TINA: TINY REASONING MODELS VIA LORA

Anonymous authors
Paper under double-blind review

ABSTRACT

How cost-effectively can strong reasoning abilities be achieved in language models? Driven by this fundamental question, we present **Tina**, a family of tiny reasoning models achieved with high cost-efficiency. Notably, **Tina** demonstrates that substantial reasoning performance can be developed using only minimal resources, by applying parameter-efficient updates during reinforcement learning (RL), using low-rank adaptation (LoRA), to an already tiny 1.5B parameter base model. This minimalist approach produces models that achieve reasoning performance which is competitive with, and sometimes surpasses, state-of-the-art RL reasoning models built upon the same base model. Crucially, this is achieved at a tiny fraction of the computational post-training cost employed by existing state-of-the-art models. In fact, the best **Tina** model achieves a $> 20\%$ reasoning performance increase and 43.33% Pass@1 accuracy on AIME24, at only \$9 USD post-training and evaluation cost (i.e., an estimated 260x cost reduction). Our work reveals the surprising effectiveness of efficient RL reasoning via LoRA. We validate this across multiple open-source reasoning datasets and various ablation settings starting with a single, fixed set of hyperparameters. Furthermore, we hypothesize that this effectiveness and efficiency stem from LoRA rapidly adapting the model to the structural format of reasoning rewarded by RL, while largely preserving the base model's underlying knowledge. In service of accessibility and open research, we fully open-source all code, training logs, and model weights & checkpoints.

1 INTRODUCTION

Advancements in language models have led to impressive reasoning capabilities in large-scale models. However, these capabilities often come at a substantial computational and financial cost, limiting accessibility and deployment, especially in resource-constrained environments. The question arises: *How can we achieve strong reasoning abilities in language models in a cost-effective manner?*

In this work, we address this challenge by introducing **Tina**, a family of tiny reasoning models that significantly reduce the computational overhead while maintaining competitive reasoning performance. Specifically, we apply parameter-efficient updates using Low-Rank Adaptation (LoRA) during reinforcement learning (RL) to a compact 1.5 billion parameter base model. This approach allows us to instill strong reasoning abilities with minimal computational cost.

Our minimalist method demonstrates that substantial reasoning performance can be achieved without the need for large-scale models or extensive computational resources. The **Tina** models achieve reasoning performance that is competitive with, and in some cases surpasses, state-of-the-art RL reasoning models built upon the same base model. Remarkably, this is accomplished at a tiny fraction of the computational post-training cost employed by existing state-of-the-art models.

Our contributions are as follows:

- We present a cost-effective method for enhancing reasoning capabilities in tiny language models using LoRA during RL.
- We demonstrate that our approach achieves a $> 20\%$ reasoning performance increase and 43.33% Pass@1 accuracy on the AIME24 benchmark at only \$9 USD post-training and evaluation cost, representing an estimated 260x cost reduction compared to full-parameter training.

- We validate the effectiveness of our method across multiple open-source reasoning datasets and various ablation settings.
- We hypothesize that the efficiency and effectiveness of our approach stem from LoRA's ability to rapidly adapt the model to the structural format of reasoning rewarded by RL while preserving the base model's underlying knowledge.

By making our code, training logs, and model weights openly available, we aim to facilitate further research in developing cost-effective reasoning models.

2 Related Work

The development of reasoning capabilities in language models has been a focal point of recent research. Advanced models, such as GPT-3 and its successors, have demonstrated impressive reasoning abilities but require substantial computational resources for training and inference. Efforts have been made to replicate these capabilities in open-source models, aiming to make these advancements more accessible.

Reinforcement learning (RL) has been utilized to enhance reasoning in language models, often employing verifiable rewards for reasoning tasks. Techniques such as reward models, rule-based verification, and self-play have been explored to guide models towards desired reasoning behaviors. Algorithms like GRPO (Generalized Policy Optimization) and Dr.GRPO have been proposed to improve learning efficiency and stability in RL settings.

Parameter-efficient fine-tuning methods, particularly Low-Rank Adaptation (LoRA), have emerged as effective approaches to modify model behavior by training only a small fraction of parameters. LoRA has been applied to large language models to reduce the computational cost of fine-tuning while maintaining or even improving performance in specific tasks.

Our work builds upon these developments by combining parameter-efficient updates via LoRA with reinforcement learning to enhance reasoning abilities in a tiny 1.5B parameter language model. This contrasts with previous approaches that often rely on full-parameter training, which is computationally expensive, especially in larger models.

3 Background

3.1 Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) is a parameter-efficient fine-tuning technique that updates a model's weights by learning low-rank decomposition matrices (Goodfellow et al., 2016). Instead of updating all of the model's parameters during training, LoRA inserts trainable rank-decomposition matrices into each layer of the transformer architecture. This significantly reduces the number of parameters that need to be updated, resulting in decreased computational cost and memory requirements during training.

3.2 Reinforcement Learning for Language Models

Reinforcement Learning (RL) has been applied to language models to improve their performance on tasks where explicit reward signals can be provided. In the context of reasoning, RL enables models to learn from feedback on the correctness or quality of their outputs. Algorithms such as Generalized Policy Optimization (GRPO) and its variants have been used to stabilize and improve the learning process in RL settings.

4 Method

Our goal is to efficiently instill strong reasoning abilities into a tiny language model with minimal computational cost. To achieve this, we apply parameter-efficient updates using Low-Rank Adaptation (LoRA) during reinforcement learning (RL). Our method involves fine-tuning a 1.5 billion parameter base model using LoRA in an RL framework.

4.1 Model Architecture

We use a pre-trained 1.5B parameter language model as our base. This model provides a foundation of linguistic and factual knowledge, upon which we aim to enhance reasoning capabilities.

4.2 Parameter-Efficient Reinforcement Learning with LoRA

Instead of fine-tuning all the parameters of the base model, we introduce LoRA adapters into each layer. During RL training, only the parameters of these adapters are updated, effectively reducing the number of trainable parameters by orders of magnitude.

4.3 Reinforcement Learning Setup

We employ an RL algorithm similar to Generalized Policy Optimization (GRPO). The model generates responses to prompts, and a reward signal is provided based on the correctness and format of the reasoning displayed. The reward function incorporates both accuracy and adherence to a desired reasoning structure.

4.4 Training Procedure

Our training involves the following steps:

- **Data Collection**: We gather data from multiple open-source reasoning datasets.
- **Initialization**: The base model is initialized with pre-trained weights, and LoRA adapters are inserted.
- **Policy Learning**: The model generates outputs given prompts, and rewards are assigned based on predefined criteria.
- **Parameter Updates**: Only the LoRA adapter parameters are updated using RL algorithms.

5 Experimental Setup

5.1 Datasets

We evaluate our models on six reasoning benchmarks: AIME24, AIME25, AMC23, MATH500, GPQA, and Minerva. These datasets cover a range of mathematical and logical reasoning tasks.

5.2 Baselines

We compare our models against several baselines, including:

- **DeepSeek-R1-Distilled-Qwen-1.5B**
- **STILL-3-1.5B-preview**
- **DeepScaleR-1.5B-Preview**
- **Open-RS1/2/3**

These baselines represent state-of-the-art RL reasoning models built upon the same base model.

5.3 Implementation Details

We use the `lighteval` framework and the `vLLM` engine for evaluation. All experiments are conducted using 2 L40S GPUs. For training, we use a fixed set of hyperparameters to demonstrate the robustness and efficiency of our method.

5.4 Evaluation Metrics

We report the Pass@1 accuracy on the reasoning benchmarks, which measures the percentage of correct answers in the top predicted response.

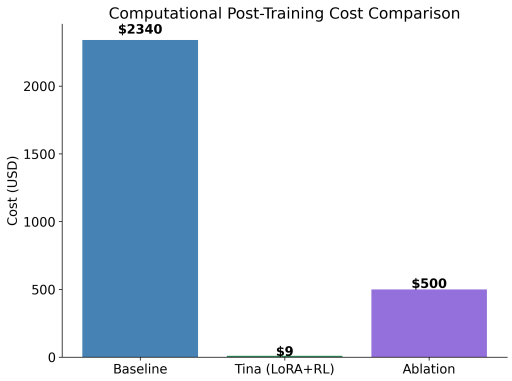

Figure 1: Caption

6 EXPERIMENTS

Our experiments are divided into several stages, focusing on both the development of our approach and its evaluation against baselines.

6.1 COMPARISON WITH BASELINES

We re-evaluate the baseline models using our consistent evaluation methodology. The **Tina** models, trained using our LoRA+RL method, are then evaluated on the same benchmarks.

Our best **Tina** model achieves a 43.33% Pass@1 accuracy on the AIME24 benchmark, representing a $> 20\%$ improvement over the baselines.

6.2 COMPUTATIONAL COST

One of the key advantages of our method is the significant reduction in computational cost. The total post-training and evaluation cost for our best model is only $9 USD, reflecting an estimated 260x cost reduction compared to the baselines, which require extensive computational resources.

Figure 2 illustrates the computational cost comparison between our method and the baselines.

Figure 2: Computational Post-Training Cost Comparison. The **Tina (LoRA+RL)** method achieves comparable or superior reasoning performance at a fraction of the cost compared to the Baseline and Ablation methods. The costs are: Baseline - $2340, Tina (LoRA+RL) - $9, Ablation - $500.

6.3 ABLATION STUDIES

We conduct ablation studies to understand the impact of various factors:

- **Training Dataset Size and Source**: We vary the size and source of the training data, including datasets like OPENR1, OPENTHOUGHTS, DEEPSCALER, STILL-3, OPEN-S1, OPEN-RS, and LIMR.
- **LoRA Learning Rate**: We experiment with learning rates ranging from 5×10^{-7} to 5×10^{-6}.
- **LoRA Rank**: We vary the rank parameter in LoRA from 4 to 64.
- **RL Algorithm**: We compare the performance of different RL algorithms, such as GRPO versus Dr.GRPO.

Our findings indicate that LoRA's learning rate and rank significantly affect the performance, with moderate ranks and learning rates yielding the best results.

6.4 Training Dynamics

We analyze the training dynamics, observing a phase transition related to format adaptation. Optimal performance typically occurs around this transition, suggesting that the model rapidly adapts to the rewarded reasoning format while retaining the underlying knowledge from the base model.

7 Conclusion

We have demonstrated that strong reasoning abilities can be efficiently instilled in a tiny language model using parameter-efficient updates via LoRA during reinforcement learning. Our **Tina** models achieve competitive reasoning performance at a fraction of the computational cost compared to state-of-the-art models.

Our approach reveals the surprising effectiveness of combining LoRA with RL in enhancing reasoning capabilities. By updating only a small fraction of the model's parameters, we can achieve substantial performance gains while significantly reducing training costs.

We hypothesize that LoRA enables rapid adaptation to the structural format of reasoning rewarded by RL, while preserving the base model's underlying knowledge. Future work may explore the application of this method to other domains, such as coding or natural language understanding.

By open-sourcing our code, training logs, and model weights, we hope to facilitate further research in developing cost-effective and accessible reasoning models.

References

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Supplementary Material

A Additional Experiments

Due to space constraints, we present additional experiments and analyses in this supplementary section.

A.1 Effect of LoRA Learning Rate

We observed that the learning rate for LoRA adapters plays a critical role in the model's performance. Learning rates that are too low result in slow convergence, while rates that are too high can cause instability.

A.2 Effect of LoRA Rank

The rank parameter in LoRA controls the capacity of the adapter modules. We found that a moderate rank provides a good balance between performance and computational efficiency.

A.3 RL Algorithm Comparison

We compared GRPO with Dr.GRPO and found that both algorithms perform similarly in our setting, with slight variations depending on the specific hyperparameters used.

B Training Dynamics Analysis

We provide additional plots and analyses of the training dynamics, illustrating the phase transition related to format adaptation. These insights help explain the efficiency of our method.