LLM Iterative Fine-Tuning as Filtered Expectation Maximization

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) can solve complex problems better by thinking about them first and rationalizing answers. The rationalization can be viewed as introducing latent variables, and we propose an EM-like algorithm for learning to rationalize correct answers. The main challenge lies in designing a sampling distribution of rationales that justify correct answers. We instantiate and compare three sampling schemes: rejection sampling with a budget, self-taught reasoner (STaR), and prompt posterior sampling (PPS), which only keeps the rationalization stage of STaR that conditions on the correct answer in the prompt. We experiment with LLM-as-a-judge calibration and summarization from feedback tasks, where conditioning on the correct answer provides a strong guidance for generating rationales. Our experiments show the efficacy of PPS over other sampling schemes, and that the sampling scheme can have a significant impact on performance.

1 Introduction

Large language models (LLMs) have shown strong performance on complex tasks, such as multiple-choice exams and math word problems (Zelikman et al., 2022; Shao et al., 2024). In these tasks, it is often difficult to answer questions directly. Instead, it is more beneficial to prompt the model to first think (rationalize) and then output the answer. This is the key idea in *chain-of-thought (CoT)* prompting, where the model is asked to *think step by step* before answering (Wei et al., 2022; Kojima et al., 2022).

Latent variable model (LVM) viewpoint. We consider the task of learning a mapping from a question x to its answer $y^\star$, and specifically focus on deterministically verifiable answers. This setting is common in domains like LLM-as-a-judge calibration (Zheng et al., 2023; Sahoo et al., 2025) and summarization from feedback tasks (Stiennon et al., 2020). When the task is complex, it is hard to model $x \to y^\star$ directly. This motivates rationalization – introducing a *latent variable* z representing the (unobserved) intermediate rationalizing process. Then, conditioned on both the question x and rationale z, the model has a much higher chance of generating the correct answer $y^\star$ than when conditioned on x alone. With the rationale, the predictive structure factors as a *latent variable model (LVM)* (Koller & Friedman, 2009)

$$\begin{tikzcd} x \arrow[r] \arrow[rr, bend left] & z \arrow[r] & y^\star \end{tikzcd} \tag{1}$$

This latent-variable perspective on rationales has been explored in various forms in prior works (Zelikman et al., 2022; Yuan et al., 2023; Phan et al., 2023; Hu et al., 2024; Zhong et al., 2025; Tang et al., 2025; Xu, 2025). We adopt it as a starting point for revisiting existing algorithms and in particular the role of sampling.

Expectation maximization (EM). Once rationales are modeled as latent variables, the *EM algorithm* (Dempster et al., 1977; Moon, 1996; Neal & Hinton, 1998) can be used to learn them. The algorithm alternates between computing a posterior distribution over rationales conditioned on the observed question-answer pairs (**E-step**) and maximizing the expected complete data log-likelihood (**M-step**). Neither of the steps can be implemented exactly in LLMs. The **E-step** requires computing an expectation over z conditioned on both x and $y^\star$ in (1). The expectation does not have a closed form, and approximating it via Monte Carlo sampling requires sampling from the exact posterior. In theory, classic statistical techniques,

such as rejection sampling (Neal & Hinton, 1998) and Markov chain Monte Carlo (Wei & Tanner, 1990; Doucet et al., 2001; Phan et al., 2023), could be used, but they are not guaranteed to be computationally or statistically efficient. The **M-step** also lacks a closed-form solution and is typically approximated by gradient-based optimization (Bottou et al., 2018). Many prior works (Balakrishnan et al., 2017; Neath, 2013; Cappé & Moulines, 2009) studied this and similar approaches.

Iterative fine-tuning LLMs. Concurrently, the LLM community has introduced algorithms that train LLMs by explicitly generating and filtering their own generations, a paradigm known as *iterative fine-tuning (IFT)*. Examples include rejection-based fine-tuning (Yuan et al., 2023; Dong et al., 2023; Singh et al., 2024; Shao et al., 2024) and self-taught reasoning (Zelikman et al., 2022). These methods share a similar high-level structure:

1. Sample one or more rationale-answer pairs $z, \hat{y}$ conditioned on the question x;
2. Verify if $\hat{y} = y^\star$;
3. Fine-tune on accepted rationale-answer pairs.

From the LVM perspective, these methods perform approximate **E-steps** followed by **M-steps**. While Zelikman et al. (2022) noted this link, they focused on a policy-gradient justification for their proposed self-taught reasoner, leaving the formal EM connection under-explored.

Contributions. In this work, we examine this connection in detail. Our contributions are as follows:

- **Filtered EM.** We derive a reward-based *filtered EM* (`FEM`) objective for tackling the LVM with a deterministic verifier. We approximate the **E-step** via a single Monte Carlo sample and the **M-step** using a *filtered* gradient update. At a high level, we sample rationale-answer pairs from the latest model for all questions and fine-tune on those with correct sampled answers. This formulation rigorously relates EM to modern reward-based optimization and frames LLM self-improvement as an iterative supervised fine-tuning procedure where latent rationales are resampled at each epoch. We also highlight its connection to reward-weighted fine-tuning of Mukherjee et al. (2025).
- **Unifying and instantiating sampling schemes.** We demonstrate that existing IFT algorithms, such as rejection sampling (`RS`) and self-taught reasoning (`STaR`), can be viewed as specific instances of our `FEM` framework with different rationale sampling schemes. Furthermore, we propose prompt posterior sampling (`PPS`), which is the second rationalization stage of `STaR` and has not been systematically studied yet.
- **Empirical evaluation of sampling schemes.** We empirically evaluate all sampling schemes on LLM-as-a-judge calibration (Zheng et al., 2023; Sahoo et al., 2025) and summarization from feedback (Stiennon et al., 2020) tasks. The evaluation is conducted on two Llama (Llama Team, 2024) and three Qwen (Qwen Team, 2024) models. We observe that `PPS` consistently outperforms other sampling schemes. Our results show that the *sampling scheme* approximating the **E-step** is critical to the effectiveness of IFT.

2 Filtered Expectation-Maximization (FEM)

Let $\mathcal{D} = \{(x_i, y_i^\star)\}_{i=1}^N$ be a dataset of N question-answer pairs. We denote the probability that an LLM $\pi(\cdot \mid \cdot; \theta)$ with parameters θ generates an answer y when prompted with x by $\pi(y \mid x; \theta)$. For LLMs that generate rationales, we write $\pi(z, y \mid x; \theta)$ for the joint probability distribution over a rationale z and final answer y. We denote by ∇ the gradient operator with respect to θ.

As discussed in Section 1, learning $x_i \to z_i, y_i^\star$ is easier and more likely to lead to higher accuracy than $x_i \to y_i^\star$ alone, where z_i is unobserved underlying rationalization. A standard way of learning LVMs is the *expectation-maximization algorithm* (EM, Dempster et al. (1977)). EM is an iterative algorithm with two

Algorithm 1 Filtered EM (FEM).

1: **Input:** Number of iterations K, dataset $\mathcal{D} = \{(x_i, y_i^\star)\}_{i=1}^N$, initial model $\theta^{(0)}$, sampling distribution $q(\cdot \mid x, y^\star; \theta)$
2: **for** $k = 1, \ldots, K$ **do**
3: **for all** $i \in [N]$ **do** $\hat{z}_i^{(k)}, \hat{y}_i^{(k)} \sim q(\cdot \mid x_i, y_i^\star; \theta^{(k-1)})$ ▷ Rationale sampling
4: $\theta^{(k)} \leftarrow \theta^{(k-1)}$
5: **for all** $i \in [N]$ **do** $\theta^{(k)} \leftarrow \theta^{(k)} + \eta^{(k)} r(\hat{y}_i^{(k)}, y_i^\star) \nabla \log \pi(\hat{z}_i^{(k)}, \hat{y}_i^{(k)} \mid x_i; \theta^{(k)})$ ▷ Gradient update
6: **Output:** Learned model $\theta^{(K)}$

steps in each iteration. The **E-step** computes the expected complete data log-likelihood given the most recent model parameter $\theta^{(k-1)}$,

$$\mathcal{J}(\theta) := \textstyle\sum_{i=1}^N \mathbb{E}_{z_i \sim \pi(\cdot \mid x_i, y_i^\star; \theta^{(k-1)})} [\log \pi(z_i, y_i^\star \mid x_i; \theta)] \,,$$

where $\pi(\cdot \mid x_i, y_i^\star; \theta^{(k-1)})$ is the ***posterior distribution*** of the latent rationale z_i under the previous iterate $\theta^{(k-1)}$. The **M-step** solves for $\theta^{(k)} = \arg\max_\theta \mathcal{J}(\theta)$.

2.1 From EM to Filtered EM

Approximating M-step. We first focus on the **M-step** and introduce a series of approximations that leads to our algorithm. First, we approximate the exact maximization by a single step of gradient ascent

$$\theta^{(k)} \leftarrow \theta^{(k)} + \eta^{(k)} \underbrace{\nabla \mathbb{E}_{\hat{z}_i} [\log \pi(\hat{z}_i, y_i^\star \mid x_i; \theta)]}_{(*)} \tag{2}$$

for each data point $i \in [N]$, where $\eta^{(k)}$ is the learning rate at iteration k and the expectation $\mathbb{E}_{\hat{z}_i}$ is taken with respect to $\hat{z}_i \sim \pi(\cdot \mid x_i, y_i^\star; \theta^{(k-1)})$.

Now note that the posterior distribution is by definition *consistent* with the evidence that we condition on. Therefore, any $\hat{z}_i, \hat{y}_i$ drawn from $\pi(\cdot \mid x_i, y_i^\star)$ must satisfy $\hat{y}_i = y_i^\star$ almost surely. This implies that checking whether the generated answer matches the ground truth is theoretically equivalent to enforcing the posterior distribution's support. When sampling from a broader proposal distribution, this ensures that the update operates only on answer-consistent samples. We formalize this using a binary reward $r(\hat{y}_i, y_i^\star) := \mathbb{1}[\hat{y}_i = y_i^\star]$, which lets us rewrite (2) with

$$(*) = \nabla \mathbb{E}_{\hat{z}_i, \hat{y}_i} [r(\hat{y}_i, y_i^\star) \log \pi(\hat{z}_i, \hat{y}_i \mid x_i; \theta)] \,, \tag{3}$$

where the expectation $\mathbb{E}_{\hat{z}_i, \hat{y}_i}$ is taken with respect to $\hat{z}_i, \hat{y}_i \sim \pi(\cdot \mid x_i, y_i^\star; \theta^{(k-1)})$.

We make two additional observations. First, the expectation is defined with respect to a distribution that depends only on the previous parameters $\theta^{(k-1)}$, but not the optimized θ. Second, the reward does not depend on θ. As a result, the gradient can be moved inside the expectation and we get

$$(*) = \mathbb{E}_{\hat{z}_i, \hat{y}_i} [r(\hat{y}_i, y_i^\star) \nabla \log \pi(\hat{z}_i, \hat{y}_i \mid x_i; \theta)] \,. \tag{4}$$

In LLMs, this expectation cannot be computed efficiently because $\hat{z}_i, \hat{y}_i$ are sequences of tokens from an exponentially large space. Therefore, we approximate it by a single Monte Carlo sample as

$$(*) \approx r(\hat{y}_i, y_i^\star) \nabla \log \pi(\hat{z}_i, \hat{y}_i \mid x_i; \theta) \,, \tag{5}$$

where $\hat{z}_i, \hat{y}_i \sim \pi(\cdot \mid x_i, y_i^\star; \theta^{(k-1)})$ is a single rationale-answer sample. This is a *filtered* gradient update of the classic Monte-Carlo EM (Wei & Tanner, 1990; Neath, 2013), where the reward $r(\hat{y}_i, y_i^\star)$ serves as a filter in broader proposal distributions (Section 3).

Approximating E-step. Exact sampling of $\hat{z}_i, \hat{y}_i$ from the posterior distribution in (5) is computationally infeasible in LLMs because it requires inverting the generative process in (1) to obtain rationales $\hat{z}_i$ leading to correct answers $y_i^\star$. Moreover, $\hat{z}_i, \hat{y}_i$ are sequences of tokens from an exponentially large space.

We address this challenge by considering a generic *rationale proposal distribution* $q(\hat{z}, \hat{y} \mid x, y^\star; \theta)$ and implement it by *prompting* the most recent model $\theta^{(k-1)}$. Then we approximate (5) by sampling $\hat{z}_i, \hat{y}_i \sim q(\cdot \mid x_i, y_i^\star; \theta^{(k-1)})$. More specifically, for each data point, we sample $\hat{z}_i, \hat{y}_i$ and fine-tune on those with correct sampled answers, $\hat{y}_i = y_i^\star$. Because the reward $r(\hat{y}_i, y_i^\star)$ is binary, the rationales leading to incorrect answers are filtered out in the gradient update.

The pseudo-code of our algorithm is given in Algorithm 1. We call it *filtered EM (*`FEM`*)* due to reward filtering in (5). The algorithm is iterative. In each iteration, it applies our approximate E- and M-steps to all question-answer pairs sequentially: Line 3 is the **E-step** applied to dataset $\mathcal{D}$ and Line 5 is the corresponding **M-step** with reward filtering. For simplicity of exposition, we assume batch size 1 but other batch sizes are possible. The design of `FEM` is motivated by *supervised fine-tuning (SFT)*. In particular, each iteration of `FEM` can be viewed as an epoch in SFT, where the training data $\hat{z}_i^{(k)}, \hat{y}_i^{(k)}$ are sampled from the learned model (Line 3). It can also be interpreted as reward-weighted fine-tuning (Mukherjee et al., 2025), where the model is updated to increase the reward-weighted log-likelihood of the sampled rationale-answer pairs.

The rationale proposal distribution $q(\cdot \mid x, y^\star; \theta)$ is a critical design choice that should satisfy two primary criteria. First, it must maximize the likelihood of the event $\hat{y} = y^\star$ to optimize training data utilization. Second, it should produce high-quality rationales $\hat{z}$ that serve as effective intermediate states, ensuring that conditioning on $x, \hat{z}$ enhances downstream target accuracy and task generalization. We explore various instantiations of q in Section 3.

2.2 Filtered EM and Reinforcement Learning

The filtered EM update can also be justified through its relationship to reward-based maximization in *reinforcement learning (RL)*. To show this, we build on the work of Mukherjee et al. (2025), which extends results of Ma et al. (2019); Liang & Vlassis (2022) from offline logged bandits to offline RL.

Lemma 1. *Let $x, y^\star$ be a question-answer pair and $r(\hat{y}, y^\star) = \mathbb{1}[\hat{y} = y^\star]$. Then for any parameter vector θ and iteration $k \geq 1$,*

$$\underbrace{\mathbb{E}_{\hat{z},\hat{y}\sim\pi(\cdot|x;\theta)}[r(\hat{y}, y^\star)]}_{RHS} \geq \underbrace{\mathbb{E}_{\hat{z},\hat{y}\sim q(\cdot|x,y^\star;\theta^{(k-1)})}[r(\hat{y}, y^\star) \log \pi(\hat{z}, \hat{y} \mid x; \theta)]}_{LHS} + C\,. \tag{6}$$

where $C \geq 0$ is independent of θ. The inequality is tight when $q(\cdot \mid x, y^\star; \theta^{(k-1)}) \equiv \pi(\cdot \mid x; \theta)$.

Proof. The proof mimics Mukherjee et al. (2025) and is provided in Appendix A for completeness. □

Lemma 1 says that reward-weighted fine-tuning (LHS) optimizes a lower bound on the reward maximization objective (RHS). Our `FEM` update in (5) iteratively refines it by alternating between updating the data sampling distribution q (**E-step**) and maximizing the lower bound (**M-step**).

3 Rationale Sampling

Now we instantiate the rationale proposal distribution $q(\cdot \mid x, y^\star; \theta)$ in Algorithm 1. We focus on three representative schemes: rejection sampling with a budget, self-taught reasoning, and prompt posterior sampling.

Rejection sampling with budget M (`RS`). A statistically correct way of sampling z from the LVM in (1) is rejection sampling (Neal & Hinton, 1998). In LLMs, it corresponds to sampling rationale-answer pairs conditioned on the question until the correct answer is sampled, with a budget of at most M. The underlying intuition is to create a virtuous cycle: as the model improves, it is more likely to sample correct answers, thereby providing more training data for the next **M-step**. This approach has been proposed before

Algorithm 2 RS

1: **Input:** Data point $(x, y^\star)$, budget $M \in \mathbb{N}$
2: **for** $m = 1, \ldots, M$ **do**
3: $\quad \hat{z}_m, \hat{y}_m \sim \pi(\cdot \mid x; \theta)$
4: $\quad$ **if** $\hat{y}_m = y^\star$ **then**
5: $\quad\quad$ **Output:** $\hat{z}_m, y^\star$
6: **Output:** $\hat{z}_M, \hat{y}_M$

Algorithm 3 STaR

1: **Input:** Data point $(x, y^\star)$
2: $\hat{z}, \hat{y} \leftarrow$ RS$((x, y^\star), 1)$ ▷ Rejection sampling
3: **if** $\hat{y} = y^\star$ **then**
4: $\quad$ **Output:** $\hat{z}, \hat{y}$
5: **else**
6: $\quad \hat{z}', \hat{y}' \sim q_{\text{PPS}}(\cdot \mid x, y^\star; \theta)$ ▷ Rationalization
7: **Output:** $\hat{z}', \hat{y}'$

Prompt and its posterior sampling variant

Task description
An instruction (may include conversation history), a response to evaluate, and a score rubric are given.
Follow these steps:
1. Evaluate the **{judged score}** of the response in detail strictly based on the score rubric.
2. Predict the score of the response from its evaluation. The score is an integer between 0 and 4. The score is an integer between 1 and 7.
3. The output format should look as follows: "Evaluation: (detailed evaluation) Score: (predicted score)"
4. Do not generate any opening or closing statements.

Instruction
{task}

Response to evaluate
{response}

Score rubric
{score rubrics}

Hint
Evaluate the response as if its score was **{conditioning}**. Do not output the hint.

Output

Figure 1: Our prompt and its posterior sampling variant. We set **{task}** to the evaluated task, **{response}** to its solution, **{judged score}** to the name of the metric, and **{score rubrics}** to its score rubrics. In prompt posterior sampling, we add the cyan text and set **{conditioning}** to the human score. The blue text is only in the HelpSteer2 prompt, and the red text is only in the Summarize from Feedback prompt.

under the names of rejection fine-tuning (Yuan et al., 2023; Dong et al., 2023; Singh et al., 2024) and online rejection sampling fine-tuning (Shao et al., 2024, Appendix A.1.3). The pseudo-code of this method is given in Algorithm 2.

Prompt posterior sampling (PPS). Modern LLMs are instruction fine-tuned to follow human feedback (Ouyang et al., 2022). Therefore, one natural way of implementing posterior sampling is by conditioning on the correct answer in the prompt, which we show in Figure 1. This directly addresses the computational inefficiency of RS. Specifically, because the correct answer is conditioned on, the model is likely to generate it and in this way increase the fraction of accepted rationale-answer pairs. We note that this does not *guarantee* that the model outputs the correct answer.

Self-taught reasoner sampling (STaR). Introduced by the seminal work of Zelikman et al. (2022), STaR has the following two stages:

1. A single rejection sampling step (RS for $M = 1$).
2. If the sampled answer is not correct, STaR samples a rationale-answer pair using a rationalization prompt, with the correct answer as a hint. Note that PPS is this step of STaR without rejection sampling in the previous step.

The pseudo-code of STaR is given in Algorithm 3. Conceptually, STaR chains RS and PPS, effectively sampling rationales from a mixture of the two distributions. While this results in more accepted rationale-answer pairs on average per FEM iteration than RS ($M = 1$) and PPS, we show empirically that the increased quantity does not translate into quality. This is because the *quality* of the sampling distribution q is the decisive factor.

Note that PPS and STaR introduce a distribution shift between training and testing: during training, the model generates the intermediate justifications z by conditioning on both the question and the target answer, whereas at test time only the question is available. Nevertheless, our empirical evaluation demonstrates that this mismatch does not degrade performance. This suggests that a target-conditioned proposal distribution q can serve as a highly effective variational approximation in practice, guiding the model to generate high-quality latent traces that generalize well, even if the proposal does not exactly samples from the LVM in (1).

4 Experiments

We evaluate FEM on *LLM-as-a-judge calibration* (Zheng et al., 2023; Sahoo et al., 2025) and *summarization from feedback* (Stiennon et al., 2020) tasks. Our domains have unique properties that make them ideal for investigating rationale sampling schemes:[1]

1. **High information gain from $y^\star$.** The correct answer is a human preference score. The score is readily available in public datasets, and conditioning on it significantly improves the generated rationale.
2. **Weak pre-trained models.** Pre-trained base models tend to perform poorly on our tasks because they lack knowledge about human scores in the domain, even if they are given the same score rubric as humans.

4.1 Setting

We experiment with two Llama (Llama Team, 2024) models (3B and 8B) and three Qwen (Qwen Team, 2024) models (1.5B, 3B, and 7B) across two tasks.[2] The first task is calibration of an LLM judge against human scores. We experiment with *HelpSteer2* dataset (Wang et al., 2024; 2025), a collection of instruction-response pairs rated on a 5-point Likert scale. The quality of the responses is measured by 5 metrics: helpfulness, correctness, coherence, complexity, and verbosity. The second task is predicting human quality scores for post summaries. We also experiment with *Summarize from Feedback* dataset (Völske et al., 2017; Stiennon et al., 2020), a collection of Reddit post-summary pairs rated on a 7-point Likert scale. We use its axis subset, which contains 4 metrics: helpfulness, accuracy, coverage, and coherence. In summary, we evaluate 5 models on predicting 9 metrics in two tasks. Our main prompt is in Figure 1. The system prompt and score rubrics are presented in Appendix B.

We evaluate three FEM sampling schemes: RS with budget $M = 1$, STaR, and PPS; and compare them to pre-trained base models and GPT-5.2. The **E-step** in (Line 3 in FEM) is implemented using vLLM (Kwon et al., 2023) and the **M-step** (Line 5 in FEM) is implemented by SFT in TRL (von Werra et al., 2020). We use AdamW (Loshchilov & Hutter, 2019) with default parameters and one epoch in SFT. FEM is run for $K = 5$ iterations. The learning rate decays linearly from $\eta = 10^{-6}$ to $\eta = 10^{-7}$ across the iterations, and it is kept constant within each iteration. All experiments are conducted on NVIDIA A100 GPUs with resources

[1]The LLM community has largely focused on complex math or code reasoning, but in those domains, the final answer may provide a weaker or differently structured signal than a human rating. Extending answer-conditioned sampling to such domains may require richer side information, such as solution sketches, verifier feedback, or step-level supervision.

[2]The specific models are Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, and Qwen2.5-7B-Instruct from HuggingeFace.

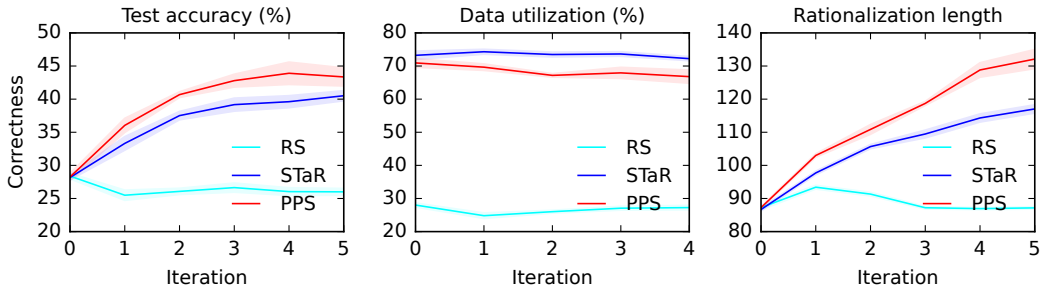

Figure 2: Test accuracy, data utilization, and rationalization length for Qwen 3B and HelpSteer2 correctness. The shaded areas are standard errors of the estimates from 3 random runs.

Evaluation task

Instruction: Bob has two sons. John and Jay. Jay has one brother and father. Father has two sons. Jay's brother has a brother and a father. Who is Jay's brother? **Response:** Jay's brother is John.

Response before first iteration

Evaluation: The response is incorrect because it states that Jay's brother is John, while the instruction clearly indicates that Bob has two sons named John and Jay. Therefore, Jay's brother cannot be John. The response does not correctly address the question posed. **Score:** 0

Response after final iteration

Evaluation: The response accurately identifies Jay's brother as John based on the information provided in the prompt. The reasoning given aligns perfectly with the facts presented: Bob has two sons named John and Jay. Since Jay has one brother (John) and a father (Bob), and John also has a brother (Jay) and a father (Bob), it logically follows that Jay's brother must be John. There are no instances of incorrect information, misleading details, or hallucinations present in the response. The explanation clearly connects the relationships described in the prompt, making it a precise and correct representation of the scenario. Therefore, the response meets the criteria for a score of 4. **Score:** 4

Figure 3: A qualitative example of a rationale improvement after learning with PPS for Qwen 3B and HelpSteer2 correctness.

scaled by model size: 1B models on 1 GPU, 3B models on 2 GPUs, and 7-8B models on 4 GPUs. All results are averaged over 3 random runs with seeds 257, 521, and 1031. We sample a random subset of 4 000 data points (2 000 train and 2 000 test) in each run. The sizes are chosen so that the initialization overhead of vLLM and SFT is negligible in the total runtime (less than 20%). Each run takes approximately 60 minutes.

We report three metrics: test accuracy, data utilization, and rationalization length in words. The test accuracy is the accuracy in predicting human evaluation scores on a held-out test set, which we expect to improve as the number of FEM iterations increases. The data utilization is the percentage of accepted rationale-answer pairs, which are used for training. We report this number to show that more successful learning is not necessarily due to more training data. The rationalization length shows whether the improved performance is due to deeper or more compact thinking, and that there is no collapse.

4.2 Main Results

Our results for Qwen 3B and HelpSteer2 correctness are reported in Figure 2. The plots for all other models and metrics are reported in Appendix D. We observe two key trends. First, the test accuracies of STaR and

Table 1: Highest test accuracies by all sampling schemes (PPS is shaded) on all metrics in HelpSteer2 dataset.

Model	Metric	Base	RS	STaR	PPS
Qwen 7B	Helpfulness	31.50	30.38	**37.97**	36.73
	Correctness	26.56	24.05	40.50	**43.73**
	Coherence	24.27	22.09	55.73	**67.95**
	Complexity	28.84	44.71	53.03	**53.99**
	Verbosity	15.56	17.97	**61.90**	61.85
Qwen 3B	Helpfulness	30.38	29.99	36.91	**37.71**
	Correctness	28.38	26.64	40.51	**43.91**
	Coherence	37.00	34.51	62.73	**67.55**
	Complexity	15.99	23.12	50.98	**51.60**
	Verbosity	23.02	42.92	58.99	**59.04**
Qwen 1B	Helpfulness	25.45	27.84	29.86	**30.91**
	Correctness	23.04	23.60	32.78	**33.01**
	Coherence	18.02	21.96	60.86	**65.52**
	Complexity	15.22	18.23	45.48	**47.99**
	Verbosity	23.62	38.02	55.42	**55.63**
Llama 8B	Helpfulness	34.86	36.87	36.83	**37.63**
	Correctness	31.99	41.63	42.14	**43.73**
	Coherence	45.27	68.99	67.22	**71.42**
	Complexity	27.89	44.45	56.78	**57.26**
	Verbosity	24.48	51.62	64.36	**66.11**
Llama 3B	Helpfulness	30.22	28.76	32.35	**33.25**
	Correctness	21.55	19.28	39.71	**39.73**
	Coherence	26.53	62.75	**69.81**	69.18
	Complexity	23.97	28.50	54.40	**54.66**
	Verbosity	38.26	50.24	61.50	**61.71**

Table 2: Highest test accuracies by all sampling schemes (PPS is shaded) on all metrics in Summarize from Feedback dataset.

Model	Metric	Base	RS	STaR	PPS
Qwen 7B	Helpfulness	17.85	18.76	**22.61**	**22.61**
	Accuracy	8.54	10.35	67.32	**71.37**
	Coverage	19.22	19.03	19.98	**21.12**
	Coherence	5.14	5.26	71.32	**73.04**
Qwen 3B	Helpfulness	17.17	17.73	19.47	**20.01**
	Accuracy	8.74	9.08	26.67	**32.71**
	Coverage	12.91	14.16	15.40	**16.54**
	Coherence	15.33	15.43	69.03	**72.90**
Qwen 1B	Helpfulness	13.86	14.56	20.18	**20.64**
	Accuracy	22.83	36.93	58.24	**66.14**
	Coverage	8.08	9.30	23.04	**24.75**
	Coherence	17.27	20.11	74.51	**76.26**
Llama 8B	Helpfulness	13.96	16.34	20.56	**20.93**
	Accuracy	9.89	15.42	**73.26**	72.50
	Coverage	17.59	18.44	22.67	**24.00**
	Coherence	11.41	61.70	71.17	**74.22**
Llama 3B	Helpfulness	14.14	18.61	**23.73**	22.19
	Accuracy	11.38	14.21	**69.49**	69.20
	Coverage	14.03	18.08	**27.15**	26.47
	Coherence	5.33	7.44	71.33	**76.66**

PPS increase with more iterations and plateau, confirming the effectiveness of multiple FEM iterations. Second, PPS outperforms all baselines across all FEM iterations, which we attribute to generating superior rationales.

Crucially, the data utilization shows that this performance gain is not driven by data quantity; PPS achieves a higher accuracy with less training data than STaR. Instead, it is driven by the quality of rationalization. The rationalization length of PPS increases to 130 words versus less than 120 for STaR. Qualitatively, the longer rationalization is more comprehensive, and we illustrate it in Figure 3. The final rationale resolves confusion in the original rationale, which says that Jay and John cannot be brothers because they have the same father. This mirrors observations in RL-based training (Guo et al., 2025) where models learn to reason more deeply.

We want to stress that this observed trend is model- and domain-dependent, as can be seen in Appendix D. In some cases, the rationalization length *decreases* while accuracy improves. This suggests that FEM does not blindly encourage rationalization length, but rather learns to reason most suitably for the specific task and model: deeper rationalization for complex judgments, or improving conciseness and compactness of the language.

We summarize the highest accuracies attained by all sampling schemes in Tables 1 and 2. In HelpSteer2 experiments (Table 1), PPS has the highest accuracy in 22 out of 25 cases and STaR in 3. Under the null hypothesis that PPS and STaR perform identically, the p-value of winning at least 22 times is less than 0.01. In Summarize from Feedback experiments (Table 2), PPS has the highest accuracy in 15 out of 20 cases, STaR in 4, and they tie in one. Under the null hypothesis that PPS and STaR perform identically, the p-value of winning at least 15 times is about 0.02. We conclude that PPS outperforms the closest baseline with a high statistical significance.

To further show that our tasks are non-trivial for state-of-the-art LLMs, we evaluate GPT-5.2 as a strong baseline. In HelpSteer2 dataset, GPT-5.2 achieves accuracies 32.91% (helpfulness), 30.84% (correctness), 65.08% (coherence), 58.88% (complexity), and 43.19% (verbosity). Our best fine-tuned models surpass these

numbers in all metrics but complexity. In Summarize from Feedback dataset, GPT-5.2 achieves 12.40% (helpfulness), 69.91% (accuracy), 16.40% (coverage), and 50.43% (coherence). Our best fine-tuned models surpass all of these. This confirms that our tasks genuinely benefit from adaptation to the domain.

4.3 Ablation Studies

We investigate the robustness of our findings through three ablation studies with Qwen 3B (Appendix C). First, we vary the rejection sampling budget $M \in \{1, 3, 5\}$ in RS. While increasing M sometimes improves performance through increased data utilization, it also increases computation time linearly in M, and still significantly underperforms PPS and STaR. Second, we test the sensitivity of our results to the prompt in Figure 1 by conditioning less explicitly. We observe that less precise conditioning decreases absolute performance of all sampling schemes. However, the relative order remains the same, and PPS continues to outperform all baselines. This shows that our results generalize beyond a single prompt and highlights the importance of precise instructions when conditioning. Finally, we also experiment with smaller and larger training set sizes.

5 Related Work

Our filtered EM (FEM) framework unifies several recent approaches to iterative fine-tuning (IFT) under the lens of inference. In this section, we explicate the connections to specific EM-style algorithms, data filtering strategies, and the broader RL as inference literature.

IFT. FEM encompasses several recent IFT algorithms. Notably, *ReSTEM* (Singh et al., 2024) corresponds to instantiating the rationale proposal distribution q in Algorithm 1 with standard *rejection sampling* (Neal & Hinton, 1998). This relationship arises from a specific modeling choice: following the variational perspective of Dayan & Hinton (1997), Singh et al. (2024) introduce a binary optimality variable $\mathcal{O}$ and maximize $\mathbb{P}(\mathcal{O} = 1 \mid x)$, treating the entire trajectory—rationale z and answer y—as a latent variable. Consequently, their **E-step** filters samples from a fixed distribution: the LLM conditioned *only* on the question x. Such reliance on a fixed sampling distribution persists in recent literature as well. For instance, *LaTRO* (Chen et al., 2024) introduces a variational framework optimizing a lower bound on correct answer likelihood; Xu (2025) proposes *EM Policy Gradient* to stabilize off-policy updates; and *RAFT++* (Xiong et al., 2025) proposes a filtered version of PPO (Schulman et al., 2017). In all these cases, the algorithms operate under a fixed sampling policy $\pi(z, y \mid x; \theta)$ that conditions only on the question.

Other works have explored alternative proposal distributions, but typically through complex probabilistic machinery or separate inference networks. *TRICE* (Phan et al., 2023) addresses the high variance of gradient estimators in the **M-step** using control variates. *BRiTE* (Zhong et al., 2025) targets the **E-step** by jointly training a separate LLM to approximate the posterior via reinforcement learning. Similarly, Hu et al. (2024) propose amortizing intractable posterior inference using GFlowNets (Bengio et al., 2023); although they do not apply this to iterative LLM training, it shares the limitation of requiring a separate trained network to sample from the true posterior.

Our FEM explores a more lightweight alternative: leveraging the LLM's inherent thinking and instruction-following capabilities (Bai et al., 2022; Ouyang et al., 2022; Wei et al., 2022; Kojima et al., 2022) to approximate q via prompting, thereby avoiding the complexity of training auxiliary models. We demonstrate that in notable domains, such as calibrating LLM-as-a-judge, this EM-based approach can be vastly improved by simply changing the sampling distribution to that of rationales conditioned on the true answer. This highlights that, as prior work has overlooked, examining the sampling scheme offers a simple yet effective direction to further improve the training efficiency.

Filtering and implicit objectives. Beyond iterative algorithms, our work connects to the literature, viewing data filtration as implicit objective modification. Shrivastava et al. (2025) propose oversampling responses and applying top-K filtering, arguing that such filtration acts as a form of reward shaping. Similarly, Karan & Du (2025) shows that inference-time sampling from a sharpened distribution can outperform RL-based post-training. From a distributional perspective, methods like *Quark* (Khalifa et al., 2021) and others (Go et al., 2023; Kruszewski et al., 2025; Kim et al., 2025) derive objectives by defining a target distribution that

discards incorrect answers while preserving the relative probabilities of correct ones, often approximated via information projection (Amari, 1985; Nielsen, 2018).

Our work is complementary to this line of research: we adopt a latent-variable perspective, treating such filtering as an implicit target distribution over rationales and answers, and study how different choices of the *sampling* distribution in the **E-step** affect learning via our FEM. Rather than viewing filtering as a heuristic for dataset cleaning, FEM frames it as a formal approximation of the posterior in a latent variable model.

RL as probabilistic inference. Finally, the derivation of FEM is grounded in the probabilistic formulation of RL, also known as Control as Inference; see our discussions with Lemma 1. The paradigm of addressing RL problems via Expectation-Maximization is well established in control theory (Dayan & Hinton, 1997; Levine, 2018), having been investigated through various lenses, including mixture models (Vlassis & Toussaint, 2009) and MCMC approximations (Hoffman et al., 2007). While this theoretical equivalence is well-documented in general RL, our work systematically operationalizes it for LLM rationalization. We underscore that, in the context of LLMs, the choice of sampling distribution is not merely a heuristic choice but a central component of the variational objective.

6 Conclusions

We revisit rationalization as a latent-variable model, bridging the gap between classic works, filtered EM, and reward-based policy gradients. Our work highlights the critical role of rationale sampling distributions in iterative fine-tuning algorithms, a factor that may have a major impact on empirical performance. In the process, we propose prompt posterior sampling, a simple yet effective sampling scheme corresponding to the rationalization stage of STaR. In our experiments, prompt posterior sampling consistently outperforms other sampling schemes, including rejection sampling and STaR, on two human-rating alignment tasks in HelpSteer2 and Summarize from Feedback datasets.

Limitations. First, we establish that the sampling distribution is an important domain-specific hyperparameter. Our results on alignment tasks provide a definitive proof of concept: the "right" choice of the rationale sampling distribution in FEM leads to substantial empirical gains. In future work, we plan to experiment with more complex tasks, such as math reasoning, and to explore richer guiding signals. Second, our current algorithm design is motivated by SFT: each iteration is essentially a single epoch where training data are sampled from the learned model. In future work, we plan to investigate if our improvements in sampling generalize to online reinforcement learning, such as GRPO (Shao et al., 2024). When viewed in this way, our current design is reminiscent of batch RL (Lange et al., 2012; Jaques et al., 2020; Levine, 2018), also with connections to prior literature on expert iterations (Anthony et al., 2017; Anthony, 2021) Last but not least, while all of our approximations in Section 2.1 are properly justified, we have not formally analyzed FEM, which we also leave for future work. This is because we focused on developing the algorithmic framework and empirically validating it.

References

Shun-ichi Amari. *α-Divergence and α-Projection in Statistical Manifold*, pp. 66–103. Springer New York, New York, NY, 1985. ISBN 978-1-4612-5056-2. doi: 10.1007/978-1-4612-5056-2_3.

Thomas Anthony, Zheng Tian, and David Barber. Thinking Fast and Slow with Deep Learning and Tree Search. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5366–5376. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d8e1344e27a5b08cdfd5d027d9b8d6de-Paper.pdf.

Thomas William Anthony. *Expert Iteration*. Ph.d. thesis, UCL (University College London), 2021. URL https://discovery.ucl.ac.uk/id/eprint/10123580/.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott

Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2204.05862*, 2022. URL https://arxiv.org/abs/2204.05862.

Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77 – 120, 2017. doi: 10.1214/16-AOS1435.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. GFlowNet Foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023. URL http://jmlr.org/papers/v24/22-0364.html.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization Methods for Large-Scale Machine Learning. *SIAM Review*, 60(2):223–311, 2018. doi: 10.1137/16M1080173.

Olivier Cappé and Eric Moulines. On-Line Expectation–Maximization Algorithm for latent Data Models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 71(3):593–613, 02 2009. ISSN 1369-7412. doi: 10.1111/j.1467-9868.2009.00698.x.

Haolin Chen, Yihao Feng, Zuxin Liu, Weiran Yao, Akshara Prabhakar, Shelby Heinecke, Ricky Ho, Phil Mui, Silvio Savarese, Caiming Xiong, and Huan Wang. Language Models are Hidden Reasoners: Unlocking Latent Reasoning Capabilities via Self-Rewarding. *arXiv preprint arXiv:2411.04282*, 2024. URL https://arxiv.org/abs/2411.04282.

Peter Dayan and Geoffrey E. Hinton. Using Expectation-Maximization for Reinforcement Learning. *Neural Computation*, 9(2):271–278, 02 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.2.271.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 12 1977. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1977.tb01600.x.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun Shum, and Tong Zhang. RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=m7p5O7zblY.

Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, New York, NY, 2001.

Dongyoung Go, Tomasz Korbak, Germàn Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning Language Models with Preferences through f-divergence Minimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11546–11583. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/go23a.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang,

Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. URL https://www.nature.com/articles/s41586-025-09422-z.

Matthew Hoffman, Arnaud Doucet, Nando Freitas, and Ajay Jasra. Bayesian Policy Learning with Trans-Dimensional MCMC. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/3a15c7d0bbe60300a39f76f8a5ba6896-Paper.pdf.

Edward J Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Ouj6p4ca60.

Natasha Jaques, Judy Hanwen Shen, Asma Ghandeharioun, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Human-centric dialog training via offline reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

Aayush Karan and Yilun Du. Reasoning with Sampling: Your Base Model is Smarter Than You Think. *arXiv preprint arXiv:2510.14901*, 2025. URL https://arxiv.org/abs/2510.14901.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A Distributional Approach to Controlled Text Generation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jWkw45-9AbL.

Minbeom Kim, Thibaut Thonet, Jos Rozen, Hwaran Lee, Kyomin Jung, and Marc Dymetman. Guaranteed Generation from Large Language Models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8roRgrjbjv.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22199–22213. Curran Associates, Inc., 2022. URL https://openreview.net/forum?id=e2TBb5y0yFf.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Adaptive Computation and Machine Learning. MIT Press, 2009. ISBN 9780262013192.

Germán Kruszewski, Pierre Erbacher, Jos Rozen, and Marc Dymetman. Whatever Remains Must Be True: Filtering Drives Reasoning in LLMs, Shaping Diversity. *arXiv preprint arXiv:2512.05962*, 2025. URL https://arxiv.org/abs/2512.05962.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. *Batch Reinforcement Learning*, pp. 45–73. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

Sergey Levine. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. *arXiv preprint arXiv:1805.00909*, 2018. URL https://arxiv.org/abs/1805.00909.

Dawen Liang and Nikos Vlassis. Local Policy Improvement for Recommender Systems. *arXiv preprint arXiv:2212.11431*, 2022. URL https://arxiv.org/abs/2212.11431.

Llama Team. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yifei Ma, Yu-Xiang Wang, and Balakrishnan Narayanaswamy. Imitation-regularized offline learning. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

T.K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996. doi: 10.1109/79.543975.

Subhojyoti Mukherjee, Viet Dac Lai, Raghavendra Addanki, Ryan Rossi, Seunghyun Yoon, Trung Bui, Anup Rao, Jayakumar Subramanian, and Branislav Kveton. Offline RL by Reward-Weighted Fine-Tuning for Conversation Optimization. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. URL https://openreview.net/forum?id=WAFD6VYIEa.

Radford M. Neal and Geoffrey E. Hinton. *A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants*, pp. 355–368. Springer Netherlands, Dordrecht, 1998. ISBN 978-94-011-5014-9. doi: 10.1007/978-94-011-5014-9_12.

Ronald C. Neath. On Convergence Properties of the Monte Carlo EM Algorithm. In Galin Jones and Xiaotong Shen (eds.), *Advances in Modern Statistical Theory and Applications: A Festschrift in Honor of Morris L. Eaton*, volume 10 of *Institute of Mathematical Statistics Collections*, pp. 43–62. Institute of Mathematical Statistics, 2013. doi: 10.1214/12-IMSCOLL1003.

Frank Nielsen. What is... An Information Projection? *Notices of the AMS*, 65(3):321–324, 2018. URL https://www.ams.org/journals/notices/201803/rnoti-p321.pdf?adat=March%202018&trk=201803321&cat=none&type=.pdf.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Du Phan, Matthew Douglas Hoffman, David Dohan, Sholto Douglas, Tuan Anh Le, Aaron Parisi, Pavel Sountsov, Charles Sutton, Sharad Vikram, and Rif A. Saurous. Training Chain-of-Thought via Latent-Variable Inference. In *Advances in Neural Information Processing Systems*, volume 36, pp. 72819–72841. Curran Associates, Inc., 2023. URL https://openreview.net/forum?id=a147pIS2Co.

Qwen Team. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Aishwarya Sahoo, Jeevana Kruthi Karnuthala, Tushar Parmanand Budhwani, Pranchal Agarwal, Sankaran Vaidyanathan, Alexa Siu, Franck Dernoncourt, Jennifer Healey, Nedim Lipka, Ryan Rossi, Uttaran Bhattacharya, and Branislav Kveton. Quantitative LLM Judges. *arXiv preprint arXiv:2506.02945*, 2025. URL https://arxiv.org/abs/2506.02945.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample More to Think Less: Group Filtered Policy Optimization for Concise Reasoning. *arXiv preprint arXiv:2508.09726*, 2025. URL https://arxiv.org/abs/2508.09726.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, Abhishek Kumar, Alexander A Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura A Culp, Lechao Xiao, Maxwell Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=lNAyUngGFK. Expert Certification.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021. Curran Associates, Inc., 2020. URL https://arxiv.org/abs/2009.01325.

Yunhao Tang, Sid Wang, Lovish Madaan, and Rémi Munos. Beyond Verifiable Rewards: Scaling Reinforcement Learning for Language Models to Unverifiable Data. *arXiv preprint arXiv:2503.19618*, 2025. URL https://arxiv.org/abs/2503.19618.

Nikos Vlassis and Marc Toussaint. Model-Free Reinforcement Learning as Mixture Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 1081–1088, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553512. URL https://orbilu.uni.lu/bitstream/10993/3376/1/09-vlassis-toussaint-ICML.pdf.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to Learn Automatic Summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4508.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouedec. TRL: Transformer Reinforcement Learning. https://github.com/huggingface/trl, 2020.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. HelpSteer 2: Open-source dataset for training top-performing reward models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=PvVKUFhaNy.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. HelpSteer2-Preference: Complementing Ratings with Preferences. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=MnfHxPP5gs.

Greg C. G. Wei and Martin A. Tanner. A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990. doi: 10.1080/01621459.1990.10474930.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A Minimalist Approach to LLM Reasoning: from Rejection Sampling to Reinforce. *arXiv preprint arXiv:2504.11343*, 2025. URL https://arxiv.org/abs/2504.11343.

Tianbing Xu. Training Large Language Models to Reason via EM Policy Gradient. *arXiv preprint arXiv:2504.18587*, 2025. URL https://arxiv.org/abs/2504.18587.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling Relationship on Learning Mathematical Reasoning with Large Language Models. *arXiv preprint arXiv:2308.01825*, 2023. URL https://arxiv.org/abs/2308.01825.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping Reasoning With Reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488. Curran Associates, Inc., 2022. URL https://openreview.net/forum?id=_3ELRdg2sgI.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36*, 2023.

Han Zhong, Yutong Yin, Shenao Zhang, Xiaojun Xu, Yuanxin Liu, Yifei Zuo, Zhihan Liu, Boyi Liu, Sirui Zheng, Hongyi Guo, Liwei Wang, Mingyi Hong, and Zhaoran Wang. BRiTE: Bootstrapping Reinforced Thinking Process to Enhance Language Model Reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 13–19 Jul 2025. URL https://openreview.net/forum?id=NME3HKUHLX.

A Proof of Lemma 1

The lemma follows from simple algebraic manipulations,

$$\begin{aligned}
\mathbb{E}_{\hat{z},\hat{y}\sim\pi(\cdot\mid x;\theta)}[r(\hat{y},y^\star)] &= \mathbb{E}_{\hat{z},\hat{y}\sim q(\cdot\mid x,y^\star;\theta^{(k-1)})}\left[r(\hat{y},y^\star)\frac{\pi(\hat{z},\hat{y}\mid x;\theta)}{q(\hat{z},\hat{y}\mid x,y^\star;\theta^{(k-1)})}\right] \\
&\geq \mathbb{E}_{\hat{z},\hat{y}\sim q(\cdot\mid x,y^\star;\theta^{(k-1)})}\left[r(\hat{y},y^\star)\left(1+\log\frac{\pi(\hat{z},\hat{y}\mid x;\theta)}{q(\hat{z},\hat{y}\mid x,y^\star;\theta^{(k-1)})}\right)\right] \qquad (z\geq 1+\log z) \\
&= \mathbb{E}_{\hat{z},\hat{y}\sim q(\cdot\mid x,y^\star;\theta^{(k-1)})}[r(\hat{y},y^\star)\log\pi(\hat{z},\hat{y}\mid x;\theta)] + \\
&\quad \underbrace{\mathbb{E}_{\hat{z},\hat{y}\sim q(\cdot\mid x,y^\star;\theta^{(k-1)})}\left[r(\hat{y},y^\star)\left(1-\log q(\hat{z},\hat{y}\mid x,y^\star;\theta^{(k-1)})\right)\right]}_{\triangleq C},
\end{aligned}$$

where $C\geq 0$ is independent of θ. The tightness follows from the fact that $z=1+\log z$ if and only if $z=1$. $\square$

B Prompts

The main prompt is shown in Figure 1. The system prompt is:

System prompt

You are a fair judge assistant tasked with providing clear objective feedback based on specific criteria, ensuring each assessment reflects the absolute standards set for performance.

Score rubrics for HelpSteer2 dataset are:

Rubric (Helpfulness)

Score 0: The response is not useful or helpful at all. The response completely missed the essence of what the user wanted.
Score 1: The response is borderline unhelpful and mostly does not capture what the user was looking for, but it is still usable and helpful in a small way.
Score 2: The response is partially helpful but misses the overall goal of the user's query/input in some way. The response did not fully satisfy what the user was looking for.
Score 3: The response is mostly helpful and mainly aligned with what the user was looking for, but there is still some room for improvement.
Score 4: The response is extremely helpful and completely aligned with the spirit of what the prompt was asking for.

Rubric (Correctness)

Score 0: The response is completely incorrect. All information provided is wrong, false or hallucinated. If the prompt asks the assistant to do a task, the task is not at all attempted, or the wrong task was attempted in the response. The response is completely irrelevant to the prompt.
Score 1: The response has some correct elements but is mostly wrong or incomplete. The response may contain multiple instances of hallucinations, false information, misleading information, or irrelevant information. If the prompt asks the assistant to do a task, the task was attempted with a small amount of success.
Score 2: The response contains a mix of correct and incorrect information. The response may miss some details, contain misleading information, or minor hallucinations, but is more or less aligned with what the prompt asks for. If the prompt asks the assistant to perform a task, the task is attempted with moderate success but still has clear room for improvement.
Score 3: The response is mostly accurate and correct with a small amount of missing information. It contains no misleading information or hallucinations. If the prompt asks the assistant to perform a task, the task is mostly successfully attempted.
Score 4: The response is completely correct and accurate to what is requested by the prompt with no necessary details missing and without false, misleading, or hallucinated information. If the prompt asks the assistant to do a task, the task is completely done and addressed in the response.

Rubric (Coherence)

Score 0: (Completely Incoherent and/or Unclear) – The response is completely incomprehensible and no clear meaning or sensible message can be discerned from it.
Score 1: (Mostly Incoherent and/or Unclear) – The response is mostly hard to follow, with inconsistencies, contradictions, confusing logic flow, or unclear language used throughout, but there are some coherent/clear parts.
Score 2: (A Little Unclear and/or Incoherent) – The response is a little unclear. There are some inconsistencies or contradictions, run on sentences, confusing statements, or hard to follow sections of the response.
Score 3: (Mostly Coherent and Clear) – The response is mostly clear and coherent, but there may be one or two places where the wording is confusing or the flow of the response is a little hard to follow. Over all, the response can mostly be followed with a little room for improvement.
Score 4: (Perfectly Coherent and Clear) – The response is perfectly clear and self-consistent throughout. There are no contradictory assertions or statements, the writing flows logically and following the train of thought/story is not challenging.

Rubric (Complexity)

Score 0: (Basic) – The response uses very easy to understand language that is clear and completely interpretable by children, adults, and anyone with a functional command of the language.
Score 1: (Simple) – The response uses relatively straightforward language and wording, but some schooling through elementary or a middle school in the language might be required to understand the response.
Score 2: (Intermediate) – People who have completed up through a high school education will probably be able to understand the vocabulary and sentence structure used, but those at the basic level or children might struggle to understand the response.
Score 3: (Advanced) – The response uses a fairly sophisticated vocabulary and terminology. Someone majoring in this subject at a college or university could have written it and would understand the response. An average adult who does not work or study in this area could not have written the response.
Score 4: (Expert) – An expert in the field or area could have written the response. It uses specific and technically relevant vocabulary. Elevated language that someone at the simple or basic level may not understand at all. The professional language of a lawyer, scientist, engineer, or doctor falls into this category.

Rubric (Verbosity)

Score 0: (Succinct) – The response is short, to the point, and the most concise it can be. No additional information is provided outside of what is requested by the prompt (regardless of if the information or response itself is incorrect, hallucinated, or misleading. A response that gives an incorrect answer can still be succinct.).
Score 1: (Pretty Short) – The response is on the shorter side but could still have words, details, and/or text removed before it's at a bare minimum of what the response is trying to convey.
Score 2: (Average Length) – The response isn't especially long or short given what the prompt is asking of the model. The length is adequate for conveying a full response but isn't particularly wordy nor particularly concise.
Score 3: (Moderately Long) – The response is on the longer side but could still have more added to it before it is considered fully detailed or rambling.
Score 4: (Verbose) – The response is particularly lengthy, wordy, and/or extensive with extra details given what the prompt requested from the assistant model. The response can be verbose regardless of if the length is due to repetition and incoherency or if it is due to rich and insightful detail.

Score rubrics for Summarize from Feedback dataset are:

Rubric (Helpfulness)

Score 1: The response is terrible.
Score 4: The response is an okay representation of the post, but could be significantly improved.
Score 7: The response is an excellent representation of the post.

Rubric (Accuracy)

Score 1: The response is completely wrong, made up, or exactly contradicts what is written in the post.
Score 4: The response says at least one substantial thing that is not mentioned in the post, or that contradicts something in the post.
Score 7: The response has no incorrect statements or misleading implications.

Rubric (Coverage)

Score 1: The response contains no information relevant to the post.
Score 4: The response is missing at least 1 important piece of information required to understand the situation.
Score 7: The response covers all of the important information required to understand the situation.

Rubric (Coherence)

Score 1: The response is impossible to understand.
Score 4: The response has mistakes or confusing phrasing that make it a bit hard to understand.
Score 7: The response is perfectly clear.

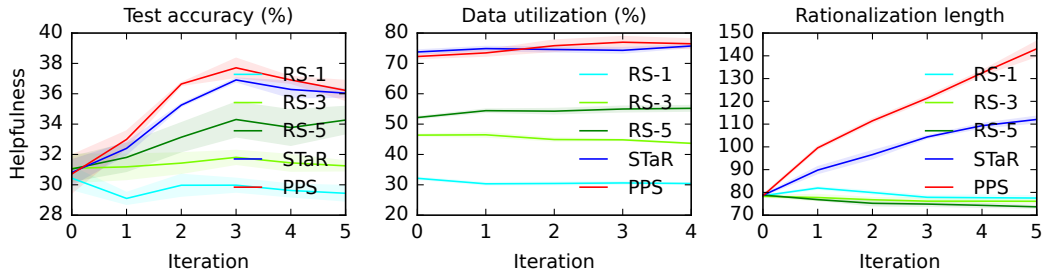

Figure 4: Test accuracy, data utilization, and rationalization length for Qwen 3B and HelpSteer2 helpfulness.

Table 3: Highest test accuracies for Qwen 3B on all metrics in HelpSteer2 dataset across $M \in \{1, 3, 5\}$ for RS.

Metric	Base	RS-1	RS-3	RS-5	STaR	PPS
Helpfulness	30.44	29.97	31.81	34.30	36.91	**37.71**
Correctness	28.30	26.59	27.96	27.86	40.51	**43.91**
Coherence	36.75	34.28	36.00	37.56	62.73	**67.55**
Complexity	15.94	22.68	37.91	40.43	50.98	**51.60**
Verbosity	23.09	42.82	56.16	58.37	58.99	**59.04**

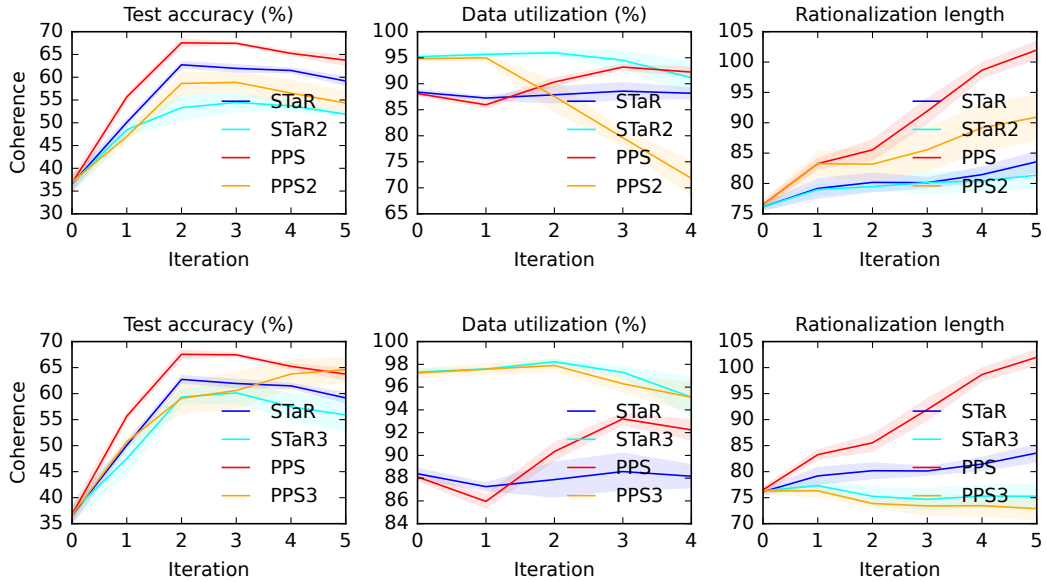

Figure 5: Test accuracy, data utilization, and rationalization length for Qwen 3B and HelpSteer2 coherence.

C Ablations Studies

We experiment with various rejection sampling budgets $M \in \{1, 3, 5\}$ in RS. In Figure 4, we show an example where the accuracy clearly increases with M. This gain is due to increased data utilization. However, even with this improvement, RS is less accurate than both PPS and STaR. In addition, its computation time is (in the worst-case) linear in M and thus M times higher than that of PPS. We summarize the highest accuracies attained by all sampling schemes on HelpSteer2 dataset in Table 3. In all cases, the accuracies of PPS are higher than those of any rejection sampling scheme.

We further examine robustness to the conditioning prompt. We experiment with two variants of the hint in Figure 1. The first is "The score of the response should be **{conditioning}**. Do not output the hint.",

Table 4: Highest test accuracies for Qwen 3B on all metrics in HelpSteer2 dataset across different variants of PPS.

Metric	Base	RS	STaR	PPS	STaR2	PPS2	STaR3	PPS3
Helpfulness	30.75	29.90	36.91	**37.71**	34.83	36.05	32.64	35.76
Correctness	28.07	26.68	40.51	**43.91**	36.96	37.43	36.09	38.65
Coherence	37.07	34.74	62.73	**67.55**	54.47	58.88	60.14	64.60
Complexity	15.66	22.47	50.98	51.60	51.29	**53.09**	51.29	52.96
Verbosity	22.91	43.13	58.99	59.04	56.90	57.94	56.62	**59.37**

Table 5: Highest test accuracies for Qwen 3B on all metrics in HelpSteer2 dataset across different dataset sizes.

Dataset size N	Metric	Base	RS	STaR	PPS
1000	Helpfulness	30.74	30.87	36.84	**38.78**
	Correctness	28.10	27.10	34.97	**37.22**
	Coherence	36.76	35.63	**58.89**	55.43
	Complexity	15.66	18.79	**43.81**	43.35
	Verbosity	22.99	30.18	**39.94**	36.68
2000	Helpfulness	30.38	29.99	36.91	**37.71**
	Correctness	28.38	26.64	40.51	**43.91**
	Coherence	37.00	34.51	62.73	**67.55**
	Complexity	15.99	23.12	50.98	**51.60**
	Verbosity	23.02	42.92	58.99	**59.04**
4000	Helpfulness	30.42	31.79	**36.74**	36.35
	Correctness	27.87	26.56	38.58	**40.43**
	Coherence	36.74	39.56	62.31	**68.83**
	Complexity	15.66	37.05	53.04	**53.37**
	Verbosity	23.32	56.49	59.01	**60.34**

and we denote the corresponding algorithms by PPS2 and STaR2. The second is "The score of the response should be **{conditioning}**.". The corresponding algorithms are denoted by PPS3 and STaR3. We show plots for all prompts and HelpSteer2 coherence in Figure 5. While the absolute accuracy of the new prompts decreases, the relative ranking of the sampling schemes remains unchanged, confirming that our findings generalize beyond a single prompt. Crucially, this accuracy drop highlights that the prompt formulation matters: precise conditioning that explicitly instructs the LLM *how* to use the evidence yields better results than simply stating the evidence as a fact. We summarize the highest accuracies attained by all sampling schemes on HelpSteer2 dataset in Table 4. In all cases, PPS has higher accuracies than STaR, PPS2 has higher accuracies than STaR2, and PPS3 has higher accuracies than STaR3.

Finally, we experiment with various dataset sizes $N \in \{1\,000, 2\,000, 4\,000\}$ in Table 5. We observe that when the dataset is much smaller than in our experiment ($N = 1\,000$), both STaR and PPS perform poorly and there is no major difference in their accuracies. On the other hand, when the dataset is much larger ($N = 4\,000$), both STaR and PPS perform comparably to $N = 2\,000$ and our earlier reported trends hold.

D Detailed FEM Runs

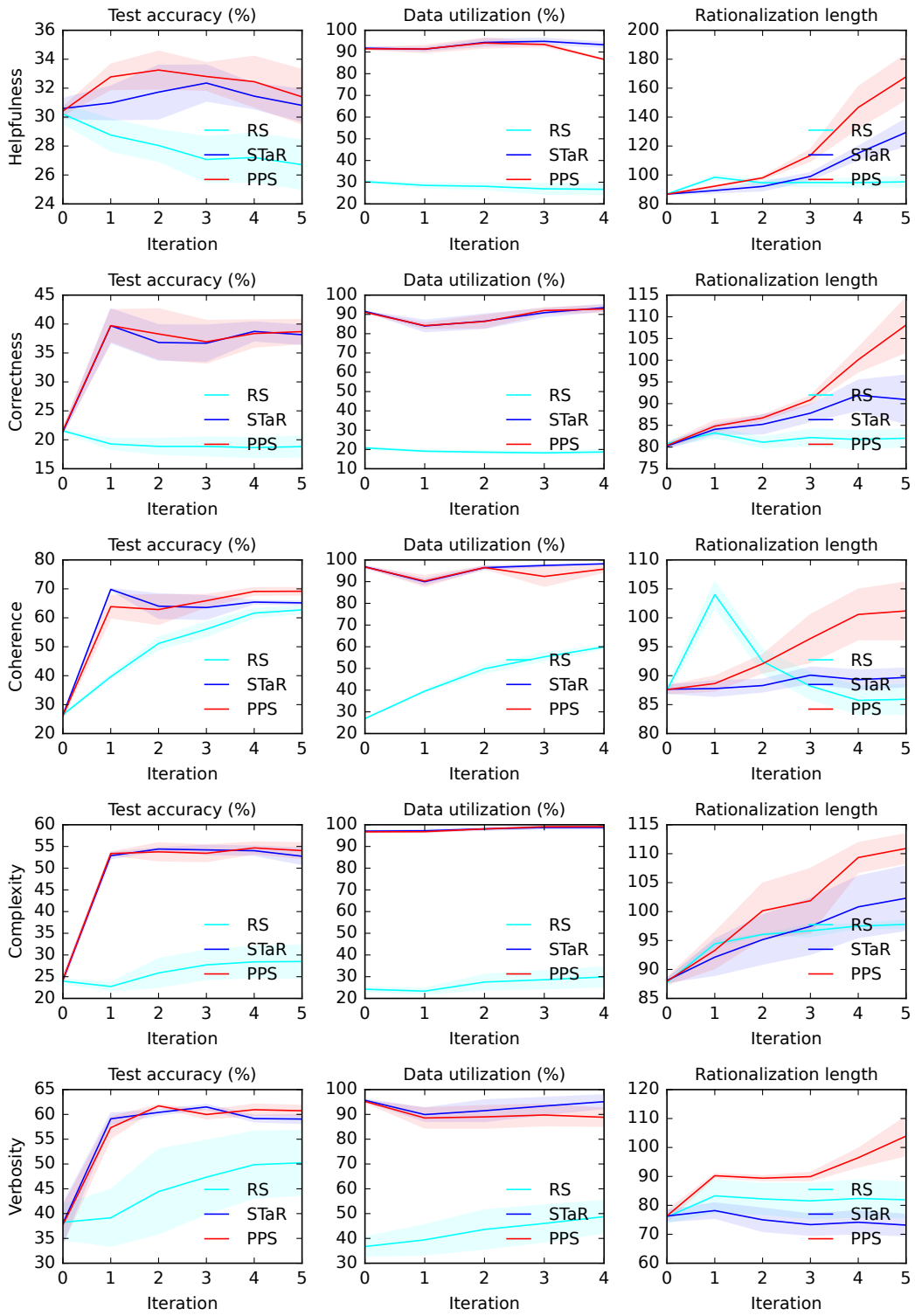

Figure 6: Test accuracy, data utilization, and rationalization length for Llama 3B and HelpSteer2 dataset.

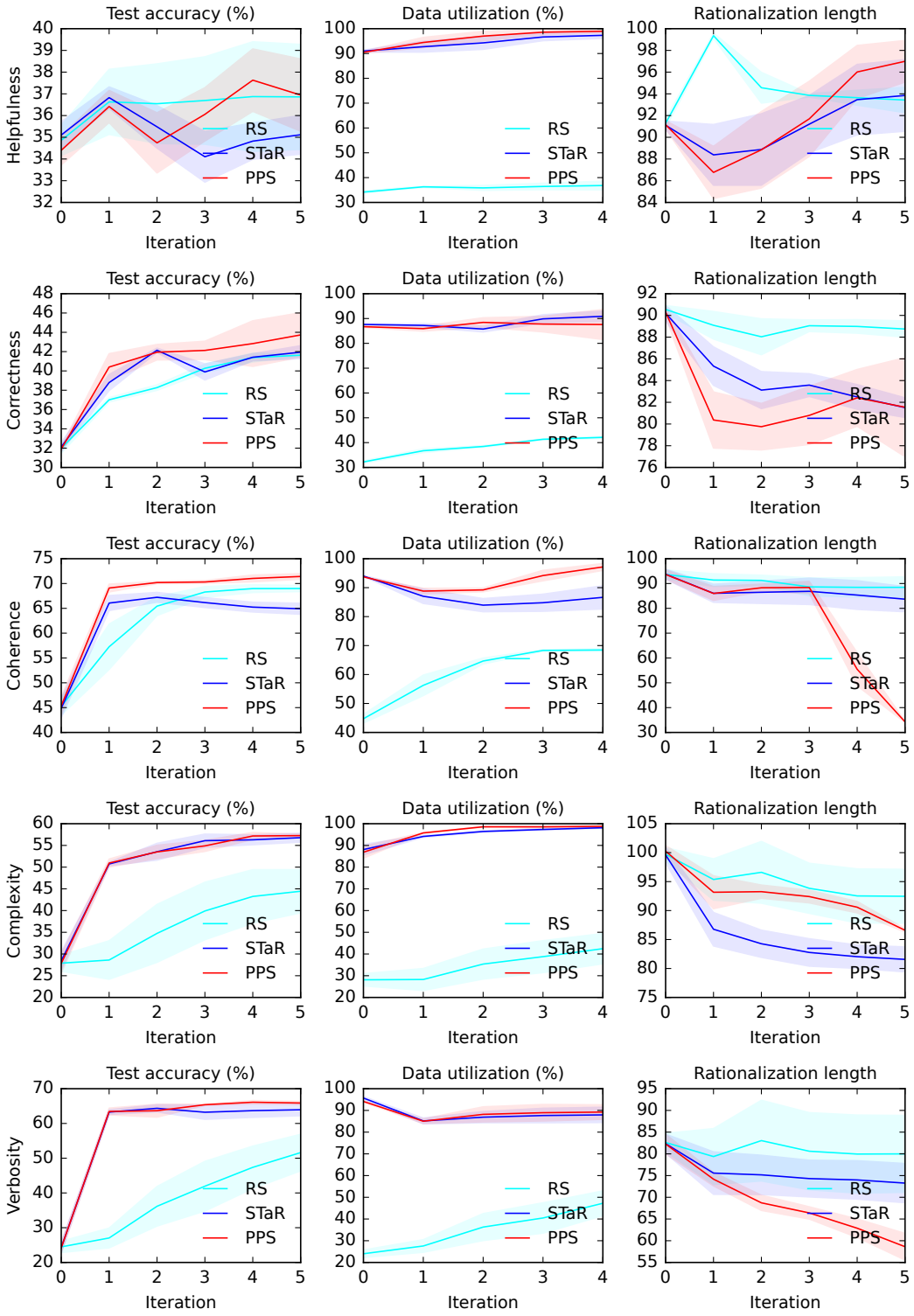

Figure 7: Test accuracy, data utilization, and rationalization length for Llama 8B and HelpSteer2 dataset.

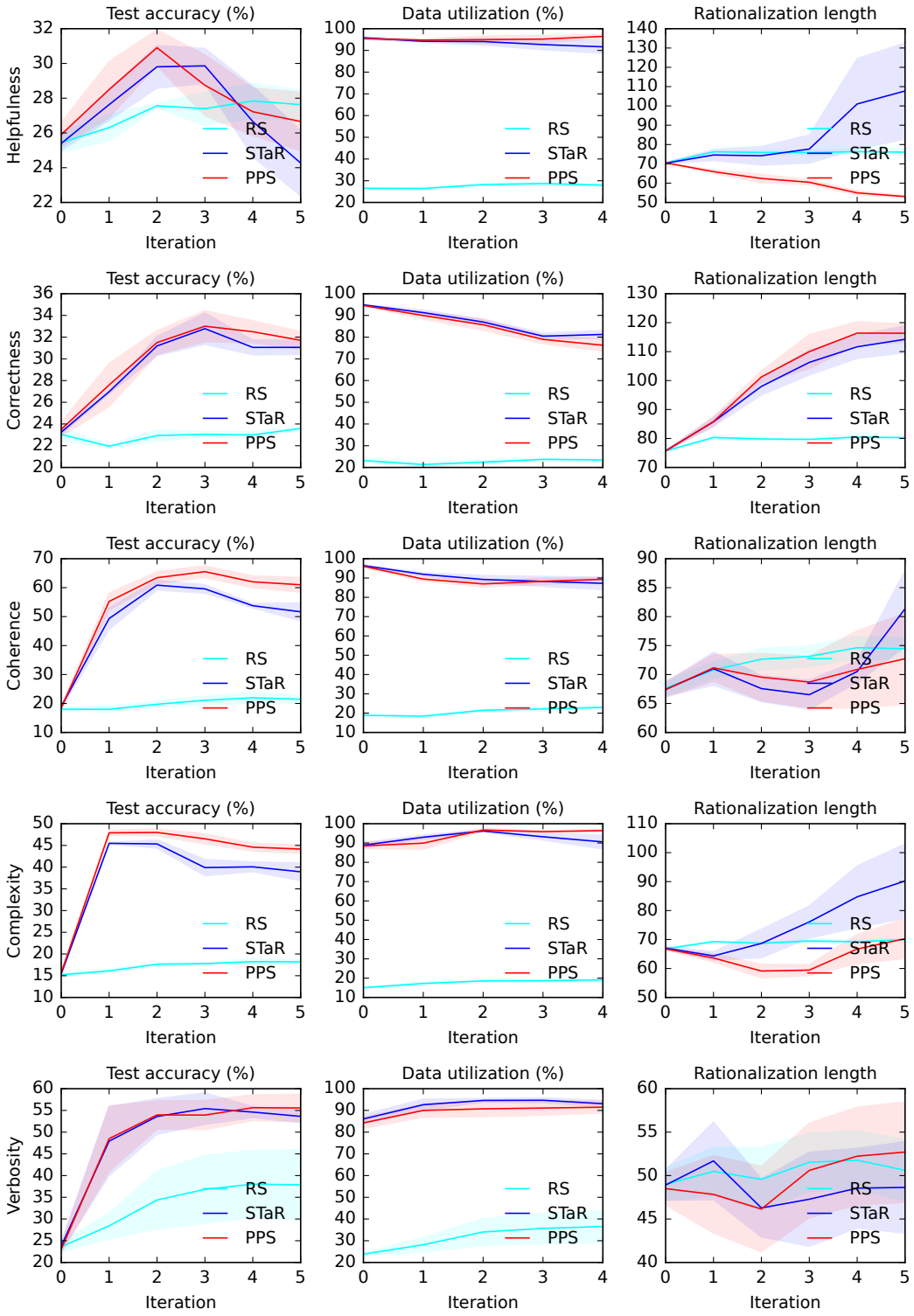

Figure 8: Test accuracy, data utilization, and rationalization length for Qwen 1B and HelpSteer2 dataset.

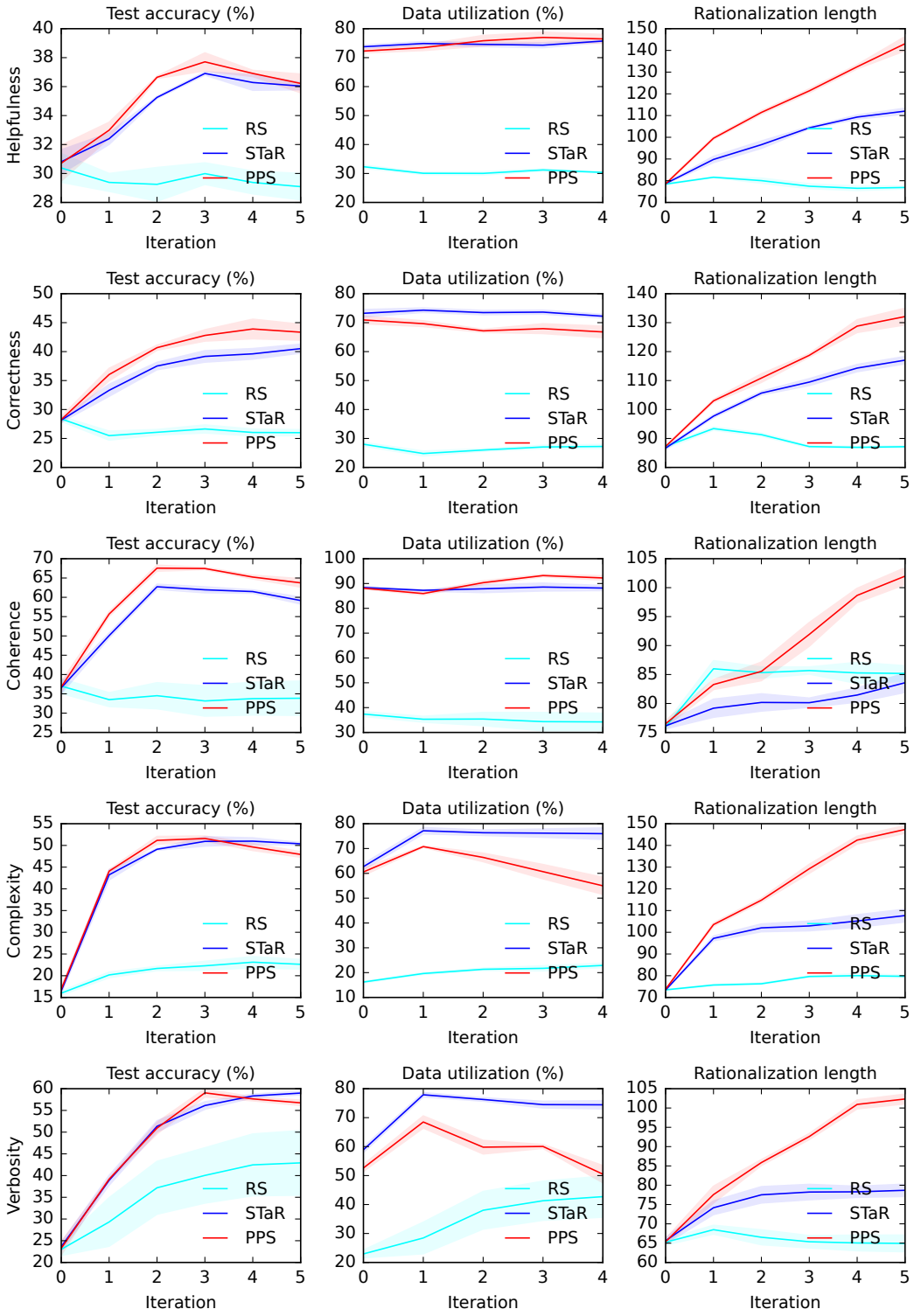

Figure 9: Test accuracy, data utilization, and rationalization length for Qwen 3B and HelpSteer2 dataset.

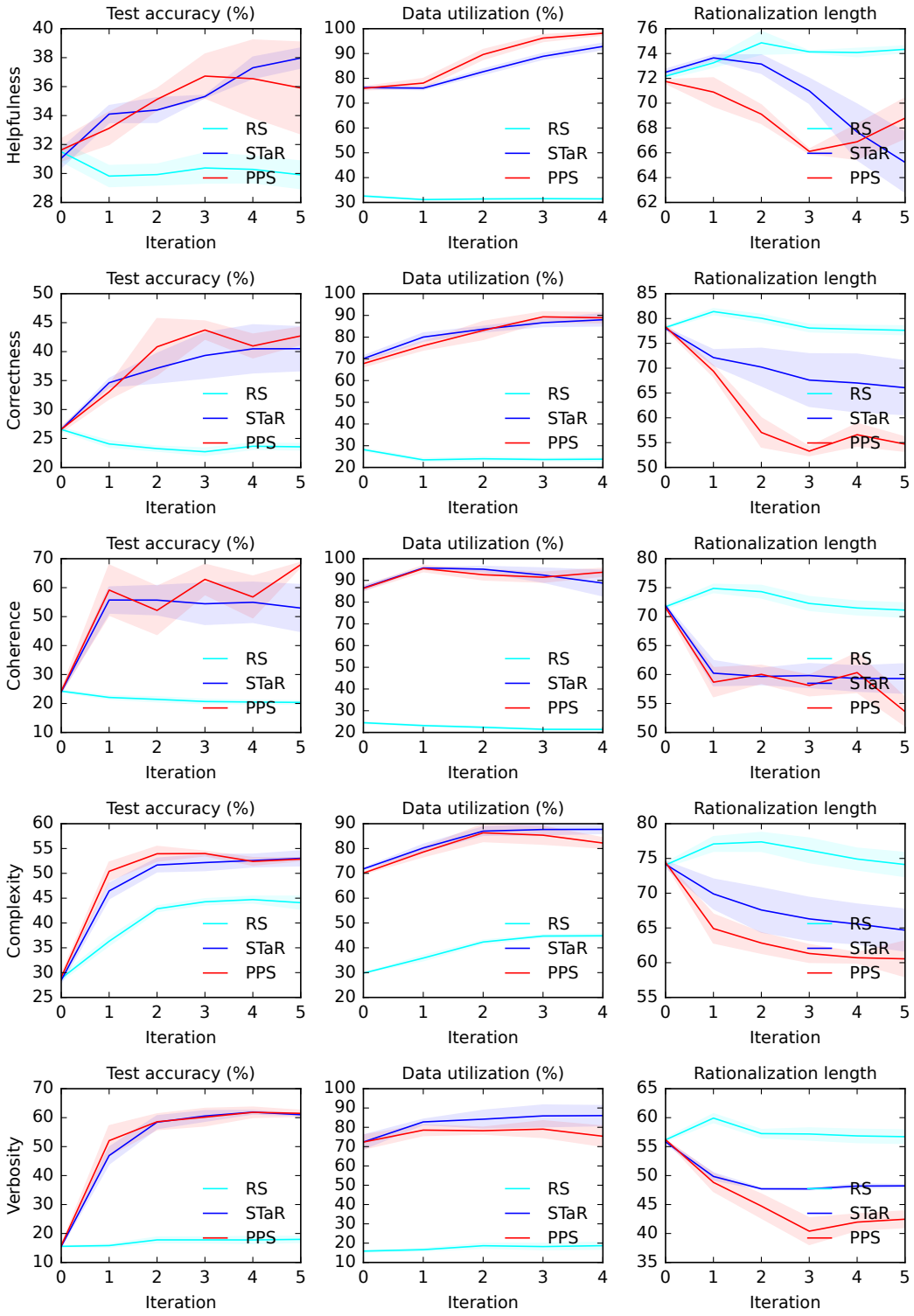

Figure 10: Test accuracy, data utilization, and rationalization length for Qwen 7B and HelpSteer2 dataset.

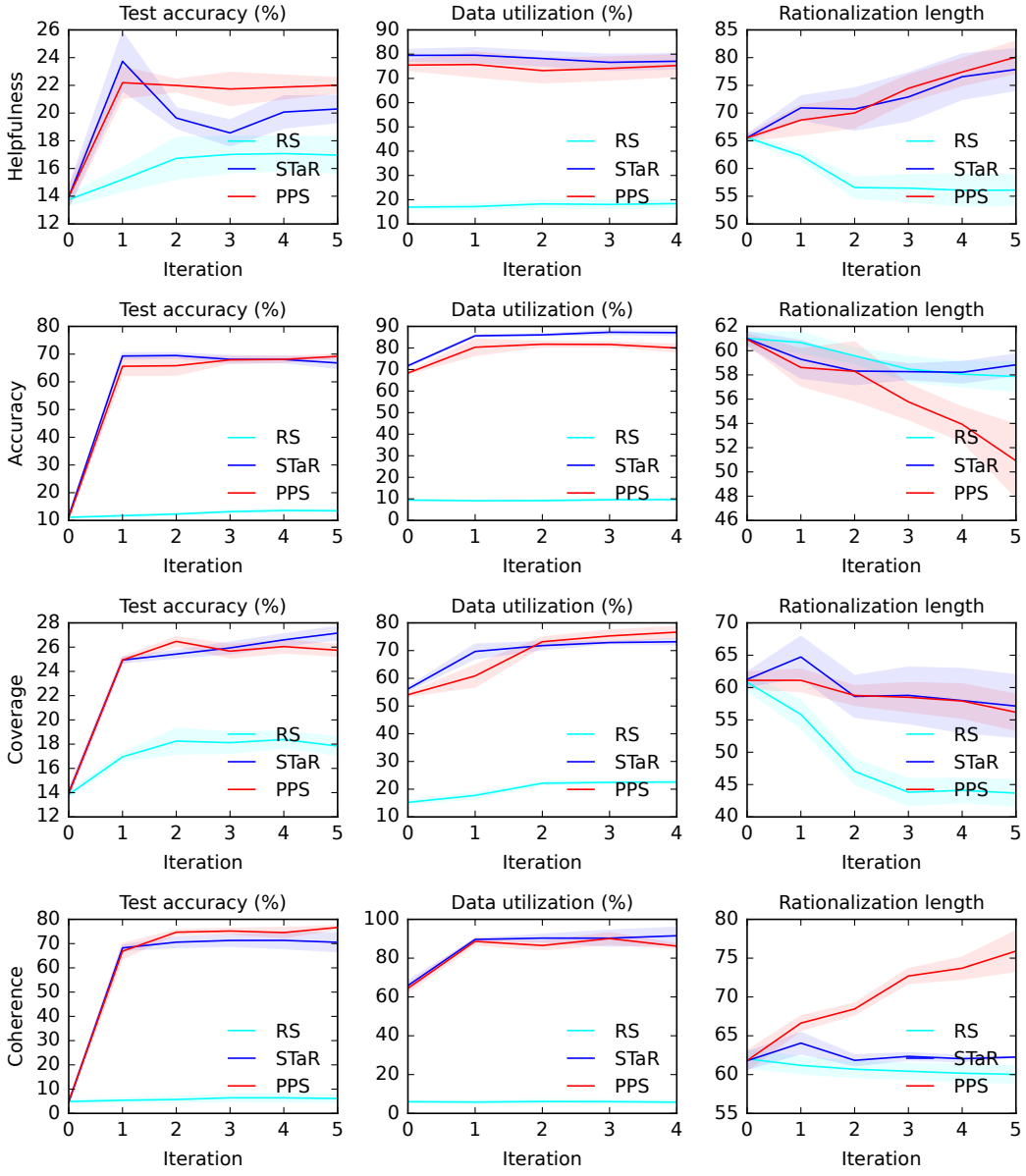

Figure 11: Test accuracy, data utilization, and rationalization length for Llama 3B and Summarize from Feedback dataset.

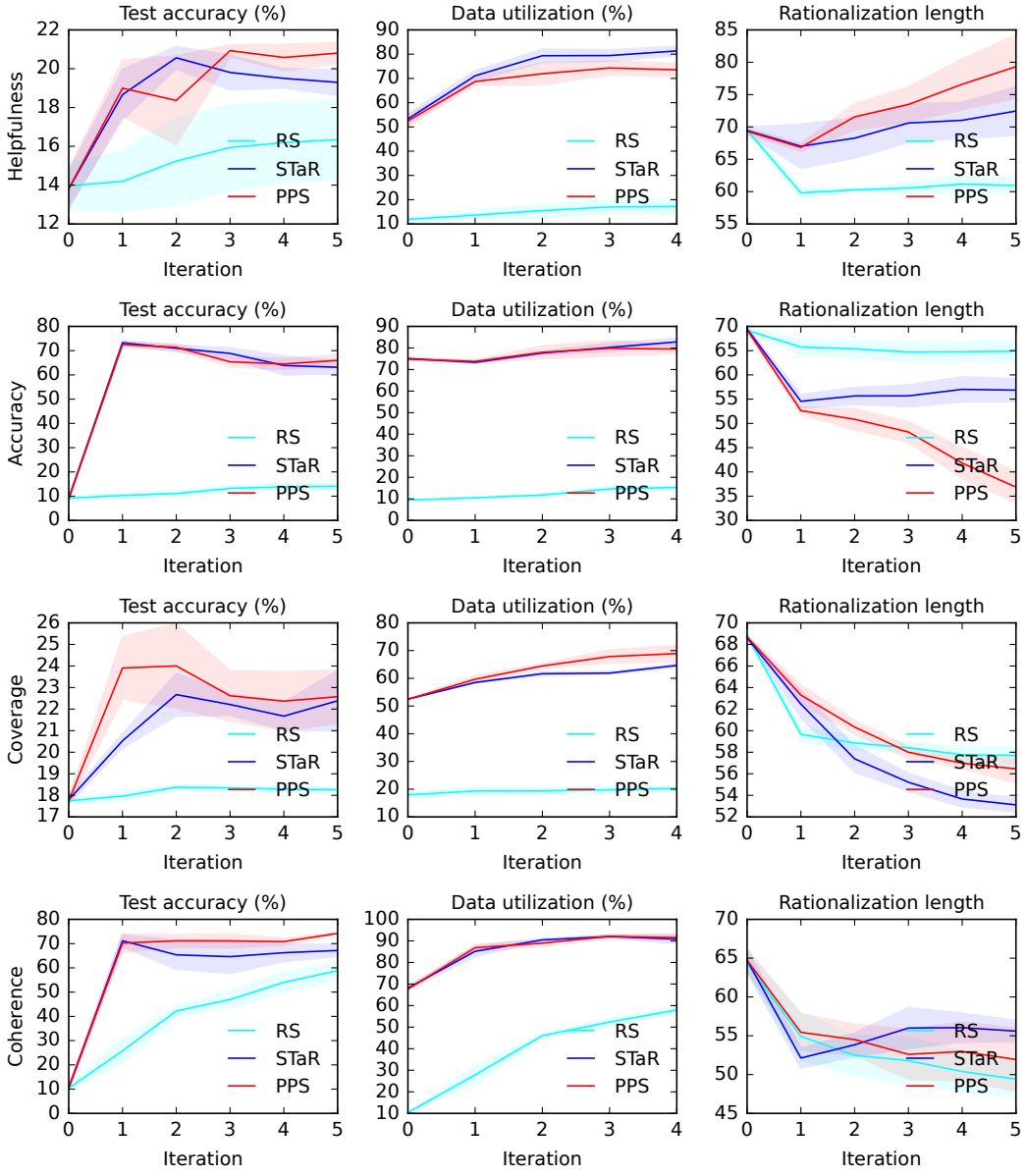

Figure 12: Test accuracy, data utilization, and rationalization length for Llama 8B and Summarize from Feedback dataset.

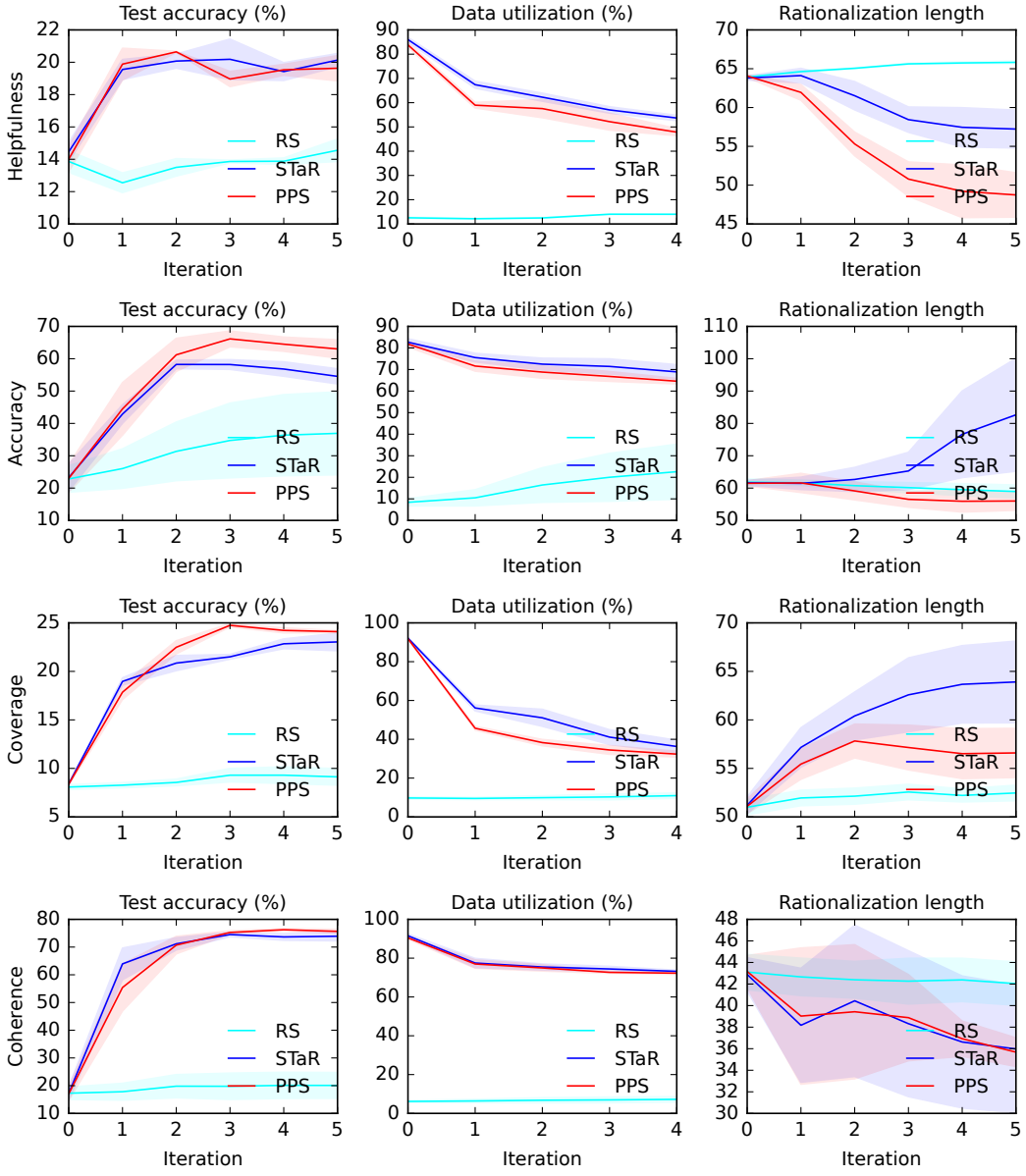

Figure 13: Test accuracy, data utilization, and rationalization length for Qwen 1B and Summarize from Feedback dataset.

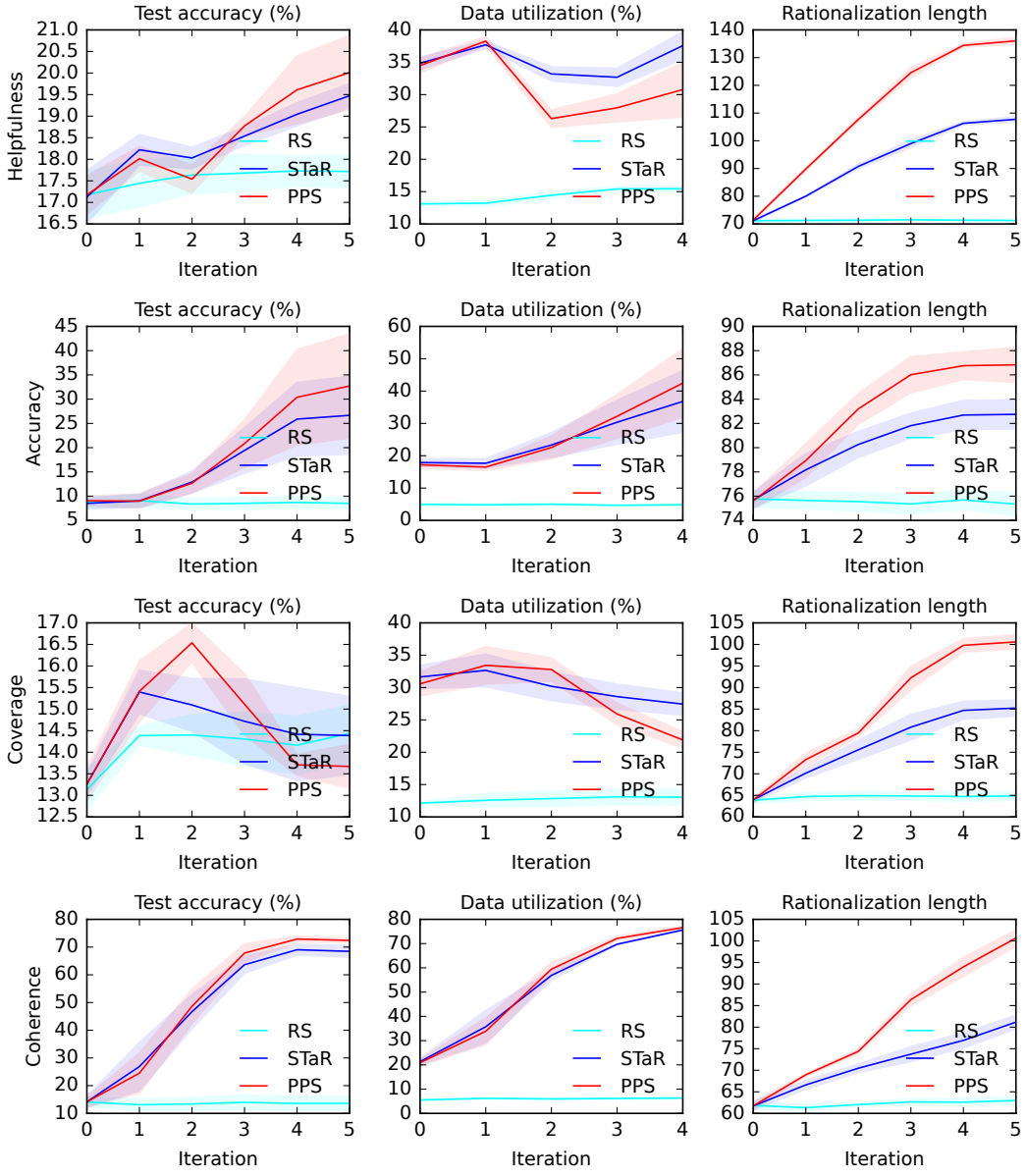

Figure 14: Test accuracy, data utilization, and rationalization length for Qwen 3B and Summarize from Feedback dataset.

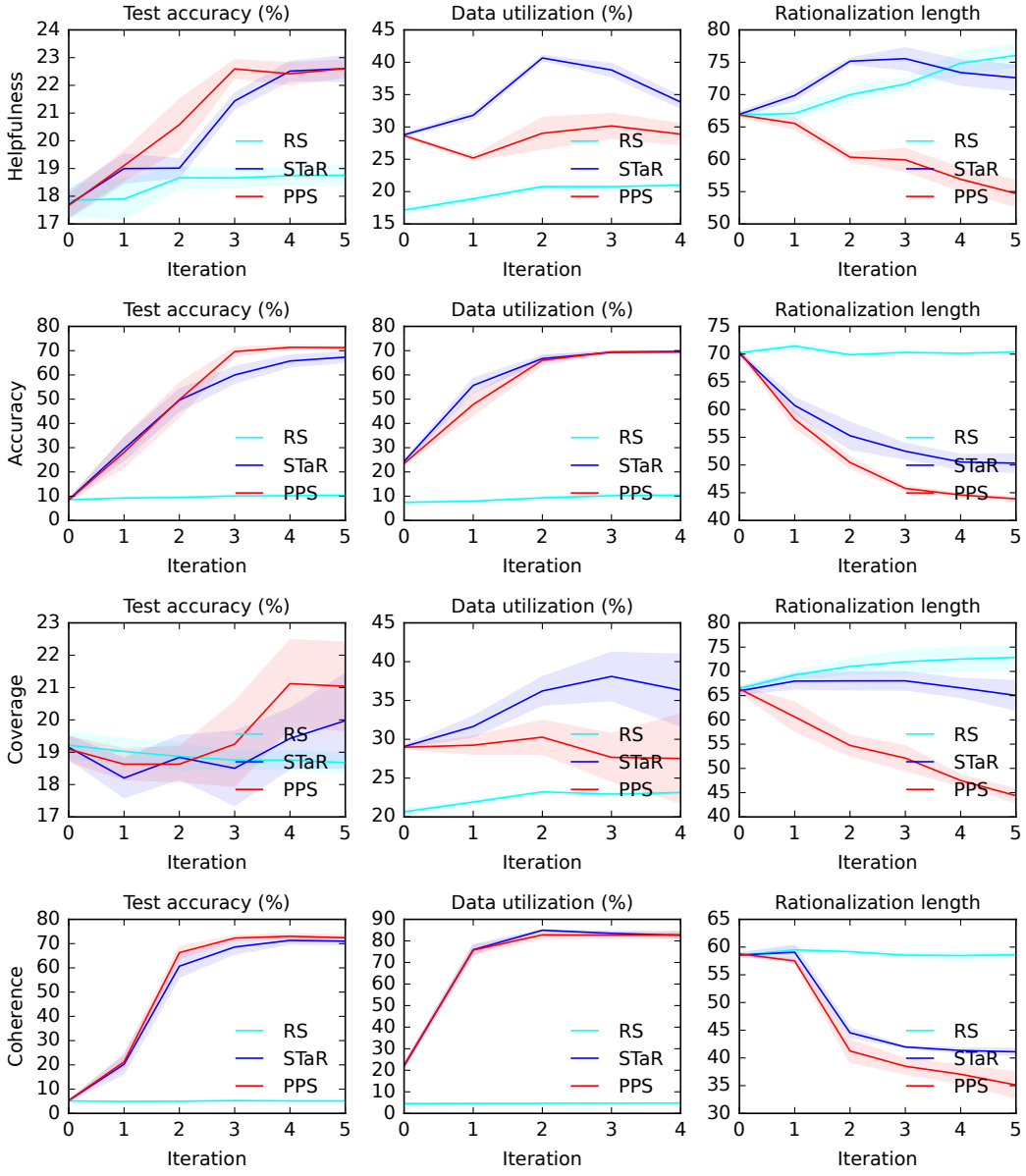

Figure 15: Test accuracy, data utilization, and rationalization length for Qwen 7B and Summarize from Feedback dataset.