F-TIS: Harnessing Diverse Models in Collaborative GRPO

Nikolay Blagoev [1 2] **Oğuzhan Ersoy** [1] **Wendelin Böhmer** [3] **Lydia Chen** [2]

Abstract

Reinforcement learning methods such as GRPO have seen great popularity in LLM post-training. In GRPO, models produce completions to a set of prompts, which are rewarded, and the policy is updated towards the relatively high reward completions. Due to the auto-regressive nature of models, the generation phase of such style of training can be extremely time consuming. As a solution, prior work has sought to distribute the inference step across many nodes, working parallel. These works assume primarily homogeneous models in the training in order to keep samples as close to on-policy as possible. This assumption may be impractical in decentralized systems, where parties with various computes and preferences may wish to collaborate on the same task. Thus, decentralized training requires an approach that can handle heterogeneous models - different models collaborating on the same tasks. However, this leads to highly off-policy samples presented during training, which prior work has identified that off-policy samples can hurt GRPO convergence. To enable heterogeneity, we propose Filtered Truncated Importance Sampling (F-TIS) - a GRPO-style training paradigm that can use off-policy samples to improve local model's learning. Our framework allows various models to collaborate in the same RL training run while being communication efficient. We extensively evaluate F-TIS in various heterogeneous setups and we show that it exhibits identical final model convergence to purely on-sample training. Furthermore, we observe in some setups better generalization on out-of-distribution tasks than on-policy training, increasing model's performance by up to 12%.

[1]Gensyn [2]University of Neuchâtel [3]TU Delft. Correspondence to: Oğuzhan Ersoy <oguzhan@gensyn.ai>.

Published as a workshop paper at SCALE - ICML 2026

1. Introduction

Reinforcement Learning (RL) has seen a great adoption for the post-training of Large Language Models (LLMs) (Shao et al., 2024; Rafailov et al., 2023; Xiao et al., 2025). Algorithms such as Proximal Policy Optimization (PPO) allow LLMs to learn user-preferred behaviour (Schulman et al., 2017), such as adhering to some ethical code during conversations. More recently, RL has been utilized to a great degree for the purposes of improving the reasoning of LLMs. This is in part due to the highly influential work of (Shao et al., 2024), which proposed Group Relative Policy Optimization (GRPO). GRPO removes the need for a value-model in PPO, thus greatly reducing the computational and memory requirements, by instead relying on the group relative advantage. In GRPO, for each prompt, multiple completions are generated and each completion's advantage is computed relative to the other completions of the same prompt. Several works have demonstrated the success of GRPO-based algorithms in improving an LLM's reasoning for various tasks (Xiao et al., 2025; Liu et al., 2025; DeepSeek-AI, 2025b).

Despite introducing lower memory footprint relative to PPO, GRPO comes with a high computational cost of generating multiple completions for each group (often 8 or more) (Wu et al., 2025). This results in a bottleneck during the generation step, as LLMs generate in an auto-regressive manner, i.e., one token at a time(Wang et al., 2025). A natural step is to distribute the generation step across many workers thus speeding up the generation (Wu et al., 2025; Team et al., 2025). Such systems, however, assume relatively homogeneous models and resources, making them impractical for decentralized training. Furthermore, even if participants start from the same models, if they solely communicate completions, their models will drift apart, due to floating-point non-associativity. Such drifts introduce off-policyness between the generator and the trainer, harming the final model's performance (Mroueh et al., 2025; Yao et al., 2025b).

Beyond these emergent drifts, we identify three types of model heterogeneity in decentralized RL post-training:

- **Model Size**: Models of various sizes (e.g. of 1 billion parameters and 3 billion parameters) can be trained

together in the same loop. More constrained devices train the smaller models, while larger devices or clusters can accommodate larger models. Alternatively, one can alleviate the costs of sampling from a large model (as the auto-regressive inference of LLMs is the biggest bottleneck in GRPO training (Wang et al., 2025)). Thus they can utilize some number of small models to populate the batch size.

- **Model Expertise**: Models of similar size, but different parameters and expertise can be trained together, as different users might have a personalized model they want to collaboratively improve on some task.
- **Trainable Parameters**: Some nodes may train a different subset of the same model's parameters (e.g. with parameter efficient fine-tuning). Such cases can arise due to resource constraints on some devices, needing a smaller set of trainable parameter, or to avoid destructive interference with other tasks (Nobari et al., 2025).

We empirically show that GRPO underperforms in such relative to training purely on-policy. To address this challenge, we propose F-TIS - a single unified framework that can make use of off-policy samples in decentralized heterogeneous Reinforcement Learning. This framework requires communication volume of only $8 \times |p|$ bytes, i.e., 8 bytes per each token, as it only communicates the log-probabilities and tokens of completions between nodes. Despite this minimal overhead, F-TIS can achieve performance identical to homogeneous RL and can even improve model's reasoning capabilities on out-of-distribution tasks (by up to 12%). Beyond its benefits in decentralized training, F-TIS provides benefits even for a single cluster-training, where multiple models can be post-trained in one training run, rather than several sequential ones.

2. Related Work

Reinforcement Learning has been commonly used for LLM post-training. Proximal Policy Optimization allowed models to learn user preference, which were implicitly hidden in data (Schulman et al., 2017). More recently, GRPO and its derivatives have been utilized to boost model's reasoning and instruction-following abilities. When a model θ is trained with GRPO, it generates a number G of completions (responses) a_i per prompt p ($p \circ a_i \ \forall i \in G$ where $\circ$ is a concatenation operation), which is termed a "group". Each of the completions is rewarded via some reward model, to get a single scalar value r_i. Commonly used reward mechanisms in GRPO are rule-based rewards (Liu et al., 2025), which check if the formatting and final answer is correct for a math task or all tests pass for a coding tasks. To replace the need for a value model, GRPO uses the group advantage $\hat{A}_i$, $\hat{A}_i := \frac{r_i - \mu_r}{\sigma_r}$, where μ and σ are the mean and standard deviation of the rewards for the completions belonging to the same prompt. The advantage is then used to compute the loss:

$$\mathcal{L}_{GRPO} = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|a_i|}\sum_{t=1}^{|a_i|} min[\frac{\pi_\theta(a_{i,t}|p \circ a_{i,<t})}{\pi_{\theta_{gen}}(a_{i,t}|p \circ a_{i,<t})}\hat{A}_i, clip(\frac{\pi_\theta(a_{i,t}|p \circ a_{i,<t})}{\pi_{\theta_{gen}}(a_{i,t}|p \circ a_{i,<t})}\hat{A}_i, 1-\epsilon, 1+\epsilon)]$$

where $\pi_{\theta_{gen}}$ is the policy that generated the completion.

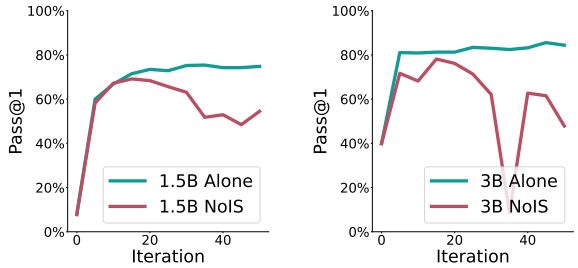
Training with GRPO can be divided into two phases - completion generation (gathering as many completions for various prompts) and training (computing the gradient based on the loss for all completions and for all prompts) (Wu et al., 2025). Prior work has identified that the KL term ($\mathcal{D}_{KL}(\pi_\theta \parallel \pi_{\theta_{ref}})$, does not benefit the training and increases memory and computational demands (Liu et al., 2025). In line with these works, we omit the KL-term in our experiments.

Distributed RL has gained popularity as a means of speeding up the generation phase of GRPO-style training (Wu et al., 2025). Deployed systems have even successfully trained models in a decentralized manner, however still assuming homogeneous models in order to prevent off-policy issues (Amico et al., 2025; Team et al., 2025). Such approaches have been classified into two categories: *vertical* and *horizontal* (Blagoev et al., 2025). In vertical RL, each device/node generates the entire group for one prompt. In horizontal RL, each device generates a subset of the group for all prompts (the same across devices). At the end of the generation phase, an all-gather is performed across devices, synchronizing the completions.

3. Decentralized Heterogeneous RL

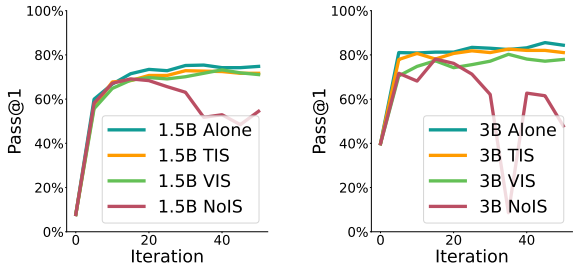
GRPO is fundamentally an on-policy algorithm (Yao et al., 2025a). While the clipped importance sampling provides some tolerance for "stale" or off-policy samples, prior work has identified that even small divergence due to difference in probabilities between the inference and training engine can lead to degradation in the RL training (Yao et al., 2025b). This acts as destructive noise that over time can collapse the policy.

We replicate these findings for decentralized RL with a motivational example of two models of different sizes (Qwen2.5-1.5B and Qwen2.5-3B (Qwen Team, 2024)) train collaboratively via vertical decentralized RL on the GSM8k dataset (Cobbe et al., 2021) (details of all hyperparameters can be found in Table 1). Here we employ the simplest approach - treat all samples as if on-policy, what we will term NoIS (No Importance Sampling) (Amico et al., 2025). We present the



(a) 1.5B model *(b)* 3B model

Figure 1. Validation curves of two models collaboratively trained via GRPO. NoIS presents heterogeneous training with no importance sampling used.



(a) 1.5B model *(b)* 3B model

Figure 2. Comparison of various importance sampling methods - NoIS (No Importance Sampling), VIS (Vanilla Importance Sampling), and TIS (Truncated Importance Sampling).

validation curves in Figure 1, where we show that both models have significantly worse performance than if they were to train alone, making collaborative training unattractive for different users.

3.1. Importance Sampling

Prior work has proposed two modifications to the GRPO loss to account for off-policy examples. The first we term VIS (Vanilla Importance Sampling), which communicates each time the $\pi_{\theta_{gen}}(a_{i,t}|p \circ a_{i,<t})$ for every generation (Wu et al., 2025; Shao et al., 2024). A different approach, proposed by (Yao et al., 2025b), which they term Truncated Importance Sampling (TIS), takes the importance sampling term outside the inner-most sum's term:

$$\mathcal{L}_{GRPO} = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|a_i|}\sum_{t=1}^{|a_i|} min\left(\frac{\pi_\theta(a_{i,t}|p \circ a_{i,<t})}{\pi_{\theta_{gen}}(a_{i,t}|p \circ a_{i,<t})}, C\right) min\left(\mathcal{R}_{i,\theta}\hat{A}_i, clip(\mathcal{R}_{i,\theta}\hat{A}_i, 1-\epsilon, 1+\epsilon)\right)$$

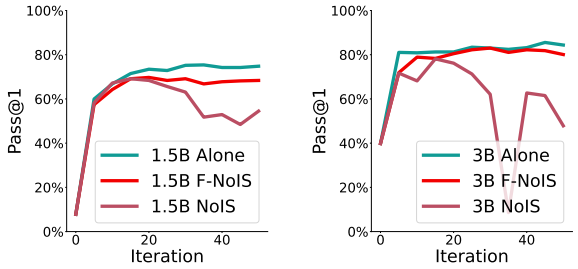
for some constant C, where $\mathcal{R}_{i,\theta} = \frac{\pi_\theta(a_{i,t}|p \circ a_{i,<t})}{\pi_{\theta_{detach}}(a_{i,t}|p \circ a_{i,<t})}$.

We compare the three methods (NoIS, VIS, and TIS) in Figure 2, with $C = 2$. While for the smaller model, TIS and VIS behave almost identically, for the larger model TIS is clearly superior than VIS. This is somewhat expected as previous work has suggested that TIS offers better performance than VIS (Yao et al., 2025b).

These approaches require the additional communication only of per-token log-probabilities, which are small in size (4 bytes per token), thus satisfying our requirement for introducing minimal communication overhead.

3.2. Filtering Samples

Another line of work has explored filtering low-quality off-policy samples as an approach of dealing with off-policy samples (DeepSeek-AI, 2025a; Blagoev et al., 2025). During the update phase, samples with advantage less than 0, $\hat{A}_i < 0$, and with KL-divergence beyond some threshold g, $\mathcal{D}_{KL}(\pi_\theta \parallel \pi_{\theta_{gen}}) > g$, are used only to compute the group advantage, but are filtered out before the update phase. The intuition is that samples with $\hat{A}_i > 0$ provide a direction for the policy to move towards, even if off-policy. Off-policy samples with $\hat{A}_i < 0$ do not - they end up amplifying less probable tokens, which the model would not have produced, resulting often times in gibberish completions. We evaluate a filtered version of NoIS, F-NoIS, with $g = 50$. The validation curves in Figure 3 demonstrate a clear improvement over its non-filtered counterpart, stabilizing the training and preventing model collapse. Even just filtering provides performance close to that of the baseline.



(a) 1.5B model *(b)* 3B model

Figure 3. The effect of filtering in heterogeneous RL. F-NoIS presents a filtered version of NoIS.

3.3. F-TIS

Our approach combines the performance of TIS with the stability of filtering into Filtered Truncated Importance Sampling (F-TIS). The modified GRPO formula can thus be

written as:

$$\mathcal{L}_{GRPO} = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|a_i|}\sum_{t=1}^{|a_i|} min\left(\frac{\pi_\theta(a_{i,t}|p \circ a_{i,<t})}{\pi_{\theta_{gen}}(a_{i,t}|p \circ a_{i,<t})}, C\right)$$
$$min\left(\mathcal{R}_{i,\theta}\hat{A}_{t,i}, clip(\mathcal{R}_{i,\theta}\hat{A}_{t,i}, 1-\epsilon, 1+\epsilon)\right)$$
$$\hat{A}_{t,i} = \begin{cases} \hat{A}_i & \text{if } \hat{A}_i > 0 \text{ or } \mathcal{D}_{KL}(\pi_\theta \parallel \pi_{\theta_{gen}}) < g \\ 0 & \text{else} \end{cases}$$

4. Results

Throughout these experiments, unless otherwise specified, we train two models via vertical decentralized RL, with a group size of 12 and batch size of 24. We use $g = 50$ and $C = 2$. All models are trained on the GSM8k (Cobbe et al., 2021) dataset for 50 iterations and evaluated on a held-out validation set via greedy-decoding ($pass@1$).

We further test the final models' performance on an out-of-distribution dataset - MATH-500 (Hugging Face H4, 2025). Additional information on the hyperparameters can be found in Table 1. We use a binary reward function (Liu et al., 2025), which is 1 if and only if the formatting and final answer are correct, with 0 otherwise. Details on the system prompt used can be found below.

System prompt for experiments

```
A conversation between User and
Assistant.  The user asks a question,
and the Assistant solves it.
The assistant needs to provide a
detailed step by step solution of the
problem.  The reasoning process is
enclosed within <think> </think> and
the answer within <answer> </answer<
tags, i.e., <think> reasoning process
here </think>
<answer> answer here </answer>
```

4.1. Model Size Heterogeneity

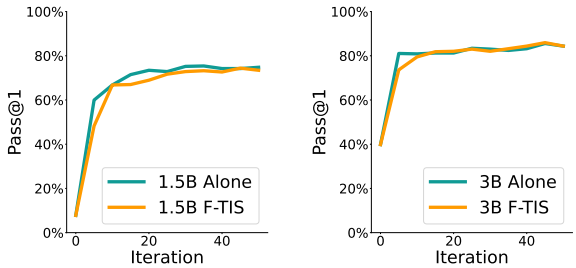
We perform two sets of experiments in vertical decentralized RL with F-TIS: collaborative training of Qwen2.5-1.5B and Qwen2.5-3B models, and collaborative training of Qwen2.5-Coder-1.5B and Qwen2.5-Coder-3B models. The validation curves during training are reported in Figures 4 and 5. We observe near identical final performance of collaborative training with F-TIS to the baseline. However, across all experiments we observe an initially slower convergence relative to the baseline, which we discuss further in Section 4.5. We report the performance on the MATH-500 dataset of the models in Table 2, where we observe that on out-of-distribution tasks the smaller models display a significant improvement in performance.

Parameter	Value
learning rate	1×10^{-6}
group size	12
batch size	16
ϵ	0.2
C	2
g	50

Table 1. Hyperparameters for the experiments.

	Model	Math-500
Alone	1.5B Base	0.406
	3B Base	0.575
	1.5B Coder	0.41
	3B Coder	0.478
	1.5B PEFT	0.412
	3B PEFT	0.5
F-TIS	1.5B Base	0.47
	3B Base	0.54
F-TIS	1.5B Coder	0.47
	3B Coder	0.59
F-TIS	1.5B Base	0.403
	1.5B Coder	0.41
F-TIS	3B Base	0.52
	3B Coder	0.53
F-TIS	1.5B Base	0.43
	1.5B PEFT	0.43
F-TIS	3B Base	0.513
	3B PEFT	0.56

Table 2. Final model evaluation on the Math-500 dataset.

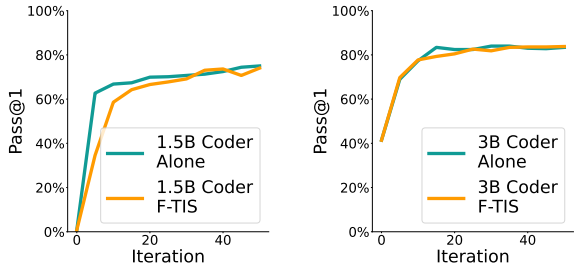


(a) 1.5B model *(b)* 3B model

Figure 4. Validation curves of a Qwen2.5-1.5B and a Qwen2.5-3B trained together.

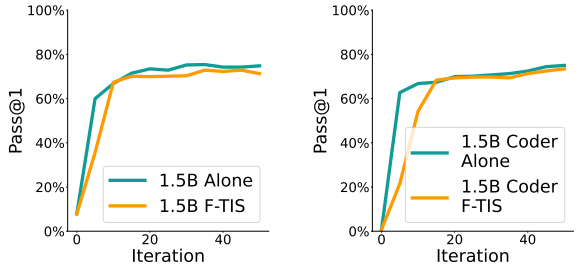
4.2. Model Expertise Heterogeneity

We further test F-TIS in collaborative setups, where model size is the same, but models may have different expertise. We study two setups - collaborative training of Qwen2.5-1.5B and Qwen2.5-Coder-1.5B models, and of Qwen2.5-3B and Qwen2.5-Coder-3B models. The validation curves are reported in Figures 6 and 7.



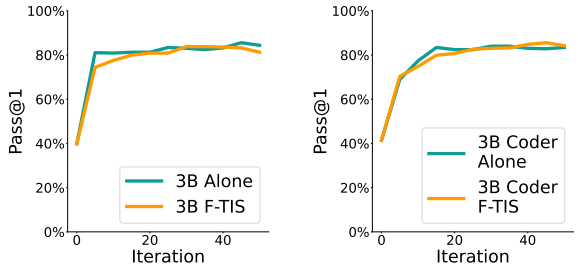
(a) 1.5B Coder model *(b)* 3B Coder model

Figure 5. Validation curves of a Qwen2.5-Coder-1.5B and a Qwen2.5-Coder-3B trained together.



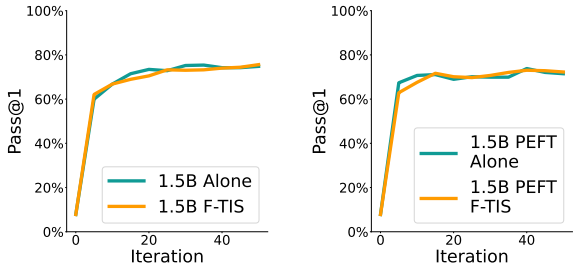
(a) 1.5B model *(b)* 1.5B Coder model

Figure 6. Validation curves of a Qwen2.5-1.5B and a Qwen2.5-Coder-1.5B trained together.



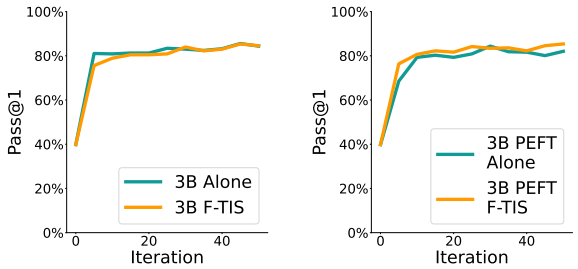
(a) 3B model *(b)* 3B Coder model

Figure 7. Validation curves of a Qwen2.5-3B and a Qwen2.5-Coder-3B trained together.



(a) 1.5B Model *(b)* 1.5B PEFT Model

Figure 8. Validation curves of a Qwen2.5-1.5B and a Qwen2.5-Coder-1.5B with LORA trained together.



(a) 3B model *(b)* 3B PEFT model

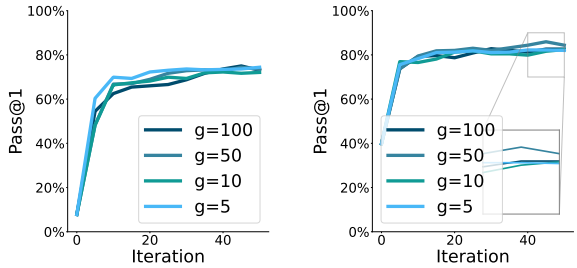
Figure 9. Validation curves of a Qwen2.5-3B and a Qwen2.5-3B with LORA trained together

4.3. Trainable Parameters Heterogeneity

To test the final model heterogeneity scenario, we perform collaborative GRPO with F-TIS on: Qwen-2.5-1.5B with and without LORA, and for Qwen-2.5-3B with and without LORA (Hu et al., 2021). We use LORA with dimension of 128, targeting the Query and Value matrices. We present the validation curves in Figures 8 and 9. Interestingly, we observe much better convergence of the 3B model with LORA when trained collaboratively with the base model. This is further collaborated by the performance on out-of-distribution tasks in Table 2. This suggests that GRPO training with PEFT can be improved by using off-policy samples of a non-PEFT model.

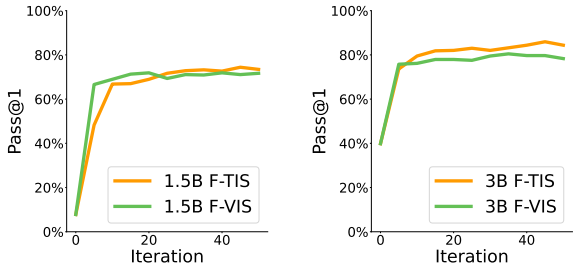
4.4. Out-of-distribution Math reasoning

We evaluate all trained models on Math reasoning on the MATH-500 dataset (Hugging Face H4, 2025), using the same system prompt and reward as before. This constitutes and out-of-distribution test - data that, albeit similar, has very different distribution. We present the results in the third column of Table 2, where models are grouped together with the respective model they were trained collaboratively with. Throughout all collaborations we observe a general trend where the model which performed worse off in the alone baseline can improve through the joint training. Most noticeably, the 3B PEFT model improves its performance by 7%. However, the model which performed better in the baseline case, tends to perform worse off in the collaborative setting on out-of-distribution evaluation. A notable exception is the 3B Coder model, which when paired with a Base model improves its performance by 5.2% and when paired with a smaller Coder model - by 12%. We attribute this to a potential over-fitting of the larger Coder model to coding tasks, reducing their reasoning ability in Math tasks.



(a) 1.5B model *(b)* 3B model

Figure 10. Ablation on the choice of g for F-TIS.



(a) 1.5B model *(b)* 3B model

Figure 11. Comparison between F-VIS and F-TIS.

4.5. Ablation on filtering

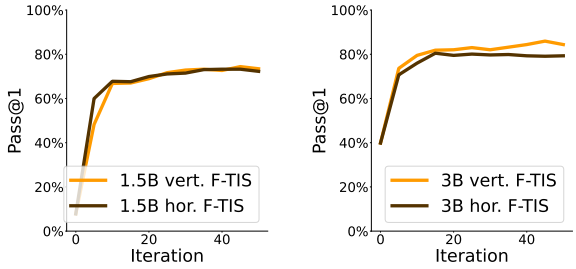
Prior work has not studied suitable values for g in GRPO-training. Throughout these experiments we have solely used $g = 50$, as empirically we found it to be the best performant constant. In this section, we perform ablations to justify our choice of the value for this hyperparameter, repeating the experiments of Section 4.1 with g of 5, 10, 50, and 100. We report the validation curves in Figure 10. Interestingly, we observe that for the 1.5B, especially initially, using a small g provides the greatest improvement in results. However, for the 3B model, using slightly higher $g = 50$ provides better performance. We attribute this to the fact that initially, models may produce really random outputs, while still learning the task. Such outputs provide the 1.5B with no good signal on how to improve, thus stalling its training. However, especially later in the training, the larger model benefits from the low-reward off-policy completions as it allows for greater exploration of the space.

4.6. F-TIS vs F-VIS

In our previous experiments we employed Truncated Importance Sampling as the base of our approach. Here we demonstrate that, even with filtering, Vanilla Importance Sampling underperforms compared to F-TIS. To this end we repeat the experiments of the size heterogeneity tests with F-VIS with $g = 50$ for both F-TIS and F-VIS. We report the validation curves of both approaches in Figure 11a for the 1.5B model and Figure 11b for the 3B model. We note the faster earlier convergence for F-VIS (similar to lower g in Figure 10), however failing to generalize in later iterations, unlike F-TIS.

4.7. Horizontal Collaboration

So far we have solely focused on vertical collaboration - one node responsible for all completions per question (Blagoev et al., 2025). This has the benefit of faster generation (as nodes can parallelize computation independently for some subset of questions), requires less synchronization,



(a) 1.5B model *(b)* 3B model

Figure 12. Comparison between horizontal and vertical F-TIS.

and maintains the advantage calculation only with respect to one model's performance. We hypothesize that the last of these listed benefits would make horizontal learning impractical, as now the advantage calculation is performed relative to the swarm's mean performance. This could introduce an unwanted bias in the computation of the policy's gradient. We verify this hypothesis by repeating the experiments for F-TIS in Section 4.1, however with horizontal collaboration. We present the results in Figure 12. As expected, horizontal learning leads to a noticeable degradation in the model's performance, though mainly noticeable for the 3B model.

5. Conclusion

In this paper, we presented F-TIS - a novel approach to collaborative training across different models in GRPO-style setups. We extensively evaluated our method in different heterogeneous settings, where model size, expertize, or trainable parameters can differ. For all, we observe similar convergence of F-TIS to that of fully on-policy learning. For some cases we even observe noticeable improvement of the final model on both in- and out-of-distribution tasks. We posit this work as one of the first works to study collaborative RL for LLMs and we hope it inspires further research into this new area.

Impact Statement

This paper presents a work whose goal the goal of which is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Amico, J., Andrade, G. P., Donaghy, J., Fielding, B., Forbus, T., Grieve, H., Kara, S., Kolehmainen, J., Lou, Y., Nies, C., Nuño, E. P. F., Ortega, D., Rastogi, S., Virts, A., and Wright, M. J. Sharing is caring: Efficient lm post-training with collective rl experience sharing, 2025. URL https://arxiv.org/abs/2509.08721.

Blagoev, N., Ersoy, O., and Chen, L. Y. Hail to the thief: Exploring attacks and defenses in decentralised GRPO. *CoRR*, abs/2511.09780, 2025. doi: 10.48550/ARXIV.2511.09780. URL https://doi.org/10.48550/arXiv.2511.09780.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Tworek, J., Such, F. P., Krueger, G., Chan, V., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models. *CoRR*, abs/2512.02556, 2025a. doi: 10.48550/ARXIV.2512.02556. URL https://doi.org/10.48550/arXiv.2512.02556.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025b. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Hugging Face H4. Math-500. https://huggingface.co/datasets/HuggingFaceH4/MATH-500, 2025. Hugging Face dataset, 500 math problems for evaluation; accessed 2026-04-22.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding r1-zero-like training: A critical perspective. *CoRR*, abs/2503.20783, 2025. doi: 10.48550/ARXIV.2503.20783. URL https://doi.org/10.48550/arXiv.2503.20783.

Mroueh, Y., Dupuis, N., Belgodere, B., Nitsure, A., Rigotti, M., Greenewald, K. H., Navrátil, J., Ross, J., and Rios, J. Revisiting group relative policy optimization: Insights into on-policy and off-policy training. *CoRR*, abs/2505.22257, 2025. doi: 10.48550/ARXIV.2505.22257. URL https://doi.org/10.48550/arXiv.2505.22257.

Nobari, A. H., Alimohammadi, K., ArjomandBigdeli, A., Srivastava, A., Ahmed, F., and Azizan, N. Activation-informed merging of large language models. *CoRR*, abs/2502.02421, 2025. doi: 10.48550/ARXIV.2502.02421. URL https://doi.org/10.48550/arXiv.2502.02421.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Team, P. I., Jaghouar, S., Mattern, J., Ong, J. M., Straube, J., Basra, M., Pazdera, A., Thaman, K., Ferrante, M. D., Gabriel, F., Obeid, F., Erdem, K., Keiblinger, M., and Hagemann, J. INTELLECT-2: A reasoning model trained through globally decentralized reinforcement learning. *CoRR*, abs/2505.07291, 2025. doi: 10.48550/ARXIV.2505.07291. URL https://doi.org/10.48550/arXiv.2505.07291.

Wang, L., Xie, H., Wang, X., Huang, T., Li, M., and Wang, D. Infinite sampling: Efficient and stable grouped RL training for large language models. *CoRR*, abs/2506.22950, 2025. doi: 10.48550/ARXIV.2506.22950. URL https://doi.org/10.48550/arXiv.2506.22950.

Wu, B., Wang, S., Tang, Y., Ding, J., Helenowski, E., Tan, L., Xu, T., Gowda, T., Chen, Z., Zhu, C., Tang, X., Qian, Y., Zhu, B., and Hou, R. Llamarl: A distributed

asynchronous reinforcement learning framework for efficient large-scale LLM training. *CoRR*, abs/2505.24034, 2025. doi: 10.48550/ARXIV.2505.24034. URL https://doi.org/10.48550/arXiv.2505.24034.

Xiao, Y., Sun, E., Chen, T., Wu, F., Luo, D., and Wang, W. Trading-r1: Financial trading with LLM reasoning via reinforcement learning. *CoRR*, abs/2509.11420, 2025. doi: 10.48550/ARXIV.2509.11420. URL https://doi.org/10.48550/arXiv.2509.11420.

Yao, C., Chen, Y., Sun, Y., Chen, Y., Zhang, W., Pan, X., Li, Y., and Ding, B. Group-relative REINFORCE is secretly an off-policy algorithm: Demystifying some myths about GRPO and its friends. *CoRR*, abs/2509.24203, 2025a. doi: 10.48550/ARXIV.2509.24203. URL https://doi.org/10.48550/arXiv.2509.24203.

Yao, F., Liu, L., Zhang, D., Dong, C., Shang, J., and Gao, J. Your efficient rl framework secretly brings you off-policy rl training. https://fengyao.notion.site/off-policy-rl, August 2025b. First published August 5, 2025; last updated October 13, 2025.