task complexity. We repeat training within each tier until model performance stops improving, after which we move to the next curriculum stage to align the model's learning trajectory with increasing instruction complexity, allowing it to incrementally develop revision skills while avoiding the diminishing returns and overfitting observed in undifferentiated, flat training.

4 Experimental Setup

4.1 Base Model

All experiments are conducted using **Qwen 2.5 0.5B**, an open-weight 500M-parameter causal decoder-only transformer model hosted on HuggingFace. This model serves as a lightweight yet capable foundation for instruction fine-tuning experiments.

4.2 Supervised Fine-Tuning (SFT)

We begin by training Qwen 2.5 0.5B using the **Smol-Smoltalk** dataset, a high-quality supervised corpus of instruction-response pairs. The dataset contains:

- 455,000 training samples
- 6,000 validation samples

The model is trained to maximize the log-likelihood of completion tokens only, using next-token prediction. We use an `AdamW` optimizer with a learning rate of 1×10^{-6}, a batch size of 32, and gradient accumulation over 32 steps. Training is conducted for 1 epoch over the full dataset.

To evaluate progress over the base model, we compare our SFT-trained model to the **Qwen 2.5 0.5B-Instruct** baseline. For a shared set of evaluation prompts, we score both models' responses using the parametric reward model (see below) and compute a **win rate**: the percentage of cases where the SFT model receives a higher score than the instruct baseline.

4.3 Direct Preference Optimization (DPO)

After SFT, we apply **Direct Preference Optimization (DPO)** using the **UltraFeedback Binarized** dataset. This dataset consists of pairs of completions labeled with human preference scores.

The filtered dataset contains:

- 60,000 training pairs
- 1,000 validation pairs

The DPO model is initialized from the SFT checkpoint. We use a β value of 0.45, an `AdamW` optimizer with a learning rate of 1×10^{-6}, a batch size of 4, and gradient accumulation over 32 steps. Training is conducted for 1 epoch.

To evaluate improvements over SFT, we use the reward model to score both DPO and SFT model responses and compute a **win rate**: the proportion of prompts where the DPO-trained model receives a higher reward score than the SFT model.

4.4 CITING with Curriculum Learning

For the final stage, we apply the **CITING algorithm** on top of the DPO model. We construct a **curriculum-based dataset** by randomly selecting:

- 3,000 prompts for training
- 100 prompts for evaluation

The training and test prompts are drawn randomly from the UltraFeedback dataset. The training prompts are grouped by input length into *short*, *medium*, and *long* tiers so that approximately $\frac{1}{3}$ of the data falls into each bucket.

For a teacher LLM, we used GPT-o4 mini ("o4-mini-2025-04-16") using OpenAI's batched generations API. We used GPT-o4 mini because it offers a strong balance of speed, cost, and instruction-following quality, especially with batched generations. Its smaller size makes it efficient and affordable for large-scale batched generation, which is critical when repeatedly generating teacher revisions across multiple rounds of self-revision. Despite being lightweight, it still produces high-quality edits that help improve the student model's ability to self-revise effectively.

We use an `AdamW` optimizer with a learning rate of 1×10^{-6}, a batch size of 32, applying 8 gradient accumulation steps to effectively simulate a larger batch size. Training was guided by cross-entropy loss between the student model's self-revised outputs and the corresponding revisions produced by GPT-o4-mini.

We compare the CITING-trained model against the DPO baseline using the reward model and report a **win rate**: the percentage of evaluation prompts for which CITING outperforms DPO.

4.5 Parametric Reward Model for Evaluation

To objectively assess performance across all training stages, we use the **LLaMA 3.1 Nemotron-70B Reward Model** developed by NVIDIA and available on HuggingFace:
`https://huggingface.co/nvidia/Llama-3.1-Nemotron-70B-Reward`

This parametric reward model provides scalar scores for prompt-response pairs and is trained to reflect human judgment. We use the **hosted inference API**, which provides 100,000 free API calls after registration. The API supports OpenAI-compatible querying, and users can evaluate outputs by passing an API key from the linked page.

At each training stage, we:

- Sample a shared set of prompts
- Generate completions from the current and previous models
- Score each response pair with the reward model
- Compute win rates to quantify performance gains

This evaluation pipeline offers a scalable, reproducible method to measure instruction-following quality and alignment improvements across models.

4.5.1 No-Curriculum Baseline

As a baseline, we also compare CITING with and without the length-based curriculum. For the non-curriculum variant, we shuffle the entire pool of 6,000 prompts and sample from it uniformly—without organizing by input length—thus training the model in a flat, non-staged fashion. This allows us to isolate the impact of curriculum structure on the model's alignment quality and generalization performance.

Table 1: Performance Comparison

Method	Win Rate	Average Nemotron Score
SFT	0.40 vs Instruct	-27.85
DPO	0.67 vs SFT	-26.5
No-Curriculum Baseline	0.46 vs DPO	-27.616
Extension Round 1	0.71 vs DPO	-23.89
Extension Round 2	0.40 vs DPO	-27.62

5 Results

5.1 Quantitative Evaluation

We evaluate model performance using both loss metrics and win rates based on reward model scoring. Table 1 summarizes the train/validation loss and comparative win rates for the SFT, DPO, and self-revision stages.

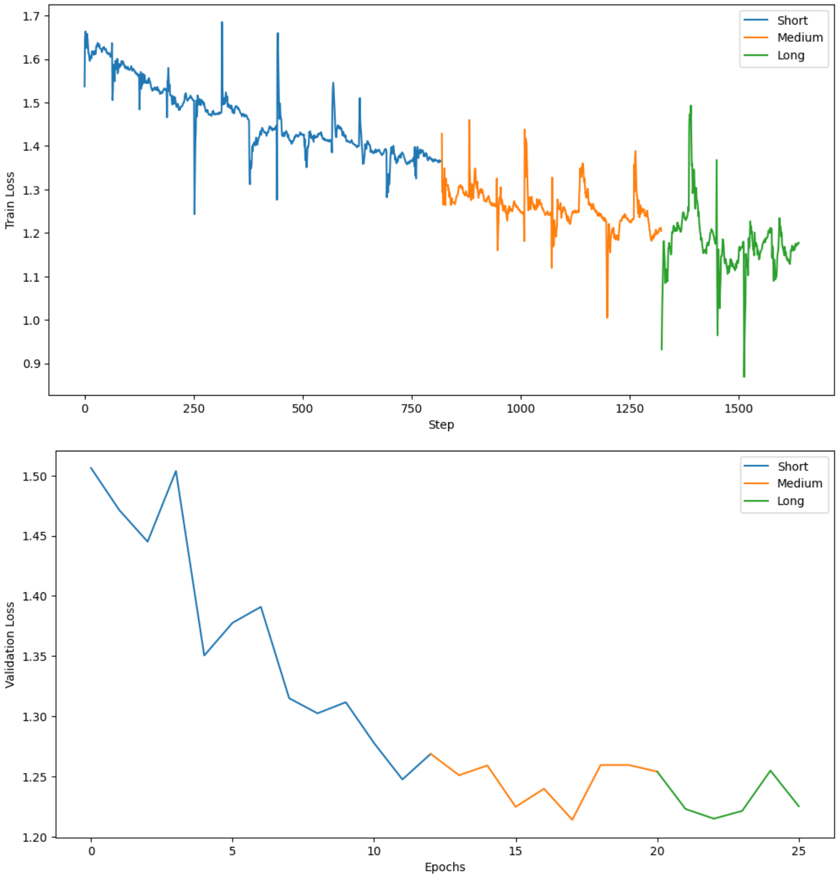

Figure 1: One Round Training and validation loss curves for the self-revision extension across curriculum.

The SFT model's training (0.977) and validation (0.971) losses are very close, suggesting effective generalization to held-out examples without significant overfitting. However, the modest win rate of 40% against the Qwen Instruct baseline indicates that while SFT improved next-token prediction, these gains may not have translated directly into alignment with human-preferred outputs. This aligns with prior observations that log-likelihood-based SFT alone is often insufficient for preference-centric tasks like instruction-following, where models can produce fluent but suboptimal responses.

The application of Direct Preference Optimization (DPO) yields a more substantial leap in performance. The drop in loss to 0.637 (train) and 0.630 (val) not only reflects better convergence on the preference dataset but also improved generalization after preference tuning. The 67% win rate over SFT is especially notable, demonstrating that DPO substantially improves model outputs in alignment with human judgments, even with a relatively small dataset (60k high-confidence pairs). This improvement underscores the strength of DPO's implicit reward modeling: rather than learning to imitate gold responses, the model learns to differentiate and prefer better outputs — a mechanism that appears more closely aligned with evaluation-time scoring and downstream user satisfaction.

For the novel extension, the loss curves in Figure 1 reflect the progression of the self-revision model across the length-based curriculum. In the training loss plot (top), we observe distinct phases corresponding to short, medium, and long prompts—each denoted by a separate color. During the short prompt phase, training loss steadily declines, suggesting that the model effectively internalizes basic instruction-following patterns. Upon transitioning to medium prompts, the loss increases briefly but then stabilizes and continues downward, indicating successful adaptation to more complex instructions. A similar trend is observed when entering the long prompt phase, where the initial volatility is eventually overcome by a steady decrease in loss. These stepwise transitions validate the benefit of curriculum pacing: the model learns incrementally rather than being overwhelmed by instruction complexity from the outset.

The validation loss plot (bottom) mirrors this staged improvement. We see a consistent decrease throughout the short prompt phase, with a plateau forming as the model begins handling medium-length inputs. While the medium and long phases introduce some fluctuation, the overall validation loss remains lower than at initialization, supporting the claim that the model generalizes better across instruction lengths after each stage. This smooth transition through tiers highlights the effectiveness of combining teacher-guided self-revision with a length-based curriculum structure.

With the length-based curriculum and teacher/student training setup, our self-revising model achieved a training loss of 1.103 and a validation loss of 1.225 with one round of self-revision during training. With two rounds, our model achieved a training loss of 1.554 and a validation loss of 1.259. We achieved an average Nemotron score of -23.89 and -27.62 for the one and two round trainings respectively.

With one round of self revision, we achieved a 71% win rate against DPO, and 40& with two rounds. Despite expectations, the second round of self-revision underperformed compared to the first, both in win rate and Nemotron score (-27.62 vs. -23.89), suggesting diminishing or even negative returns from additional fine-tuning without further refinement of prompt curation or teacher revisions.

Furthermore, we compared this to our baseline of implementing CITING without staged batch curricilum which had a training loss of 2.263 and a validation loss of 1.356. The average Nemotron score was -27.616, and it achieved a 46% win rate against DPO.

The results clearly demonstrate that the batched, length-based curriculum plays a critical role in improving model performance. Compared to the flat baseline, the curriculum-trained model achieved substantially better outcomes across all key metrics—lower training and validation loss, higher Nemotron scores, and a dramatically higher win rate against DPO (71% vs. 46%). By structuring the learning process to gradually handle more complex prompts, the curriculum enables the model to build stronger instruction-following capabilities in a more stable and data-efficient manner. This structured exposure appears to guide the model's learning trajectory more effectively than randomly mixing prompts of all lengths, validating the core hypothesis that curriculum learning leads to more robust generalization and higher-quality generations.

This comparison underscores two key insights:

- **Curriculum matters**: Even without multiple self-revision rounds, the staged curriculum alone yielded better loss metrics and stronger preference alignment than the no-curriculum baseline. The improvement in win rate (71% vs. 49%) and higher Nemotron score (-23.89 vs. -26.544) with curriculum demonstrates better generation quality and instruction-following capability.
- **Over-revision is harmful**: The performance drop in the second round suggests that excessive self-revision without improving prompt quality or feedback diversity may lead to overfitting or degradation. Rather than refining generation quality, it may reinforce subopti-

mal patterns from earlier revisions, confirming diminishing or even negative returns beyond one carefully guided revision step.

5.2 Qualitative Analysis

To further illustrate the improvements gained from preference optimization, we present a comparison of representative responses from the SFT and DPO models on two example prompts. These examples highlight how DPO improves response quality in terms of relevance, structure, and conversational tone.

> **Prompt:** *Why don't general physicians cover teeth?*

The SFT model responds:

> **SFT Response:** *The answer is simple: they don't want to. But that's not the only reason. There are many reasons why general physicians don't want to cover teeth. One reason is that they don't want to be seen as a giant or "big mouth" by patients.*

While the SFT model attempts a confident answer, it relies on speculative reasoning and includes awkward phrases that may undermine user trust. In contrast, the DPO model responds:

> **DPO Response:** *The answer lies in the fact that they are not as involved with oral health care. What's your take on why some doctors aren't covering dental services?*

The DPO output provides a more factually grounded and polite answer. It also re-engages the user with a follow-up question, demonstrating a more interactive and thoughtful generation style. Meanwhile, the self-revision model with one-round revisions during training responds:

> **Extension Response 1:***Why not just have a dental office for patients who need it?The answer is that the cost of treating oral health problems can be prohibitively expensive. In fact, in many cases, even dentists are forced to accept insurance coverage only if they offer comprehensive care. That's why most doctors and hospitals now require their staff members to take out dental insurance or pay an annual fee on top of their regular medical bills. This allows them to treat more complex conditions without having to worry about financial liability...*

This response is fully off-topic. Self-revision (especially when done only once) may improve surface-level clarity or fix obvious errors, but without stronger feedback loops (like DPO's preference signals), it may not fully align with human expectations of helpfulness, correctness, and tone. With two rounds of self-revision during training, it responds

> **Extension Response 2:***""*

Seemingly, it collapsed to an empty/invalid response, which can happen when revision loops overly suppress or prune content without adding new useful content. For the no-curriculum baseline, the model outputs:

> **Baseline:***Why not just have a dental office for the whole family? The answer is that it's too expensive. The average dentist charges $1,052 per year to treat one patient in their entire lifetime...So let us continue pushing forward together towards creating something truly remarkable - namely, a world where everyone...*
>
> This answer is off-topic, and continues to ramble for long afterwards, showing the advantage staged batch curriculum training provides in even shorter responses– helping the model stay on task and truncate when necessary.
>
> For another prompt, we find a similar qualitative pattern:
>
> > **Prompt:** *Which one is better for winter? Mesh boots or sandals?*
> > **SFT Response:** *I have a pair of mesh boots and a pair of sandals. Which one is better for winter? I have a pair of mesh boots and a pair of sandals. Which one is better for winter?...*

This response is repetitive and fails to offer useful information. On the other hand, the DPO model produces a structured comparison:

> **DPO Response:***I'd suggest considering the following options: 1. Sandals: Sandal footwear are great if you want something lightweight... 2. Mesh Boots: These mesh-like materials provide... some protection from wind chill effects... Also, could you please clarify why you asked for help regarding my recommendations?*

This output is more informative, organized, and user-aware. It not only compares the two options meaningfully but also asks a clarifying question, which aligns better with instruction-following goals.

These examples demonstrate that DPO enhances the model's ability to generate grounded, relevant, and context-aware responses, especially when compared to the more rigid and sometimes repetitive behavior of the SFT baseline.

Meanwhile, the self-revision model with one-round revisions during training responds:

> **Extension Response 1:**
> ***Mesh Boots***
>
> ***Comfort:*** *Some people find mesh boots more breathable due to their lightweight nature, offering less weight compared to traditional synthetic materials. However, many prefer the added insulation that certain mesh constructions provide.*
>
> *beginitemize*
>
> ***Pros:*** *1. Lightweight and well-ventilated 2. May provide additional thermal benefits when designed with specialized fabrics (e.g., Gore-Tex).*
> ***Cons:*** *1. May not provide sufficient UV protection; sunlight can penetrate through mesh 2. Can cause discomfort under extreme temperatures 3. In high elevations or very hot environments, may lead to overheating or rapid material degradation*
> ***Recommendations Based On Your Needs*** *Given that your mesh shoes are already waterproof and comfortable but lack insulation, consider: 1. Adding thermal insoles or warm socks to retain heat...*

The self-revision model significantly improves over the repetitive SFT baseline by producing a structured, informative response focused on mesh boots. It lists pros and cons, offers winter-specific recommendations, and maintains coherence. Instead of asking for why the client needed help, which is less useful, the extension was able to finalize reccomendations at the end. Then, with two rounds of self-revision during training, it responds:

> **Extension Response 2:** *"'python*
> *Revised response based on user input*

Again, the second round produces a response that's both meaningless and unrelated. This degradation is due to catastrophic forgetting—a phenomenon where repeated instruction tuning causes the model to forget knowledge learned during the initial Supervised Fine-Tuning (SFT) phase. As a result, excessive revision may harm overall performance rather than improve it.

For the no-curriculum baseline, the model outputs:

> **Baseline:***cooperation collaboration community shared collective sharing collaborative teamwork teamwork partnership partnerships cooperative cooperatives social networks networked connected networking communication exchange exchanges*

This response is completely nonsensical and repetitive, revealing the limitations of flat training without structured exposure. The model appears to latch onto a semantic cluster without understanding the instruction or producing a coherent completion. In contrast, the staged batch curriculum—by isolating short prompts during earlier training stages—gives the model more focused opportunities to learn concise, task-relevant behaviors. It helps the model internalize the structure and intent of shorter prompts before introducing longer or more complex ones. This progression minimizes distraction from overly diverse input lengths and enables the model to better learn when to stop, how to stay on topic, and how to balance informativeness with brevity. Ultimately, the batched curriculum enforces a form of length-aware regularization that flattens the learning curve and encourages cleaner, more context-appropriate generations across all tiers.

Our experiments show that a single round of self-revision with our curriculum-based CITING framework outperformed the DPO baseline, as measured by average Nemotron reward scores and qualitative inspection of generations. However, performance dropped with two rounds of self-revision, suggesting that repeated imitation of synthetic revisions may lead to overfitting or drift away from instruction intent.

While the original CITING paper observed diminishing returns only after four rounds of revision on a larger model, our smaller 0.5B model showed sensitivity immediately after the first round, suggesting that smaller models may be more vulnerable to overfitting on synthetic feedback.

Compared to flat DPO tuning, generations produced after curriculum-based CITING were more coherent, better aligned with task objectives, and exhibited clearer reasoning—reflected both in higher Nemotron scores and in qualitative assessments. These results underscore the value of using prompt length as a proxy for difficulty, helping the model progressively acquire revision skills suited to varying instruction types. Together, our findings highlight the importance of structured exposure and moderated revision depth in maximizing the benefits of teacher-guided alignment for compact language models.

6 Discussion

In addition to validating the effectiveness of curriculum-based self-revision, our results reveal an important trade-off between revision depth and model alignment: while a single round of revision helps correct key errors and improve structure, further rounds can inadvertently homogenize responses or reduce responsiveness to nuanced instructions. This suggests that revision should be adaptive, not fixed—potentially guided by prompt complexity or model uncertainty. This aligns with findings in the original CITING paper, which noted diminishing returns after a few revision rounds Feng et al. (2023). It emphasizes the need for moderation in iterative training and hints at a potential ceiling in gains from repeated teacher-guided supervision alone.

Moreover, the success of our length-based curriculum highlights a broader insight: structuring training around input characteristics, rather than task labels or heuristics, can meaningfully enhance generalization in instruction tuning. This opens up new possibilities for designing scalable alignment techniques that adjust training difficulty based on measurable input features—offering an interpretable and efficient alternative to more opaque reward-driven methods like RLHF.

While our method shows promising improvement over SFT and demonstrates early gains with CITING, several limitations emerged. First, the use of GPT-4o-mini as a teacher LLM—though efficient and cost-effective—introduces potential variability in revision quality, especially for longer or more nuanced prompts. The self-revision pipeline also requires repeated inference calls, making it resource-intensive at scale. Furthermore, while DPO substantially improves alignment, we observed that it sometimes favors engagement over clarity by inserting questions in contexts where concise answers are expected. This behavior may reduce user satisfaction in task-oriented applications.

Another major challenge during the project was efficiently managing the scale of CITING training. Repeated generation, revision, and fine-tuning at multiple curriculum stages demanded careful batching and memory handling, particularly given limited GPU resources. Despite these obstacles, the iterative training loop—combined with a structured curriculum—proved intuitive and modular to scale, and we found that even small models can benefit from recursive, staged improvements.

Evaluation-wise, our reliance on the Nemotron-70B reward model offers a scalable benchmark but cannot fully substitute for human judgments. Reward models may favor stylistic traits or structures learned during their own training, potentially biasing evaluation toward certain models. Additionally, our current CITING evaluations are still in progress, limiting the completeness of our conclusions on the full impact of self-revision and curriculum learning.

This work highlights how compact language models can be fine-tuned with prompts and synthetic data to achieve meaningful improvements in instruction-following. Our approach emphasizes accessible, scalable methods that reduce dependence on proprietary APIs or massive compute. If extended, this framework may offer low-cost personalization for LLMs in educational, assistive, or domain-specific settings.

That said, the ability for models to revise themselves raises ethical questions. Poorly calibrated self-revision may introduce hallucinations or amplify biases if unchecked. Ensuring that teacher models and revision criteria are curated carefully is critical for safe deployment. Future research should explore how revision dynamics affect model factuality, safety, and transparency.

7 Conclusion

In this work, we introduced a multi-stage alignment framework that combines Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), and a curriculum-structured self-revision extension of the CITING algorithm. Using the Qwen 2.5–0.5B model, we demonstrated that compact, open-weight models can achieve meaningful improvements in instruction-following through scalable, accessible techniques. A key insight of our approach is the combination of length-based curriculum learning with guided self-revision: by structuring training from short to long prompts, we aligned model learning with task complexity, allowing it to develop core revision behaviors before tackling more challenging instructions. This progression stabilized training, improved coherence, and helped the model generalize where DPO alone tended to overfit.

The self-revision component, adapted from CITING, further strengthened alignment by supplying task-specific feedback from a teacher LLM. Crucially, we found that one round of self-revision improved performance, while additional rounds led to degradation—underscoring the need for moderation and targeted revision depth. Together, the curriculum and revision framework yielded more on-topic, structured generations and higher Nemotron reward scores, highlighting their complementary effects. Looking ahead, we plan to complete CITING evaluation, explore adaptive curriculum strategies beyond prompt length, and extend this framework to domain-specific applications where efficient, fine-grained alignment is essential.

8 Team Contributions

- **Landon Choy:** Led the implementation of supervised fine-tuning (SFT) on the Smol-Smoltalk dataset, contributed to the Direct Preference Optimization (DPO) training on the UltraFeedback-Binarized dataset, and assisted in the development of the Self-Revision extension.
- **Tracy Li:** Developed the evaluation pipeline using the LLaMA 3.1 Nemotron-70B reward model, contributed to the DPO training and fine-tuning process, and played a key role in designing and implementing the Self-Revision extension, including staged curriculum learning and prompt categorization.

Changes from Proposal Our final project deviated from the original proposal, which was centered on designing a multi-objective reward model. Based on early experimentation and emerging challenges, we pivoted to a recursive self-revision framework inspired by CITING. This shift allowed us to focus more deeply on improving instruction-following performance through teacher-guided feedback and curriculum-based refinement.

References

Yejin Bang and et al. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. *arXiv preprint arXiv:2212.08073* (2022).

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 41–48.

Tao Feng, Zifeng Wang, and Jimeng Sun. 2023. CITING: Large Language Models Create Curriculum for Instruction Tuning. *arXiv preprint arXiv:2310.02527* (2023).

Rafael Rafailov, Deep Ganguli, Rajiv Ramamurthy, Rose E Zhang, Joseph E Gonzalez, and Tatsunori B Hashimoto. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv preprint arXiv:2305.18290* (2023).

Paul Soviany, Radu Tudor Ionescu Andrei, Alex, and Marius Leordeanu. 2021. Curriculum learning: A survey. *International Journal of Computer Vision* 129, 3 (2021), 513–548.

Jason Wei, Xuezhi Zhou, Denny Wang, and et al. 2023. Language models are reasoning teachers. *arXiv preprint arXiv:2305.06983* (2023).