ITERATIVE LABEL REFINEMENT MATTERS MORE THAN PREFERENCE OPTIMIZATION UNDER WEAK SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

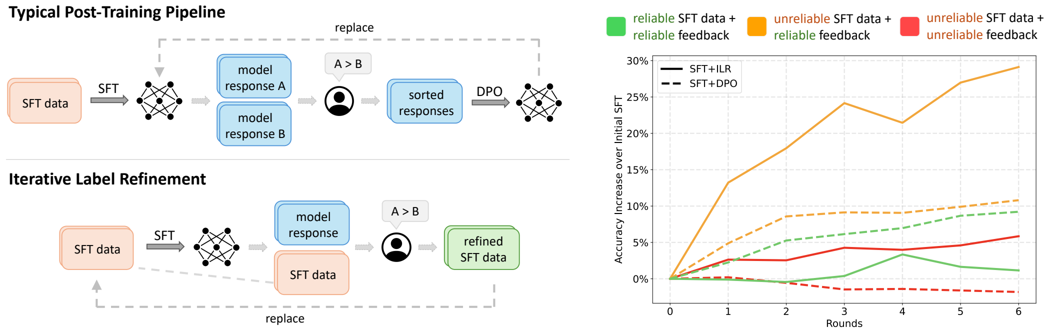
Language model (LM) post-training relies on two stages of human supervision: task demonstrations for supervised finetuning (SFT), followed by preference comparisons for reinforcement learning from human feedback (RLHF) via algorithms like proximal preference optimization (PPO) or direct preference optimization (DPO). As LMs become more capable, the tasks they are given become harder to supervise. Will post-training remain effective under unreliable supervision? To test this, we simulate unreliable demonstrations and comparison feedback using small LMs and time-constrained humans. We find that in the presence of unreliable supervision, SFT still retains some effectiveness, but DPO fails to improve the model beyond SFT. To address this, we propose *iterative label refinement* (ILR) as a replacement for RLHF with unreliable supervision. ILR directly improves the SFT data by using comparison feedback to decide whether human demonstrations should be replaced by model-generated alternatives, then retrains the model via SFT on the updated data. SFT+ILR outperforms SFT+DPO on several tasks with LM-simulated unreliable supervision (math, coding, safe instruction-following), with results further verified by human experiments on instruction-following. Our findings suggest that as LMs take on complex tasks where human supervision is unreliable, RLHF may no longer be the best use of human comparison feedback; instead, it is better to direct feedback towards improving the training *data* rather than continually training the *model*.

1 INTRODUCTION

Language models (LMs) learn rich knowledge when pretrained on internet-scale corpora (Achiam et al., 2023; Dubey et al., 2024). To elicit their full capabilities and align them to human values, they are typically post-trained with two types of human supervision: task *demonstrations* for the initial supervised finetuning (SFT) stage, followed by preference *comparisons* used in reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022) via algorithms like proximal policy optimization (PPO) (Schulman et al., 2017) or direct preference optimization (DPO) (Rafailov et al., 2024; Dubey et al., 2024).

As LMs continue to advance, they are trained to solve complex tasks that are difficult for humans to supervise (Amodei et al., 2016; Leike & Sutskever, 2023; Wen et al., 2024). For example, current models like Github Copilot learn bugs from human-written code (Asare et al., 2023), and industrial-level ChatGPT training data rated as “flawless” has recently been found to contain flaws (McAleese et al., 2024). As task complexity increases, human supervision may become even less reliable. Thus, it is imperative to understand how both stages of current post-training pipelines (SFT + RLHF) perform under unreliable supervision.

Studying this is difficult because tasks where human supervision is unreliable are by nature difficult to obtain ground truth for. Ideally, we want tasks with known ground truth where we can also collect human-like unreliable data. We employ two approaches to simulate this unreliable supervision. First, we use smaller LMs that often make mistakes, in line with Burns et al. (2023) and Dubois et al. (2024). Second, we recruit human workers to label data under time constraints. We refer to the two forms of supervision collected via either approach as *unreliable demonstrations* and *unreliable comparisons*, respectively. Then, we simulate post-training under unreliable supervision by running SFT on unreliable demonstrations followed by DPO with unreliable comparisons.



(a) (b)

Figure 1: (a) In contrast to RLHF, which iteratively updates the SFT *model*, ILR directs comparison feedback towards improving the SFT *dataset*. (b) With reliable supervision, DPO effectively improves the SFT model and ILR is unnecessary. However, DPO become less effective with unreliable demonstrations and breaks down completely when comparison feedback is also unreliable. In contrast, ILR consistently improves on the initial SFT model with unreliable supervision, even in the most extreme case.

We find models trained via SFT on unreliable demonstrations have higher test accuracy than the small model that generated their training data, a phenomenon termed weak-to-strong generalization (W2SG) that has been previously observed by Burns et al. (2023) in classification tasks. However, since these models inevitably learn to imitate errors in their unreliable training data, the resulting performance is still far from their full capability when trained on ground truth. Thus, ideally, the subsequent DPO stage should improve these SFT models.

Unfortunately, we find DPO breaks down under unreliable supervision. With unreliable demonstrations and comparisons, DPO offers little to no performance improvement on top of SFT (Figure 1b). Our experiments suggest that this may be due to the tendency of DPO to overoptimize (Gao et al., 2023; Azar et al., 2024). DPO implicitly penalizes KL divergence from the SFT model, and we find that a large penalty is necessary to prevent overoptimization when feedback is unreliable. However, this large penalty prevents the large changes in the model that are necessary to correct the errors learned from the unreliable demonstrations during SFT (Section 4.1).

To address this, we propose *iterative label refinement* (ILR). In contrast to RLHF, which iteratively updates the SFT *model*, ILR updates the SFT *dataset* using the same type of comparison feedback. Specifically, ILR uses this feedback to compare a demonstration in the SFT dataset with an alternative one written by the SFT model; if the model-written demonstration is chosen, it replaces the original demonstration in the SFT data. This improves the SFT data by replacing low-quality or incorrect demonstrations with better model-written ones. Thus, when a new SFT model is trained on the updated dataset, it should perform better than the original SFT model, facilitating further improvements to the dataset via additional rounds of ILR (Figure 1a). Unlike DPO, we find that updating the SFT data via ILR leads to significant improvements from one round to the next (Figure 1b).

SFT+ILR improves over SFT+DPO in several tasks, including math, coding, and safe instruction following. To further confirm these results, we perform a human study where we recruit workers to provide demonstrations and comparison feedback under time constraints for the Alpaca (Taori et al., 2023) instruction-following task. In this setting with more realistic human errors, ILR continues to be effective in improving the initial unreliable human demonstrations and outperforms DPO.

In summary, our findings suggest that preference-based training like RLHF may no longer be the best use of comparison feedback as we train LMs to solve complex tasks where human supervision is unreliable. Instead, it is better to direct human feedback towards fixing errors in the unreliable training *data* rather than continually training the *model*.

2 Related Work

Scalable oversight. Scalable oversight aims to develop methods to supervise AI on tasks that are difficult for humans (Amodei et al., 2016; Bowman et al., 2022). The primary focus in scalable oversight has been on designing human-AI collaboration mechanisms that enhance humans' ability to evaluate AI outputs more accurately (Christiano et al., 2018; Irving et al., 2018; Michael et al., 2023; Khan et al., 2024; Kenton et al., 2024) and with less cognitive burden (Saunders et al., 2022; McAleese et al., 2024; Kirchner et al., 2024). In contrast, recent work on weak-to-strong generalization (W2SG) (Burns et al., 2023) explores a complementary direction that emphasizes developing better learning algorithms capable of training models that generalize beyond limited weak supervision as if they were trained on higher-quality supervision or even ground truth. The SFT stage in our LM-based simulation employs a similar setting as the setup used for studying W2SG in Burns et al. (2023) but extended to text generation tasks; Burns et al. (2023) only consider classification tasks.

Language model post-training. Pretrained language models (PLMs) possess a rich understanding of language but require post-training to align their behavior with human preferences. This generally involves supervised finetuning (SFT) (Wei et al., 2021) on human written demonstrations and then reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022). RLHF is typically done by algorithms like PPO (Schulman et al., 2017), which uses an explicit reward model, or DPO (Rafailov et al., 2024; Dubey et al., 2024) and its variants (Liu et al., 2023; Azar et al., 2024), which uses and implicit reward model. Recent work has also investigated the impact of noisy preference data on preference optimization (Chowdhury et al., 2024; Fisch et al., 2024; Gao et al., 2024), while our work addresses a more challenging scenario where both SFT demonstrations and preference data are unreliable.

Learning with imperfect supervision. Research in weakly-supervised learning has long focused on training models with noisy labels (Reed et al., 2014; Rolnick et al., 2017; Song et al., 2022). Our LM-simulated setup and time-constrained human studies differ from traditional label noise studies, as weak demonstrations and comparison feedback generated by LMs may contain structured errors rather than uniformly random noise, such as mistakes caused by specific misconceptions or biases. The cross-labeling framework in our proposed method shares conceptual similarities with co-teaching (Han et al., 2018) and co-training (Blum & Mitchell, 1998) in semi-supervised learning, although these methods primarily focus on classification settings and rely on accurate model confidence scores.

3 Problem Definition: Post-Training With Unreliable Supervision

Training LMs to be useful assistants involves two stages: pre-training and post-training. During pre-training, the LM is trained to autoregressively predict a large corpus of text, typically taken from the internet and books (Radford et al., 2019; Brown et al., 2020). Afterwards, the pretrained language model (PLM) has acquired knowledge and skills from its pre-training data, but cannot yet be easily used for realistic applications. To elicit the PLM's capabilities for some class of tasks, post-training techniques are used (Wei et al., 2021; Dubey et al., 2024; Ouyang et al., 2022; Bai et al., 2022). We denote an LM as a conditional distribution $p(y \mid x)$ over responses y given a prompt x.

Standard post-training pipeline. Post-training a PLM typically involves two stages. In the initial SFT stage, a PLM is trained on human-written task demonstrations. That is, given an SFT dataset $\mathcal{D}_{\text{SFT}} = (\mathcal{X}_{\text{SFT}}, \mathcal{Y}_{\text{SFT}})$ of prompts $\mathcal{X}_{\text{SFT}}$ and human-written responses $\mathcal{Y}_{\text{SFT}}$, the PLM is trained to minimize $-\frac{1}{|\mathcal{D}_{\text{SFT}}|}\sum_{(x,y)\in(\mathcal{X}_{\text{SFT}},\mathcal{Y}_{\text{SFT}})} \log p(y \mid x)$. We call the result of this stage the SFT model, which has typically learned the correct input/output format for the given task and has partially elicited the PLM's capabilities.

Subsequently, to further improve performance, the SFT model undergoes RLHF via algorithms like PPO (Ouyang et al., 2022; Bai et al., 2022; Schulman et al., 2017), DPO (Rafailov et al., 2024; Dubey et al., 2024), or related methods. In this stage, a new dataset is constructed using a set of prompts $x \in \mathcal{X}_{\text{RLHF}}$ and response pairs (y_1, y_2) sampled from the SFT model. Each prompt and pair of responses are shown to a human annotator that picks the response they prefer, creating a dataset of triples $(x, y_+, y_-) \in \mathcal{D}_{\text{RLHF}}$, where y_+ is preferred to y_-. This dataset is used for preference optimization via methods like PPO, which trains a reward model from $\mathcal{D}_{\text{RLHF}}$ for online RL, or DPO, which uses an implicit reward model. This RLHF stage typically increases the probability that the SFT model generates preferred responses. In this work, we focus on DPO due to its computational efficiency and stability. We call the output of the this stage the SFT+DPO model.

Simulating unreliable human supervision To study how current post-training pipelines perform under unreliable supervision, we need tasks with known ground truth so that we can evaluate performance. We also need supervision that is unreliable in the way human oversight may be when supervising superhuman systems on complex tasks. We employ two approaches to simulate this. For most of our experiments (Section 4 and 5), we focus on a small-LM-based simulation, in line with Burns et al. (2023) and Dubois et al. (2024). Then, we verify our results in a more realistic setup with time-constrained human annotators (Section 6).

Our LM-based simulation consists of two components (more details in Appendix B):

1. *Unreliable task demonstrations*: We finetune a small PLM $\tilde{p}$ on ground-truth demonstrations and use it to generate responses for a held-out set of prompts.
2. *Unreliable comparison feedback*: We finetune a classification model $\tilde{q}$ to select the better response when given two responses y_1, y_2 and a prompt x, where $\tilde{q}(y_1 \succ y_2 \mid x)$ denotes the probability that y_1 is preferred to y_2. Note that $\tilde{q}$ is based on the same PLM as $\tilde{p}$ but with an additional classification head. Each element of $\tilde{q}$'s training data consists of a prompt and two responses: one response is from the ground-truth demonstrations used to train $\tilde{p}$ and the other is a lower-quality output generated by intermediate checkpoints of $\tilde{p}$. $\tilde{q}$ is trained to choose the ground truth using a standard binary classification loss.

SFT+DPO with unreliable supervision. We start by finetuning a larger PLM on an unreliable SFT dataset $\widetilde{\mathcal{D}}_{\text{SFT}} = (\mathcal{X}, \widetilde{\mathcal{Y}})$ to obtain an initial SFT model $\widehat{p}_{\text{SFT}}$, where $\widetilde{\mathcal{Y}}$ is a set of unreliable demonstrations given by $\tilde{p}$ or time-constrained humans. We then perform K rounds of DPO training. In each round k, we sample completions y_1, y_2 from the current SFT+DPO model $\widehat{p}^{\,k-1}_{\text{SFT+DPO}}$ for each prompt $x \in \mathcal{X}$ (where $\widehat{p}^{\,0}_{\text{SFT+DPO}} \equiv \widehat{p}_{\text{SFT}}$), collect unreliable comparisons using either $\tilde{q}$ or time-constrained humans to construct an unreliable preference dataset $\widetilde{\mathcal{D}}^{k}_{\text{RLHF}}$, and use this to train $\widehat{p}^{\,k-1}_{\text{SFT+DPO}}$ via DPO, resulting in $\widehat{p}^{\,k}_{\text{SFT+DPO}}$. The final model after K rounds is denoted as $\widehat{p}^{\,K}_{\text{SFT+DPO}}$ or simply $\widehat{p}_{\text{SFT+DPO}}$. Further implementation and hyperparameter details can be found in Appendix C.

3.1 Tasks and Models

Datasets. We consider three text generation tasks, including mathematical problem-solving using GSM8K (Cobbe et al., 2021), SQL code generation with BIRD (Li et al., 2024), and safe instruction following with SaferPaca (Bianchi et al., 2023), which is a mix of the Alpaca dataset (Taori et al., 2023) and refusal demonstrations to unsafe instructions. All datasets are formatted as question-answer pairs following the template `USER:\n{question}\n\nASSISTANT:{answer}\n`.

Models. We use three open-source PLMs of varying sizes in our experiments: Gemma 2B (Team et al., 2024), Mistral 7B (Jiang et al., 2023), and Meta Llama 3 70B (Dubey et al., 2024). Specifically, our LM-simulated experiments include four settings. For GSM8K, we experiment both with using the 2B model to supervise the 7B model, and using the 7B model to supervise the 70B model. For SaferPaca and BIRD, we consider 2B supervising 7B and 2B supervising 70B, respectively, to balance limited compute while maintaining sufficient performance gaps between models. In our simulation with time-constrained humans, we only experiment with the largest 70B model.

Evaluation metrics. For GSM8K, we parse the final numerical answer and compute exact match accuracy against the ground truth. For BIRD, we measure execution accuracy by running the generated code on corresponding test databases, following Li et al. (2024). For SaferPaca, we follow Li et al. (2023) and use GPT-4o (OpenAI, 2024) to compute win rate against reference answers,

Further details on datasets, evaluation metrics, model architectures, training and inference configurations are provided in Appendix A, B, and C.

4 DPO Is Ineffective with Unreliable Supervision

In this section, we present results of SFT+DPO under LM-simulated unreliable supervision. We find that SFT still retains some effectiveness, but DPO fails to improve the SFT model. We further show that the failure of DPO appears to be caused by its overoptimization problem with unreliable comparison feedback, which motivates our alternative approach introduced in the next section.

SFT shows limited robustness to unreliable demonstrations. We first investigate the effectiveness of SFT under LM-simulated unreliable supervision. When finetuned on $\widetilde{\mathcal{D}}_{\text{SFT}}$ with unreliable demonstrations given by $\tilde{p}$, the larger model $\widehat{p}_{\text{SFT}}$ consistently outperforms $\tilde{p}$ across all tasks, demonstrating

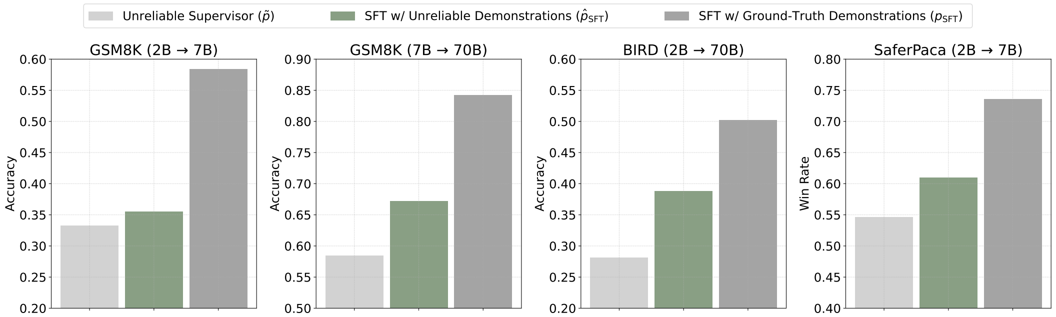
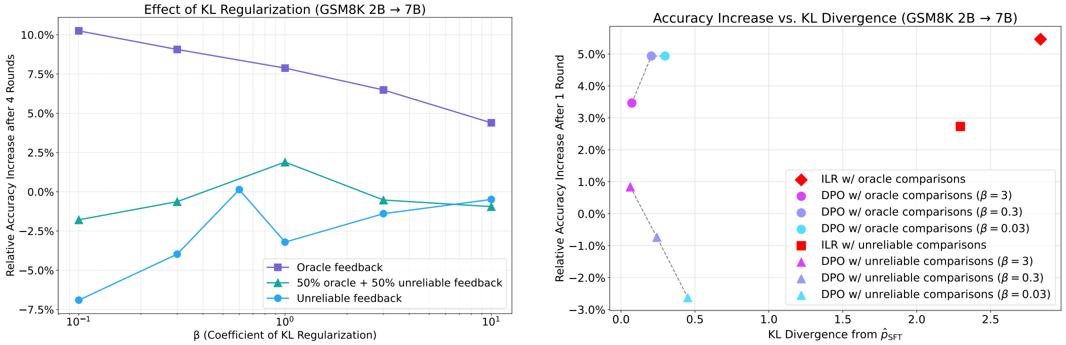


Figure 2: Models finetuned on unreliable demonstrations generated by a smaller supervisor model show higher accuracy than the supervisor, demonstrating SFT's robustness to unreliable supervision. However, SFT models' full capability is only partially recovered with unreliable demonstrations. A $\rightarrow$ B denotes using demonstrations generated by the model with size A to finetune the model with size B.



(a) DPO effectively improves SFT models with oracle comparisons, especially with weak regularization. However, with unreliable feedback, improvements are limited to a narrow range of regularization strengths because weak regularization causes overoptimization.

(b) Strong regularization in DPO limits useful model updates, while small regularization leads to overoptimization of unreliable feedback. In contrast, ILR facilitates large model updates, allowing for faster improvement and efficient use of comparison feedback.

Figure 3: DPO struggles to avoid overoptimization of noisy preferences while also updating the initial suboptimal SFT model sufficiently to improve performance.

SFT's robustness to imperfect demonstrations (Figure 2). This result in our text generation tasks aligns with the weak-to-strong generalization phenomenon observed by Burns et al. (2023) in classification tasks. However, models finetuned on unreliable demonstrations inevitably learn to imitate errors in them. When we finetune the same PLM on ground-truth demonstrations, which we denote p_{SFT}, we find that it significantly outperforms $\widehat{p}_{\text{SFT}}$ (Figure 2). This shows that SFT on unreliable demonstrations only partially recovers a model's full capability, leaving a large performance gap between p_{SFT} and $\widehat{p}_{\text{SFT}}$. Thus, ideally, the subsequent DPO stage should further improve upon $\widehat{p}_{\text{SFT}}$.

DPO does not improve over SFT with unreliable comparisons. Contrary to DPO's past success (Rafailov et al., 2024; Dubey et al., 2024), we find that it consistently fails to improve performance over SFT across all tasks when both demonstrations and comparison feedback are unreliable (Figure 4). That is, when we further finetune $\widehat{p}_{\text{SFT}}$ via multiple rounds of DPO on comparison feedback given by $\tilde{q}$, the resultant model $\widehat{p}_{\text{SFT+DPO}}$ shows little to no performance gain (or performance even declines). To investigate the degree to which this is due to the unreliable comparison feedback versus the unreliable SFT data, we also run DPO in two other settings on the GSM8K task. In the first, we train starting with $\widehat{p}_{\text{SFT}}$ but with an oracle evaluator that always selects the better answer by comparing model outputs to ground truth. In the second, we again use the oracle evaluator, but start from an SFT model trained on a filtered subset of $\widetilde{\mathcal{D}}_{\text{SFT}}$ that only contains correct demonstrations. As shown in the right of Figure 1b, DPO with the reliable oracle evaluator does manage to improve over the starting SFT model in both cases. These results reveal that DPO's effectiveness heavily relies on high-quality supervision.

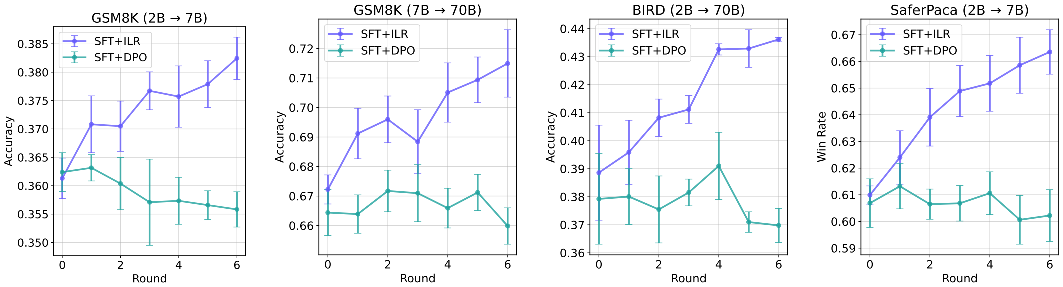


Figure 4: ILR consistently provide more improvements for the SFT model than DPO in four settings with LM-simulated unreliable demonstrations and comparison feedback. Round 0 represents accuracy of $\widehat{p}_{\text{SFT}}$.

4.1 DPO with Unreliable Feedback Exhibits Overoptimization

We hypothesize that DPO's failure under unreliable supervision stems from the problem of overoptimization that preference-based learning methods often suffer from (Gao et al., 2023). Optimizing an explicit reward function (in RLHF with PPO) or an implicit one (in DPO) based on preference feedback can initially lead to an increase in performance but eventually causes a decline as the reward function diverges from the true objective. To combat this, both PPO and DPO regularize based on the KL divergence between the initial and updated policies. We find that the optimal amount of regularization for DPO, which is controlled by the hyperparameter β, depends strongly on the reliability of comparison feedback.

To demonstrate this, we measure DPO's relative improvement over SFT under varying feedback quality levels and KL regularization strengths (Figure 3). More results are presented in Figure 8.

With unreliable comparison feedback, DPO is especially prone to overoptimization and needs very strong regularization. In Figure 3a, we consider three feedback quality levels of comparison feedback: unreliable feedback from $\tilde{q}$, mixed feedback (50% $\tilde{q}$, 50% oracle), and pure oracle feedback. With oracle feedback, smaller β values consistently yield better performance. However, as feedback quality decreases, only a suitably large β enables positive improvements. Smaller β values in the presence of unreliable feedback lead to worse performance, suggesting that unreliable feedback may cause overoptimization without adequate regularization.

Strong regularization limits useful model updates during DPO training. Figure 3b demonstrates that larger divergence correlates with higher accuracy increases in the case of oracle feedback. This indicates that setting small β to allow substantial model updates is crucial for improving upon the initial suboptimal SFT model. However, with unreliable feedback, this is not possible due to the above observation that insufficient regularization with unreliable comparisons leads to overoptimization.

These two observations suggests a tension between preventing overoptimization of unreliable comparison feedback and making large enough model updates. Since unreliable demonstrations in the SFT data lead to an SFT model $\widehat{p}_{\text{SFT}}$ that is quite suboptimal, it is necessary to update $\widehat{p}_{\text{SFT}}$ considerably to recover performance close to p_{SFT}. However, with unreliable comparison feedback, significant regularization (*i.e.*, large β) needs to be applied to DPO, which limits improvements beyond the poor performance of the SFT model.

Motivating an alternative method to use comparison feedback. Given the limitations of preference optimization under unreliable supervision, it would be ideal if there was an alternative that could better use comparison feedback to improve over the SFT model. Such an alternative would need to both be more robust to unreliable comparisons and allow larger updates from the SFT model. As we argue in the following section, one way to achieve both aims is to direct comparison feedback towards updating the SFT *dataset* rather than the SFT *model*.

5 Iterative Label Refinement

Given the limitations of preference optimization under unreliable supervision, we propose *iterative label refinement* (ILR) as an alternative method for improving an SFT model with unreliable comparison feedback. In contrast to RLHF methods, which iteratively updates the SFT *model*, ILR focuses on improving the unreliable demonstrations in the initial SFT *dataset*. In this section, we introduce how ILR works and demonstrate its superiority over DPO in the LM-simulated setting. ILR avoids the issues with preference optimization discussed in Section 4.1 and allows for large changes to the initial SFT model by updating the training data and retraining. Further comparison of ILR and DPO using time-constrained human data is presented in Section 6.

5.1 Methodology

In ILR, comparison feedback is used for comparing two responses of a prompt $x \in \widetilde{\mathcal{D}}_{\text{SFT}}$: the existing response $\tilde{y}$ in $\widetilde{\mathcal{D}}_{\text{SFT}}$ and a new one generated by an SFT model. If the annotator chooses the model-generated response, it replaces the original response in the SFT data. Once the annotator has considered each model-generated response to each prompt, an improved SFT model is trained on the updated dataset. As long as the evaluator chooses better responses more than half the time, the overall accuracy of the SFT data will increase.

It is possible to obtain these better responses for improving the SFT data because of the W2SG phenomenon observed in Section 4, where models finetuned on unreliable demonstrations outperform their unreliable supervisor. This suggests that on a held-out set, where the SFT model hasn't directly learned to imitate incorrect answers, it can potentially generate more accurate labels than its supervisor. However, if the SFT model is trained on the entire SFT dataset, then this is not possible: its newly proposed labels will contain mistakes imitated or memorized from the SFT data. Thus, we implement ILR by training *two* SFT models on different halves of the SFT data; then, these models cross-label the half they were not trained on. This makes it more likely the model-generated responses are different and better than the existing responses, enabling improvement of the dataset.

The label refinement process. Formally, each iteration of ILR consists of the following steps:

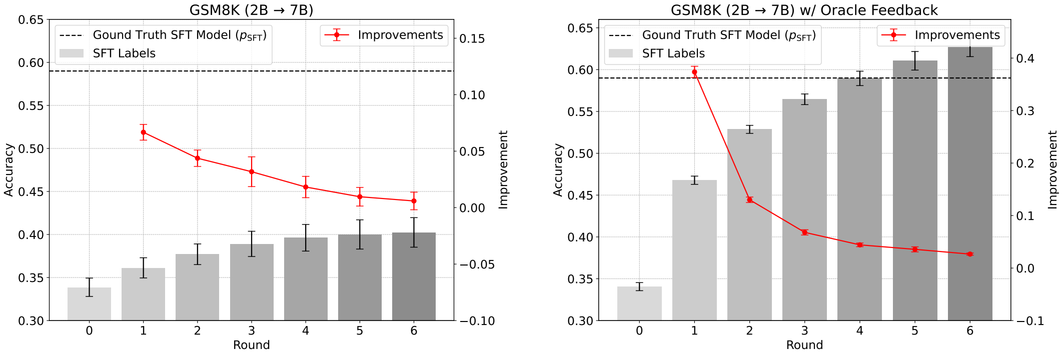
1. *Data splitting*: We start with a dataset of task demonstrations $\mathcal{D}_k = (\mathcal{X}, \tilde{\mathcal{Y}}_k)$, where k is the current iteration and $\mathcal{D}_0 \equiv \widetilde{\mathcal{D}}_{\text{SFT}}$. The dataset $\mathcal{D}_k$ is evenly split into two disjoint subsets, $\mathcal{D}_k^1 = (\mathcal{X}^1, \mathcal{Y}_k^1)$ and $\mathcal{D}_k^2 = (\mathcal{X}^2, \mathcal{Y}_k^2)$.
2. *Model training*: Two models, $\widehat{p}^{\,1}_{\text{SFT}}$ and $\widehat{p}^{\,2}_{\text{SFT}}$, are finetuned on $\mathcal{D}_k^1$ and $\mathcal{D}_k^2$ respectively.
3. *Cross-labeling*: Each model generates label proposals for the subset it wasn't trained on, *i.e.*, we sample $\mathcal{Z}_k^2 \sim \widehat{p}^{\,1}_{\text{SFT}}(\cdot \mid \mathcal{X}^2)$ and $\mathcal{Z}_k^1 \sim \widehat{p}^{\,2}_{\text{SFT}}(\cdot \mid \mathcal{X}^1)$.
4. *Proposal evaluation*: The unreliable supervisor provides comparison feedback on the new proposal z_i and original label $\tilde{y}_i$ for a prompt x_i to decide which of them is preferred, deciding whether $\tilde{y}_i$ should be replaced by z_i.
5. *Label updating*: Based on the comparisons in step 4, we form an updated dataset $\mathcal{D}_{k+1} = (\mathcal{X}, \widetilde{\mathcal{Y}}_{k+1})$, where each element of $\widetilde{\mathcal{Y}}_{k+1}$ is either the accepted new proposal or the retained original label. We control the label refinement speed by setting a hyperparameter α, allowing at most $\alpha|\mathcal{X}|$ labels to be updated in each iteration. When more than $\alpha|\mathcal{X}|$ proposals are accepted in step 4, we choose the ones with higher evaluation confidence, determined by either $\tilde{q}$'s log probability or human annotators' self-reported confidence. In all experiments we set $\alpha = 0.15$. Further analysis of α is presented in Appendix E.3.

At the end of each iteration, we finetune a new model starting from the base PLM of $\widehat{p}_{\text{SFT}}$ using the entire refined dataset $\mathcal{D}_K$, resulting in the SFT+ILR model $\widehat{p}^{\,k}_{\text{SFT+ILR}}$. This process is repeated for K iterations and the final model is $\widehat{p}^{\,K}_{\text{SFT+ILR}}$.

In the LM-simulated setting, we only consider proposals that are sufficiently different from the original labels during step 4. This is determined by having different final answer (GSM8K), different execution result (BIRD), or large embedding distance (SaferPaca). We present the pseudo-code (Algorithm 1) and more implementation details of ILR in Appendix C.

5.2 LM-simulated Experiment Results

SFT+ILR leverages unreliable supervision better than SFT+DPO. We compare SFT+ILR and SFT+DPO across four settings with LM-simulated unreliable demonstrations and comparison feedback (further validation using time-constrained human data is presented in Section 6). As shown in



(a)

(b)

Figure 5: SFT label accuracy increases across rounds during ILR. With high-quality feedback, ILR can lead to labels that approach or even surpass the accuracy of the model trained on ground truth (p_{SFT}).

Figure 4, SFT+ILR consistently outperforms SFT+DPO in all scenarios under unreliable supervision. In GSM8K, ILR shows more significant improvements as model size increases from 7B to 70B, suggesting that it benefits from model scaling and may remain effective for future models that are even more capable.

Comparison feedback is effective for improving demonstration quality. ILR relies on refining SFT data using comparison feedback, which is often easier to obtain than high-quality demonstrations, especially for complex tasks that humans struggle with. Figure 5a shows that unreliable comparison feedback guides this refinement process effectively: the accuracy of the SFT data steadily increases with more rounds of ILR. As suggested, this is likely because evaluation of AI output is typically easier than the demonstration of an ideal output (Leike et al., 2018), allowing even imperfect feedback to contribute to meaningful improvements in the SFT data. With reliable feedback (Figure 5b), the SFT demonstrations can be improved even further, approaching or even surpassing the accuracy of the model $\widehat{p}_{\mathrm{SFT}}$ trained on ground truth. This highlights ILR's potential when combined with other scalable oversight techniques that enhance humans' evaluation capability.

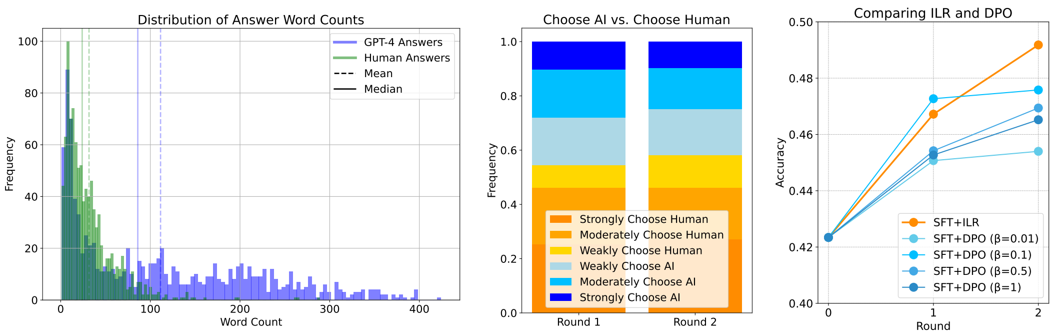
ILR enables larger model updates and more efficient use of comparison feedback. As illustrated in Figure 3b, models trained with ILR exhibit significantly higher KL divergence from the initial SFT model compared to those trained with DPO even after a single round. Moreover, when comparison feedback is reliable while SFT data is not, ILR shows much higher efficiency than DPO, using the same amount of feedback to enable larger improvement over the initial SFT model. These can be attributed to ILR's direct modification of the unreliable SFT data, which fundamentally alters models' training dynamics during SFT. By doing so, ILR allows each subsequent model to learn from new data independently and from scratch, without inheriting the errors of previous models, unlike the continual preference-based training seen in DPO.

Supervision for refinement is necessary. Since W2SG means that the SFT model typically outperforms its training data, it appears that it might be possible to simply replace the initial SFT data with new proposals without using any comparison feedback. To understand the importance of comparison feedback within ILR's refinement process, we compare it to a naive approach that directly replaces an α fraction of the original labels with new proposals in each round. As shown in Figure 9 (Appendix E.2) this naive method leads to performance degradation in all tasks, showing that supervision in the refinement process, even if unreliable, is necessary. It could be that training on model outputs without any curation leads to model collapse, a phenomenon observed when training generative models with synthetic data (Shumailov et al., 2023; Ren et al., 2024; Gerstgrasser et al., 2024).

6 Time-constrained Human Study

To validate our findings from LM-simulated settings, we conduct a human study where we recruit workers through CloudResearch Connect[1] to provide both task demonstrations and comparison feedback for an instruction-following task in limited time. Although the tasks they are required to

[1] https://connect.cloudresearch.com



(a) (b) (c)

Figure 6: (a) Distribution of word count of human written demonstrations compared to GPT4-generated answers collected by (Peng et al., 2023). (b) In both rounds of ILR, participants are at least moderately confident in choosing newly proposed labels over original human demonstrations 20% of the time, but they tend to update less in later rounds. Participants are overall more confident when choosing human demonstrations, indicating that the improvements brought by W2SG may not always be as apparent. (c) ILR enables more improvements on top of the initial SFT model, while DPO plateaus similar to what is observed in LM-simulated settings.

Question

Below is an instruction that describes a task... Instruction: Explain the concept of random forest.

Human Demonstration

A random forest is a forest, usually in nature, that is not accounted for with maps or other documentation.

Chosen AI output

A random forest is a machine learning method for predication or classification using multiple decision trees...

Figure 7: Example of lable refinement in ILR: When instructed to explain "random forest," an inaccurate human demonstration misinterpreted it as a natural forest. Human workers then chose to replace this with a more accurate model-generated explanation, improving the SFT dataset quality.

solve are not beyond human capability, imposing time constraints allows us to create annotator noise and obtain unreliable data. This simulates future human supervision of AI on challenging problems that humans struggle with.

Task and data collection. For simplicity, we focus on the Alpaca instruction-following dataset (Taori et al., 2023; Peng et al., 2023) without unsafe questions. Evaluation is done using the complete test set of AlpacaEval (Li et al., 2023) and GPT-4o as a judge, similar to evaluation in our SaferPaca setting. We collect human demonstrations for 1,000 randomly sampled instructions, with workers instructed to spend 1-2 minutes to write each response. Each worker was assigned 15 questions, plus 3 attention checks used to filter out low-quality responses. As shown in Figure 6a, human demonstrations were generally shorter than GPT-4-generated responses, suggesting suboptimal quality due to time constraints. For the comparison feedback used in DPO and ILR, workers spent approximately 30 seconds per question, with 50 questions assigned per worker and 3 additional questions for quality checks. We collected 1,000 comparisons for each algorithm in each round and conducted two rounds of both DPO ($\beta = 0.1$) and ILR ($\alpha = 0.1$). We also used the comparison data collected for DPO with $\beta = 0.1$ to run DPO with several other β values. More details of our human study setup are provided in Appendix D.

ILR outperforms DPO with time-constrained human supervision. With time-constrained unreliable human supervision, ILR demonstrates fast and consistent improvement compared to DPO (Figure 6c). While DPO with strong regularization (large β) improves slowly, DPO with less regularization (small β) improves quickly in the beginning but then plateaus in performance. We hypothesize that DPO may be overoptimizing with insufficient regularization, similarly to our LM-simulated experiments. Notably, the results of ILR and DPO with the collected time-constrained human data are most similar to LM-simulated scenarios with unreliable demonstrations but *reliable* comparison feedback, as seen in Figure 3. This suggests that comparing Alpaca outputs remains relatively straightforward for humans even under time pressure. Future work could explore more complex tasks or even shorter time constraints to better simulate more unreliable human feedback.

ILR effectively improves SFT data because models finetuned on unreliable demonstrations can surpass human supervisors. We find humans are at least moderately confident in choosing model-generated outputs over original human demonstrations for more than 20% of the questions in both rounds of ILR (Figure 6b). This further confirms the W2SG phenomenon, where models trained on unreliable supervision can surpass their supervisor. For example, Figure 7 demonstrates that a participant chose to replace human-written misinterpretation of "random forest" with a more correct and detailed model-generated explanation.

However, improvements brought by W2SG may not always be significant, since participants are often more confident when choosing human demonstrations while less confident when choosing model outputs, as shown in Figure 6b. Also, we observed a decrease in human selection of model outputs in the second round of ILR. This suggests either a limitation in human evaluation capability or W2SG being impossible for certain questions. To address this, future work could explore better refinement mechanisms that combine human demonstrations with AI outputs, potentially allowing for more nuanced refinement in cases where both human and AI responses are partially correct in complementary ways.

7 Discussion

In this work, we study the effectiveness of a typical SFT+DPO pipeline for language model post-training under unreliable supervision, as simulated by small LMs and time-limited humans. We show that, unlike with reliable feedback, SFT+DPO fails to improve upon SFT. Our analysis suggests that DPO struggles to avoid overoptimization with unreliable comparison feedback and requires heavy regularization, preventing it from correcting significant errors in the SFT model.To address this, we propose ILR to redirect comparison feedback towards improving unreliable demonstrations in the SFT dataset. ILR enables larger model updates without the overoptimization risks in preference optimization and consistently outperforms DPO under unreliable supervision.

Limitations. Our focus on RLHF via DPO may not fully capture the nuances of RLHF pipelines that use reward modeling and PPO. Future research should verify whether our findings generalize to these more complex RLHF setups. Also, our human study using time-constrained data collection may not perfectly simulate human errors that arise from more realistic capability constraints. Exploring more challenging tasks such as competition math or coding could potentially provide a closer analogy to supervising AI on tasks that humans truly struggle with.

Future work. Our findings open several exciting directions for future research. For example, one could explore hybrid approaches that combine ILR with RLHF. Applying PPO or DPO after several rounds of ILR could potentially yield more improvements, since the problem of imitating errors in unreliable demonstrations is relieved after improving the SFT data with ILR. Additionally, investigating more sophisticated label refinement strategies, such as synthesizing human demonstrations with model-generated proposals or implementing ensemble techniques through more than two cross-labeling splits, could potentially further enhance the effectiveness of ILR.

Broader impact. Our findings suggests a potential paradigm shift in how we approach human oversight of AI systems under unreliable supervision. We show that the canonical RLHF post-training pipeline may no longer be the best use of human comparison feedback. Instead, we need methods like ILR that leverage model outputs to improve and learn from unreliable human supervision.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016. 1, 2

Owura Asare, Meiyappan Nagappan, and N Asokan. Is github's copilot as bad as humans at introducing vulnerabilities in code? *Empirical Software Engineering*, 28(6):129, 2023. 1

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024. 2, 3

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 1, 3

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*, 2023. 4, 14

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998. 3

Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiūtė, Amanda Askell, Andy Jones, Anna Chen, et al. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022. 2

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023. 1, 2, 3, 4, 5, 15

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust dpo: Aligning language models with noisy feedback. *arXiv preprint arXiv:2403.00409*, 2024. 3

Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts. *arXiv preprint arXiv:1810.08575*, 2018. 3

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 4, 14

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1, 3, 4, 5

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 4

Adam Fisch, Jacob Eisenstein, Vicky Zayats, Alekh Agarwal, Ahmad Beirami, Chirag Nagpal, Pete Shaw, and Jonathan Berant. Robust preference optimization through reward model distillation. *arXiv preprint arXiv:2405.19316*, 2024. 3

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023. 2, 6

Yang Gao, Dana Alon, and Donald Metzler. Impact of preference noise on the alignment performance of generative language models. *arXiv preprint arXiv:2404.09824*, 2024. 3

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, et al. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024. 8, 19

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018. 3

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 16

Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. *arXiv preprint arXiv:1805.00899*, 2018. 3

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 4

Zachary Kenton, Noah Y Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah D Goodman, et al. On scalable oversight with weak llms judging strong llms. *arXiv preprint arXiv:2407.04622*, 2024. 3

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive llms leads to more truthful answers. *arXiv preprint arXiv:2402.06782*, 2024. 3

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 16

Jan Hendrik Kirchner, Yining Chen, Harri Edwards, Jan Leike, Nat McAleese, and Yuri Burda. Prover-verifier games improve legibility of llm outputs. *arXiv preprint arXiv:2407.13692*, 2024. 3

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. 14

Jan Leike and Ilya Sutskever. Introducing superalignment, Jul 2023. URL https://openai.com/blog/introducing-superalignment. 1

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018. 8

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36, 2024. 4, 14

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023. 4, 9, 14

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023. 3

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. Llm critics help catch llm bugs. *arXiv preprint arXiv:2407.00215*, 2024. 1, 3

Julian Michael, Salsabila Mahdi, David Rein, Jackson Petty, Julien Dirani, Vishakh Padmakumar, and Samuel R Bowman. Debate helps supervise unreliable experts. *arXiv preprint arXiv:2311.08702*, 2023. 3

OpenAI. Hello gpt-4o, 2024. URL https://openai.com/index/hello-gpt-4o/. Accessed: 2024-09-22. 4, 14

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 1, 3

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023. 9, 14

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3, 5

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014. 3

Yi Ren, Shangmin Guo, Linlu Qiu, Bailin Wang, and Danica J Sutherland. Language model evolution: An iterated learning perspective. *arXiv preprint arXiv:2404.04286*, 2024. 8, 19

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017. 3

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022. 3

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1, 3

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pp. 4596–4604. PMLR, 2018. 16

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023. 8, 19

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 34(11):8135–8153, 2022. 3

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023. 2, 4, 9, 14, 17

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024. 4

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 3

Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Samuel R. Boman, He He, and Shi Feng. Language models learn to mislead humans via rlhf, 2024. 1

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020. 17

A Tasks and Evaluation

A.1 Datasets

GSM8K (Cobbe et al., 2021): This dataset contains math word problems widely used to evaluate language models' mathematical problem solving capabilities. In total, there are 8.5K question-answer pairs for training and 3K for testing.

BIRD (Li et al., 2024): In this task, models are required to generate an SQL query that answer a question of interest, given context of database descriptions. We filter out questions with token lengths greater than 1024 and obtained 8.5K examples for training and 1.4K for testing.

SaferPaca (Bianchi et al., 2023): This dataset is a mixture of instruction following questions from (Taori et al., 2023) and demonstrations of refusals to unsafe instructions. It requires a model to serve as an assistant that follows human users' instructions while generating responsible responses and rejecting harmful prompts. For efficiency, we use a subset that contains 9500 general instructions with responses labeled by GPT-4 (collected by Peng et al. (2023)) and 500 safety-related instructions with the original labels in (Bianchi et al., 2023). The test set we use contains 300 questions and reference answers taken from AlpacaEval (Li et al., 2023), alongside 278 safety questions and simple rejections ("Sorry, I cannot help with that.") of the `I-MaliciousInstructions` and `I-CoNA` datasets used in (Bianchi et al., 2023).

A.2 Evaluation

Sampling. For all tasks and models, we sample one response for each prompt using vLLM (Kwon et al., 2023) with beam search enabled and beam size set to 4.

Metrics. For GSM8K, we simply parse the final numerical answer and compute exact match accuracy. For BIRD, we download all databases and execute the generated code on them to compute execution accuracy, following (Li et al., 2024). We ensure all models' generated code is executed on the same machine for fair comparison. For SaferPaca, we adopt the CoT prompt (shown below) from (Li et al., 2023) and use GPT-4o (OpenAI, 2024) to compute model answers' win rates against the reference answers. Specifically, we compute GPT-4o's mean probability of generating a token that chooses the model's answer over the reference answer.

Prompt template for GPT-4o evaluation

```
Select the output A or B that best matches the given instruction.
Choose your preferred output, which can be subjective. Your answer
should first contain a concise explanation and then it should end
with ONLY the token: ‘A‘ or ‘B‘ (no dot, no backticks, no new line,
no quotes, no ’it depends’, no ’both’ ...), because we will use
‘output[-1]‘ to extract the best output.

Here’s an example:

# Example:
## Instruction:
Give a description of the following job: "ophthalmologist"

## Output A:
An ophthalmologist is a medical doctor who specializes in the
diagnosis and treatment of eye diseases and conditions.

## Output B:
An ophthalmologist is a medical doctor who pokes and prods at your
eyes while asking you to read letters from a chart.

## Concise explanation followed by ‘A‘ or ‘B‘

### Concise explanation
A provides a comprehensive and accurate description of the job of an
```

```
ophthalmologist. In contrast, B is more of a joke.

### Which is best, `A` or `B`?
A

# Task:
Now is the real task.

## Instruction:
{instruction}

## Output `A`:
{A}

## Output `B`:
{B}

## Concise explanation followed by `A` or `B`

### Concise explanation
```

B LM-BASED SIMULATION DETAILS

In each task, we split the training set into two halves. We use the first half as ground truth to train $\tilde{p}$ and $\tilde{q}$, and then use $\tilde{p}$ to generate labels for the second half or use $\tilde{q}$ to generate comparison feedback for answers to questions in the second half. This is similar to the setup in (Burns et al., 2023), despite that we are operating on text generation tasks with both task demonstrations and comparison feedback.

To collect training data for $\tilde{q}$, we evenly save 10 intermediate checkpoints when training $\tilde{p}$ and use them to sample answers for questions in their training set. Then, we pair these low-quality answers with ground truth answers, and train $\tilde{q}$ to select the ground truth with a standard binary classification loss. In SaferPaca, we balance refusals and acceptances (*i.e.*, following the instruction) in the training data to avoid class bias in $\tilde{q}$. This is done by first training two $\tilde{p}$ on safe and unsafe instructions in SaferPaca respectively and sampling answers from them for each instruction. In this way, we obtain both acceptance and refusal responses for each instruction, and then pair them up with the ground truth as $\tilde{q}$'s balanced training data.

We use the following template to compose two answers and the question into a prompt for $\tilde{q}$:

Prompt template for $\tilde{q}$

```
QUESTION:
{{question}}

ANSWER (A):
{{answer_a}}

ANSWER (B):
{{answer_b}}

You gave the answer (A) to the question. Do you accept to
change it to answer (B) proposed by another model?
```

Note that the prompt states an order of "your answer" and "answer proposed by another model" due to historical reasons. In practice, we randomize the two answers' order to avoide position bias.

Table 1: Training Hyperparameter for Each Dataset

Dataset	Epoch	Batch Size	Max Answer Token
GSM8K	2	32	256
BIRD	2	32	256
SaferPaca	4	32	512

Algorithm 1: Iterative Label Refinement (ILR)

```
Input  : Initial SFT dataset $\mathcal{D}_0 = (\mathcal{X}, \tilde{\mathcal{Y}}_0)$
         Comparison feedback annotator $\tilde{q}$
         Maximum iterations $K$
         Refinement rate $\alpha$
Output : Refined dataset $\mathcal{D}_K$
         Final model $\widehat{p}^{\,K}_{\text{SFT+ILR}}$
for $k = 0$ to $K - 1$ do
    Split $\mathcal{D}_k$ into two disjoint subsets:
    $\mathcal{D}_k^1 = (\mathcal{X}^1, \tilde{\mathcal{Y}}_k^1)$ and $\mathcal{D}_k^2 = (\mathcal{X}^2, \tilde{\mathcal{Y}}_k^2)$, where $\mathcal{X}^1 \cup \mathcal{X}^2 = \mathcal{X}$
    Train models $\widehat{p}^{\,1}_{\text{SFT}}$ on $\mathcal{D}_k^1$ and $\widehat{p}^{\,2}_{\text{SFT}}$ on $\mathcal{D}_k^2$
    foreach $x_i \in \mathcal{X}^1$ do
        Sample one proposal $z_i \sim \widehat{p}^{\,2}_{\text{SFT}}(\cdot \mid x_i)$
    foreach $x_i \in \mathcal{X}^2$ do
        Sample one proposal $z_i \sim \widehat{p}^{\,1}_{\text{SFT}}(\cdot \mid x_i)$
    foreach $x_i \in \mathcal{X}$ do
        Collect comparison feedback for $(z_i, \tilde{y}_{k,i})$ using $\tilde{q}$
        Proposal $z_i$ is accepted if the feedback is $z_i \succ \tilde{y}_{k,i}$
    Let $A$ be the set of indices where $z_i$ is accepted
    Limit $|A| \leq \alpha|\mathcal{X}|$ by selecting proposals with highest confidence
    Update labels:
    $\tilde{y}_{k+1,i} = \begin{cases} z_i & \text{if } i \in A, \\ \tilde{y}_{k,i} & \text{otherwise} \end{cases}$
    Form the updated dataset $\mathcal{D}_{k+1} = (\mathcal{X}, \tilde{\mathcal{Y}}_{k+1})$
    Train model $\widehat{p}^{\,k+1}_{\text{SFT+ILR}}$ via SFT on $\mathcal{D}_{k+1}$
return $\mathcal{D}_K, \hat{p}_{\text{SFT+ILR}}[K]$
```

C Implementation Details of SFT, DPO, and ILR

In all our experiments, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2021) for efficient model training. We set $r = 64$ and $\alpha = 128$ for all models.

SFT. For each task, we use a consistent setting of epoch, batch size, and max answer token for all models (Table 1). We use set learning rate to $5e^{-4}$ for Gemma 2B and $1e^{-4}$ for Mistral 7B and Meta Llama 70B across all tasks. We use Adam (Kingma, 2014) optimizer for Gemma 2B and Mistral 7B and AdaFactor (Shazeer & Stern, 2018) for Meta Llama 70B. We enable gradient checkpointing and use gradient accumulation with a mini batch size of 1 for all models.

ILR. We present the pseudo-code of ILR in Algorithm 1. Each round of ILR uses the same training configuration as the initial SFT, since the only difference is that the dataset is refined. When training the half-data models used for generating new proposals, we also adopt the same SFT training configuration for simplicity.

In our LM-simulated experiments, we only collect comparison feedback for proposals and initial demonstrations that are sufficiently different. In GSM8K and BIRd, this is determined by having different final numerical answer or execution result. For SaferPaca, we use OpenAI Embeddings API to compute the embeddings for the two responses, and then take dot product to compute their embedding distance. We only collect comparison feedback for pairs with the top 50% largest embedding distance.

DPO. We use the `DPOTrainer` in HuggingFace Transformers (Wolf et al., 2020) to perform DPO training. In all tasks, we use the suggested `rmsprop` optimizer with learning rate set to $1e^{-6}$, and we train for the same number of epochs as SFT. For all LM-simulated tasks, we sample 6 completion for each prompt, create 3 pairs with them, and then gather comparison feedback using $\tilde{q}$. We subsample the top 15% confident feedback (measured by $|\tilde{q}(y_1 \succ y_2 \mid x) - 0.5|$) when constructing $\widetilde{\mathcal{D}}_{\text{RLHF}}$, since we find this performs much better than using all feedback generated by the unreliable $\tilde{q}$ in DPO. Specifically, in GSM8K (2B → 7B), model accuracy after one round of DPO is 0.32±0.0043 when using all feedback and 0.36±0.0032 when using the 15% most confident feedback.

D Human Study Details

We use CloudResearch Connect[2] to recruit online workers to write human demonstrations for 1000 Alpaca (Taori et al., 2023) instructions and provide comparison feedback that are used in two rounds of ILR and DPO.

D.1 Collecting human demonstrations

In our survey for collecting human demonstrations, we assign each worker 15 questions with 3 additional screening questions formatted in the same way for filtering out low-quality responses. Specifically, the screening questions and filtering critiria are:

1. **Instruction:** How many letter R's does the word 'STRAWBERRY' have?
   Expected answer: Anything containing 3 or three.
2. **Instruction:** Weng earns $12 an hour for babysitting. Yesterday, she just did 50 minutes of babysitting. How much did she earn?
   Expected answer: Anything containing 10 or ten.
3. **Instruction:** I want to know what dish I can cook with these ingredients: eggs, tomatoes, salt, oil. Also, please give me a list of steps to cook it.
   Expected answer: Anything containing the mentioned ingredients and at least 2 steps for cooking them.

The task instruction we provided at the beginning of the survey is:

Task instruction for writing response demonstrations

Thank you for participating in our study! We would like your help in training an AI assistant that can follow users' instructions in a helpful, honest, and harmless way.

In this survey, you will be presented with 15 questions posed by human users to an AI chatbot. For each question, please take more than 1 minute and less than 2 minutes to write a response that you think is as helpful, honest, and harmless as possible. Please aim for high-quality detailed responses within the given time.

Additionally, there will be 3 randomly placed quality check questions. These questions are super easy and you will definitely get them right if you are attentive. We will manually check your answers to these questions, and if you don't pass, your response may not be qualified. However, you will still receive a base payment of $0.75. In such a case, please manually return the assignment to ensure we can pay you the base amount. Please let us know if you find our judgement incorrect.

To get full compensation, please complete all questions. Otherwise, you will only receive a base payment of $0.75.

At the end, we welcome your feedback on improving the survey for future participants!

[2] https://connect.cloudresearch.com

NOTE: Please refrain from searching for answers during the survey. No use of external sources (e.g., Google or ChatGPT) is allowed.

D.2 Collecting human comparison feedback

In each round of ILR and DPO, we collect human comparison feedback for all 1000 questions. For DPO, model completions are generated by models trained with $\beta = 0.1$. In our survey for collecting human comparison feedback, we assign each worker 50 questions with 3 additional screening questions formatted in the same way for filtering out low-quality responses. Specifically, the screening questions and filtering critiria are:

1. **Instruction:** Where is water that has its salt removed before it can be used as drinking water most likely to have come from?
   Response A: Water that has its salt removed before it can be used as drinking water is most likely to have come from a sea.
   Response B: Water that has its salt removed before it can be used as drinking water is most likely to have come from a lake.
   Expected answer: A with any confidence level.
2. **Instruction:** Help me solve this math problem. Input: How can I compute the area of a circle with radius 5?
   Response A: The area of it is 25π.
   Response B: Note that the area of a circle with radius r is π * r^2. Therefore, the area of a circle with radius 5 is π * 5 * 5 = 25π.
   Expected answer: B with any confidence level.
3. **Instruction:** Immediately before and after running a 50 metre race, your pulse and breathing rates are taken. What changes would you expect to find?
   Response A: After running a 50-metre race, an increase in pulse and breathing rate is the changes one would expect to find immediately before and after.
   Response B: After running a 50 metre race, you would expect to find an increase in pulse but no change in breathing rate when your pulse and breathing rates are taken immediately before and after the race.
   Expected answer: A with any confidence level.

The task instruction we provided at the beginning of the survey is:

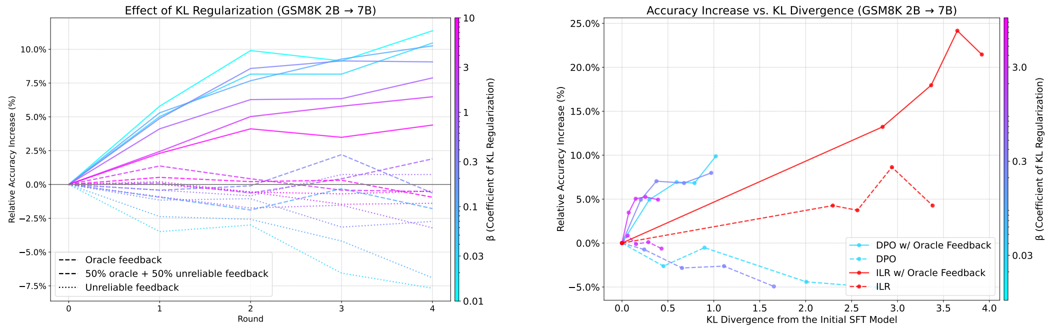
Task instruction for providing comparison feedback

Thank you so much for participating in our study! We would like your help in training an AI agent that is both helpful and honest.

First, you will be shown three screening questions, and if you get them correct, you will be able to participate in the rest of our survey. If you do not qualify, you will be eligible to receive a base payment of $0.75, and you will be automatically redirected to a Google Form. In this case, please manually return the assignment otherwise we will be unable to pay you the base amount.

For the main task, you will be shown fifty request-response groups: each request is posed by a user to an AI chatbot, and you will see two potential responses that the AI chatbot has generated. Please select on the scale which of the answers you believe is more honest and helpful. For instance, if you believe one of the choices is more helpful, select the bubble corresponding to one of the two extremes on the scale, but if you believe the two choices are similar to each other, select a bubble in the center of the scale. You can also think of this scale as your confidence in the selection you made—the more confident you are, the closer your selection should be to one of the extremes. Optionally, you can explain why you chose the response that you did in the scratch space provided on each page.

You should spend around 30 seconds and no more than 1 minute for each question. The total time limit of this survey is 50 minutes. Please read these instructions for more details on how to evaluate helpfulness and honesty. You may open the instructions in a separte tab for reference whenever you want. We know this is some interesting trivia, but please resist the urge to search for the answers until



(a) DPO effectively improves SFT models with oracle feedback, especially with weak regularization. However, with unreliable feedback, improvements are limited to a narrow range of regularization strengths because weak regularization causes overoptimization.

(b) Strong regularization in DPO limits useful model updates, while small regularization leads to overoptimization of unreliable feedback. In contrast, ILR facilitates large model updates, allowing for faster improvement and efficient use of comparison feedback.

Figure 8: DPO struggles to avoid overoptimization of noisy preferences while also updating the initial suboptimal SFT model sufficiently to improve performance.

you submit the survey! No use of external sources, including Google or ChatGPT, is permitted.

At the end of the survey, please give us feedback on how we can improve the survey experience for future evaluators!

NOTE: Please do not take this survey more than once! You will not be compensated for more than one attempt.

Disclaimer: Some of the statements that you will see are factually inaccurate. Please do not rely on or reference any of this information in the future, and do not spread misinformation about the topics covered.

D.3 DPO and ILR Hypeperameters

In both DPO and ILR, we only use comparison data with at least moderate confidence. We tested the first round of DPO using the top 10%, 20%, 30%, 40% and 80% most confident comparisons and found 30% yields the best performance, hence this is applied throughout the two rounds. We also tested $\beta \in \{0.01, 0.1, 0.5, 0.1\}$ for the first round and found $\beta = 0.1$ performs the best. For ILR, we tested $\alpha \in \{0.1, 0.2, 0.3\}$ for the first round and applied the best-performing $\alpha = 0.1$ across the two rounds.

E Additional Experiments

E.1 KL Regularization in DPO

In Figure 8, we present the complete experiment results when using different levels of KL regularization and feedback quality for DPO.

E.2 Failure of Naive ILR: Supervison for Refinement is Necessary

To understand the importance of comparison feedback within ILR's refinement process, we compare it to a naive approach that directly replaces an α fraction of the original labels with new proposals without any feedback. As shown in Figure 9, this naive method leads to performance degradation in all tasks, showing that supervision in the refinement process, even if unreliable, is necessary. It could be that training on model outputs without any curation leads to model collapse, a phenomenon observed when training generative models with synthetic data (Shumailov et al., 2023; Ren et al., 2024; Gerstgrasser et al., 2024).

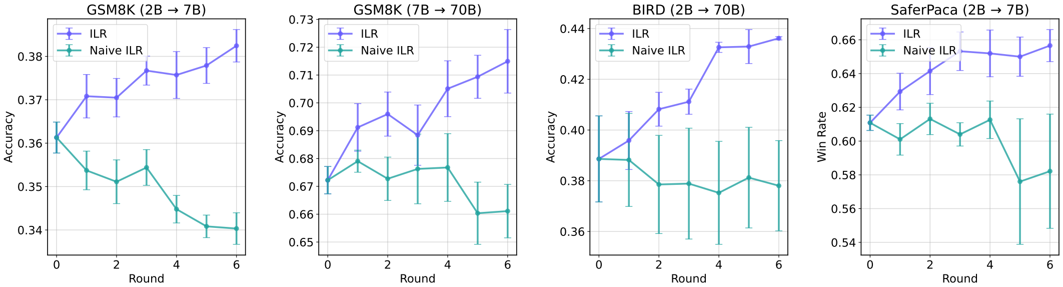


Figure 9: Naively replacing initial SFT labels with new model's proposals lead to performance degradation across all tasks, suggesting the importance of refinement oversight.

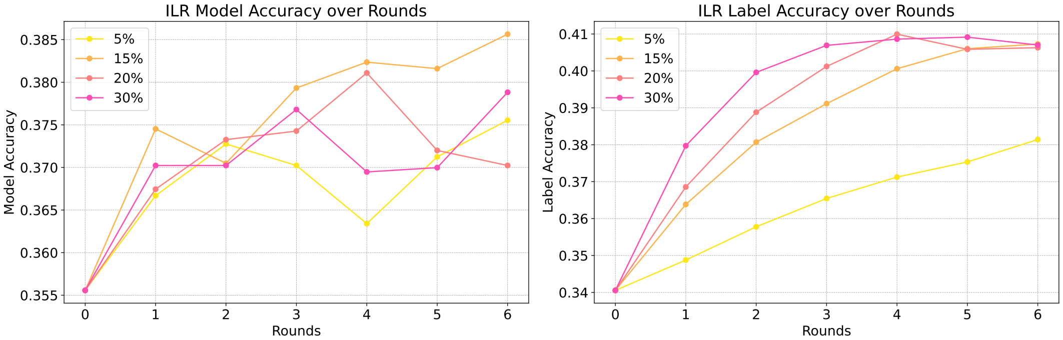


Figure 10: **Left:** Overall, ILR is not particularly sensitive to choice of α, although larger α may lead to less stable performance, while small α does not allow enough improvement. **Right:** When α is large enough, SFT data label accuracy converge to a similar level. When α is overly small, label accuracy is not efficiently improved.

E.3 Effect of Controlling Refinement Speed in ILR

The refinement speed in ILR, controlled by α, plays a crucial role in maintining stability. Figure 10 illustrates the impact of different α values on model performance in GSM8K. We find $\alpha = 0.05$ too small for effective improvement of label accuracy, while large α's make the refinement process less stable. According to this result, we set $\alpha = 0.15$ for all other experiments in LM-based simulation and find that this choice leads to relatively stable performance across all tasks, showing that α is not particularly sensitive to change in data distribution. Future work can explore adaptive refinement speed to remove the potential need for manual hyperparameter tuning.