Towards Robust Reinforcement Learning for Small-Scale Language Model Agents

Md Rezwanul Haque[1], Md. Milon Islam[2], and Fakhri Karray[1,3]

***Abstract*—The alignment of Small Language Models (SLMs) in the 70–500M parameter range using reinforcement learning is often considered unstable, though the underlying failure mechanisms have not been systematically investigated. In the State-of-the-Art (SOTA) research, fifteen (model, corpus) configurations were trained using Proximal Policy Optimization (PPO). The experiments included Pythia-70M, 160M, 410M and SmolLM2-135M, 360M on the TinyStories, CNN/DailyMail, and Wikitext-103 corpora. Three reproducible failure modes were identified in small-scale language models: silent LoRA parameter freezing in standard PEFT/TRL pipelines, numerical overflow in importance ratios when using bfloat16, and catastrophic policy collapse due to reward-model error. These issues were addressed using a merge-and-reinitialize adapter technique, float32 precision during PPO updates, and a three-layer safety mechanism comprising reward whitening, importance-ratio guarding, and weight rollback. In this paper, a capacity-headroom hypothesis is proposed, which states that PPO performance at the SLM scale depends on both a fluent supervised model (PPL < 20) and a discriminative reward signal, rather than on the number of model parameters. The proposed system converged stably in all experiments and improved preference win rate over the SFT baseline in configurations with a fluent prior and an informative reward signal. Furthermore, it outperformed instruction-tuned baselines while requiring significantly less training data. All checkpoints, preference datasets, and training scripts are publicly released[§].**

***Index Terms*—Model Alignment, Reinforcement Learning from Human Feedback, Small Language Models, Proximal Policy Optimization, Low-Rank Adaptation, On-Device Agents.**

I. Introduction

Small Language Models (SLMs) in the 70–500M parameter range are suitable for resource-constrained, privacy-preserving, and on-device agents in human–machine systems. Unlike billion-parameter models, SLMs can perform real-time inference on commonly available hardware, facilitating deployment in agentic systems for tasks such as summarization, controlled generation, and dialogue at the edge. The alignment of SLM agents via reinforcement learning is important for both machine learning and systems engineering, as the PPO control loop must remain stable at scales where the learning signal is weak and the policy capacity is limited. Models with more than 100 billion parameters have achieved human-level performance across various benchmarks [1]. However, high training and deployment costs make them difficult to use in academic laboratories and edge-deployed applications. Data curation and architectural improvements have reduced the performance gap between SLMs and larger models [2]–[6], making models with fewer than 500 million parameters a practical choice for on-device inference.

This simplified formulation provides a stability foundation for more advanced applications involving tool use and multi-turn interactions. However, the alignment of SLMs remains an open research problem. Reinforcement Learning from Human Feedback (RLHF) [7], [8] is the standard method for aligning large-scale models, though application of the models with fewer than 500 million parameters remains limited. PPO [9] is commonly considered unstable at this model scale, often leading to exploding gradients and reward collapse. Therefore, Supervised Fine-Tuning (SFT) or Direct Preference Optimization (DPO) [10] is often used instead, without investigating the underlying causes of PPO instability. This assumption is used to determine whether classical PPO can effectively support small language model agents and identify the technical requirements for stable training. The agent is represented as a parameterized policy π_θ [9] that selects actions from a discrete action space and receives scalar reward from the environment. The fifteen configurations considered in this research use a single-turn form of the resulting Markov Decision Process (MDP). In this setting, the action space is defined by the vocabulary $\mathcal{V}$, the horizon T consists of a single turn ending with the end-of-sequence token, and the reward is provided at the terminal state. The empirical evaluation is restricted to the single-turn case, while the framework for multi-turn interactions is released with this paper.

This question is addressed through an empirical study using an end-to-end system. Five base models (Pythia-70M, 160M, 410M and SmolLM2-135M, 360M) [3], [4] are trained on three distinct datasets (TinyStories [2], CNN/DailyMail [11], and Wikitext-103 [12]) using a full SFT $\rightarrow$ reward-model $\rightarrow$ PPO cycle. This process generates fifteen fully trained configurations. All hyperparameters are kept the same across all experiments to investigate the effects of model capacity and domain difficulty. This approach allows us to distinguish configuration-specific issues from general structural failure modes.

[1]The authors are with the Centre for Pattern Analysis and Machine Intelligence, Department of Electrical and Computer Engineering, University of Waterloo, N2L 3G1, Ontario, Canada. (e-mail: rezwan@uwaterloo.ca*).

[2]The author is with the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh. (e-mail: milonislam@cse.kuet.ac.bd, milonislam@uwaterloo.ca).

[1,3]The author is with the Centre for Pattern Analysis and Machine Intelligence, Department of Electrical and Computer Engineering, University of Waterloo, N2L 3G1, Ontario, Canada, and Department of Machine Learning, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates. (e-mail: karray@uwaterloo.ca, fakhri.karray@mbzuai.ac.ae).

[§]Source code: https://github.com/rezwanh001/SLM-RL-Agents
Model checkpoints: https://huggingface.co/mr3haque/SLM-RL-Agents
Preference datasets: https://huggingface.co/datasets/mr3haque/SLM-RL-Agents-Data

*Correspondence to: Md Rezwanul Haque <rezwan@uwaterloo.ca>.

Three engineering failure modes specific to the small-scale architectures were identified. First, in certain Parameter-Efficient Fine-Tuning (PEFT) implementations of the Transformer Reinforcement Learning (TRL) library [13], Low-Rank Adaptation (LoRA) parameters are silently registered as non-trainable. Second, bfloat16 arithmetic generates probability-ratio overflow in models with fewer than 200 million parameters. Third, unbounded reward values combined with unclipped Kullback–Leibler (KL) penalties can cause the policy to generate incoherent outputs. These failure cases are addressed using a merge-and-reinitialize method, float32 precision during training, and a multi-layer safety framework incorporating reward whitening and a weight-rollback mechanism.

A *capacity-headroom hypothesis* is proposed, suggesting that PPO effectiveness depends on both a fluent SFT prior and a discriminative reward signal, rather than on the model parameter count. The proposed system achieves performance comparable to publicly available instruction-tuned baselines, including SmolLM2-Instruct [4] and Qwen2.5-Instruct [6], while using significantly less training data.

The major contributions of this work are as follows:

1) Three reproducible failure modes of PPO at the SLM scale are identified: silent LoRA gradient freezing in PEFT, bfloat16 importance-ratio overflow, and reward-driven policy collapse. Specific solutions are proposed and organized as a three-layer cybernetic safety framework.
2) The capacity-headroom hypothesis is evaluated across fifteen configurations. The results show that PPO effectiveness at the SLM scale depends on the combination of SFT fluency and reward discriminability, providing a decision rule ($\text{PPL}_{\text{SFT}} < 20$) for practitioners.
3) A reproducible alignment framework is released, including fifteen checkpoints, preference datasets, training scripts, an interactive verification application, and a forward-compatibility system for multi-turn agentic extensions.

The remainder of this paper is organized as follows. Section II reviews related works. Section III describes the methodology and stabilization framework. Section IV presents the experimental setup. Section V presents and analyzes the experimental results. Section VI discusses practical implications and Section VII concludes the paper.

II. Related Works

The application of RLHF using SFT, reward modeling, and PPO [7]–[9], [14] is well developed for models with 1.3–175 billion parameters [15]. However, its behavior at the 70–500 million parameter scale has received limited attention. Existing SLM-alignment research primarily focuses on developing high-quality supervised models through SFT [2]–[4] or using preference-learning methods that avoid the explicit reinforcement-learning process, such as DPO [10], KTO [16], and GRPO [17]. While these alternative methods reduce training instability, they remove the explicit reward model, which is important for diagnostic transparency in agentic systems.

Parameter-efficient training using LoRA [18] and QLoRA [19], typically implemented through the TRL library [13], is a standard approach. However, silent gradient-flow problems can occur at the small-scale model level. Similarly, stabilization techniques such as adaptive KL [20], Generalized Advantage Estimation (GAE) [21], and NEFTune [22] are adopted from large-scale models without sufficient evaluation at the SLM scale. This work provides a different research direction from the DPO/GRPO-based approaches. This work identifies reproducible failure modes of standard PPO at the SLM scale, rather than proposing a new optimization objective, and provides engineering solutions to maintain the diagnostic value of a scalar reward signal for practitioners.

III. Methodology

A three-stage RLHF system is implemented, including supervised fine-tuning, Bradley–Terry reward modeling, and PPO with a KL divergence penalty. Each stage is tailored to models with fewer than 500 million parameters. The overall data flow and the three engineering mechanisms used to ensure training stability are illustrated in Fig. 1.

The agent policy π_θ operates within a finite-horizon MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, T)$. The state $s_t \in \mathcal{S}$ is represented as a structured tuple containing the task prompt, generated tokens, external observations, the turn index, and the remaining budget. The action space $\mathcal{A}$ consists of four distinct action types: vocabulary token generation $a \in \mathcal{V}$, external tool invocation with arguments, clarifying-query generation, and termination. The transition function $\mathcal{P}$ is deterministic for vocabulary generation and termination, stochastic for tool invocation, and history-dependent in all cases. The reward $\mathcal{R}$ consists of three components: a verifiable component based on deterministic checks, a preference component obtained from the Bradley–Terry model r_ϕ, and a shaping component based on the per-token KL divergence from the SFT prior.

The fifteen empirical configurations use the single-turn form of $\mathcal{M}$, where $T = 1$, $\mathcal{A} = \mathcal{V}$, $\mathcal{P}$ is deterministic, the reward is provided only at the terminal state, and the observation set is empty. In this form, the state is reduced to the context $(x, y_{<t})$ and the MDP follows the standard language-modeling formulation used in large-scale alignment [8], [14]. The stability requirements investigated in this paper, including gradient-flow integrity, importance-ratio computation, and reward regulation, are necessary for any form of $\mathcal{M}$ and can also be applied to the multi-turn version released as a forward-compatibility layer. From a control-theoretic perspective, the PPO update can be viewed as a discrete-time feedback loop involving three connected signals: reward, KL divergence, and importance ratio. The safety mechanism described in Section III-D provides specific controls that improve the robustness of the loop against reward-model misspecification. The three failure modes mentioned in Section III-D are considered as closed-loop instabilities and are addressed at the controller level using signal conditioning, deadband limiting, and state recovery.

A. Stage 0: Data and Preference Pair Preparation

Three target corpora, TinyStories ($\mathcal{D}_{\text{TS}}$), CNN/DailyMail ($\mathcal{D}_{\text{CNN}}$), and Wikitext-103 ($\mathcal{D}_{\text{WT}}$), are divided into separate splits for SFT training, preference-pair construction for reward model training, and held-out evaluation. Synthetic preference

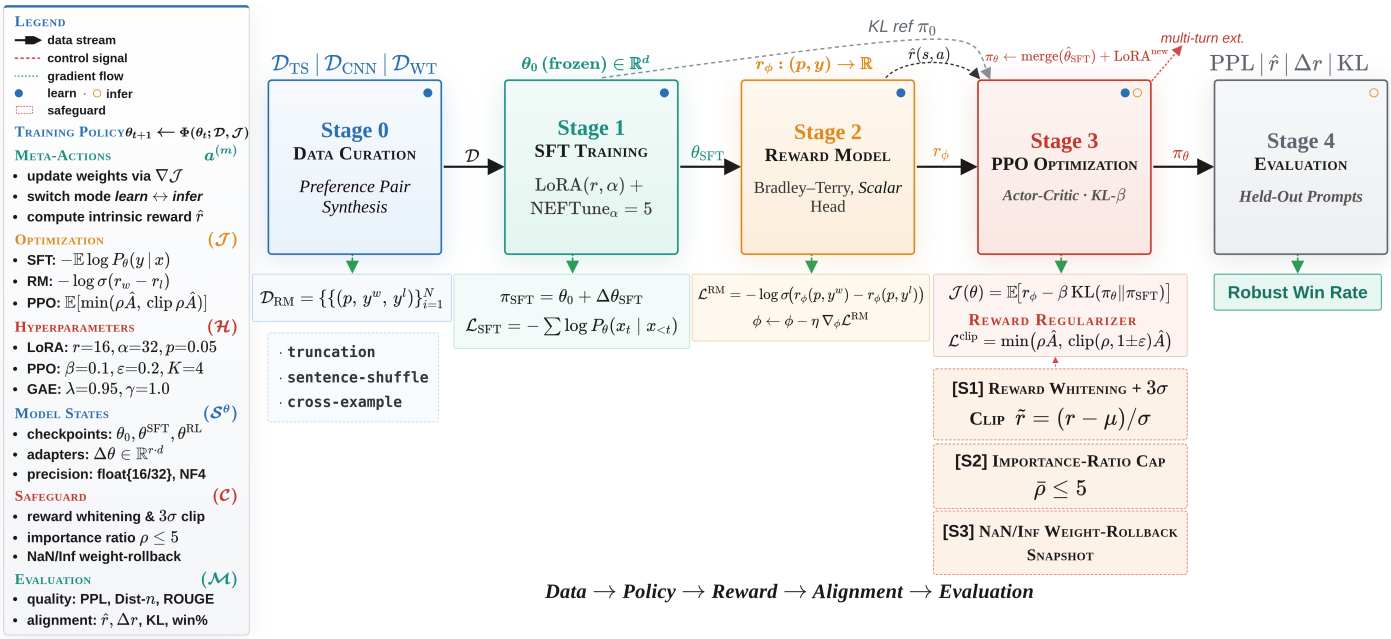

Fig. 1. **End-to-end RLHF pipeline for aligning small language model agents.** The pipeline consists of five sequential stages: (0) Data Curation, where preference pairs are generated using truncation, shuffling, and mismatch; (1) SFT Training, generating the supervised prior π_{SFT}; (2) Reward Model training using the Bradley–Terry objective; (3) PPO Optimization, applying a merge-and-reinitialize procedure and a three-layer safety mechanism (reward whitening, importance-ratio capping, and weight-rollback) to reduce scale-specific instabilities; and (4) Evaluation using held-out prompts. Solid arrows represent primary data flow, dashed red arrows indicate control signals (e.g., KL reference), and dotted arrows show gradient flow into the state boxes of each stage. The colored dots represent operating mode (blue: learn; gold: inference). *Sidebar notation:* $\hat{r}(s, a)$ is the reward from the Bradley–Terry model $r_\phi(p, y)$, and (s, a) reduces to (p, y) in the single-turn setting; Φ and $\mathcal{J}$ denote the update operator and PPO objective $J(\theta)$, both distinct from the CDF Φ and MDP tuple $\mathcal{M}$ used in the text.

pairs are generated using three different degradation strategies. Truncation generates rejected continuations by cutting a sentence at 50% of its length, usually in the middle of a clause, to penalize premature stopping. Sentence shuffling randomly changes the order of sentences in multi-sentence passages to penalize overall incoherence. Cross-example mismatch pairs a prompt with a reference text from another example to enforce topical coherence. These three strategies are used in equal proportions to construct the preference dataset $\mathcal{D}_{\text{RM}} = \{(p_i, y_i^w, y_i^l)\}$.

B. Stage 1: Supervised Fine-Tuning

The causal language-modeling objective is minimized using the SFT dataset $\mathcal{D}_{\text{SFT}} = \{x^{(i)}\}_{i=1}^N$, as defined in (1).

$$\mathcal{L}_{\text{SFT}}(\theta) \;=\; -\frac{1}{N}\sum_{i=1}^{N}\sum_{t}\log P_\theta\Big(x_t^{(i)} \,\Big|\, x_{<t}^{(i)}\Big) \qquad (1)$$

Here, $\theta_0 \in \mathbb{R}^d$ represents the pre-trained base-model parameters, which are frozen. The updates are applied only to the LoRA adapter [18], parameterized by $(A \in \mathbb{R}^{r\times d}, B \in \mathbb{R}^{d\times r})$, where $r \in \{8, 16, 32\}$ depends on the model size and $\alpha = 2r$. The effective weight matrix is given by $W_0 + \frac{\alpha}{r}BA$. LoRA is applied to all attention projection matrices and the dense layers of the Feed-Forward Network (FFN). In the GPT-NeoX architecture [23], this includes the combined query-key-value projection and the primary FFN up-projection. For Llama-style architectures [15], LoRA updates are applied to the individual query, key, and value projections, as well as the gate and down-projection matrices. NEFTune noise ($\alpha_{\text{NEFT}} = 5$) [22] is applied to reduce over-memorization during training. Optimization is performed using AdamW [24] at a peak learning rate of 2×10^{-5}, with a 6% linear warm-up and cosine decay over 5 epochs. The resulting adapter $\Delta\theta_{\text{SFT}}$ is saved as the SFT checkpoint π_{SFT}.

C. Stage 2: Reward Model Training

The reward model $r_\phi : (p, y) \mapsto \mathbb{R}$ is constructed by replacing the causal language-modeling head of π_{SFT} with a single linear projection layer. A second LoRA adapter is trained together with the reward head and the transformer body. The scoring layer is fully fine-tuned to ensure the scalar reward is properly learned. Using the preference dataset $\mathcal{D}_{\text{RM}}$, the Bradley–Terry loss [25] is minimized as defined in (2).

$$\mathcal{L}_{\text{RM}}(\phi) = -\mathbb{E}_{(p,y^w,y^l)\sim\mathcal{D}_{\text{RM}}}\Big[\log\sigma\big(r_\phi(p, y^w) - r_\phi(p, y^l)\big)\Big] \qquad (2)$$

where σ represents the logistic sigmoid function. Training is performed for two epochs using AdamW [24] with a learning rate of 10^{-5}, a per-device batch size of 8, and a gradient accumulation step of 2. The training budget is limited to reduce the risk of generating brittle reward patterns. Due to the additive invariance of the Bradley–Terry loss, the reward difference Δ is reported separately for each configuration, while a shared reward model is used for comparisons across models.

D. Stage 3: PPO Fine-Tuning and Stabilization

The policy π_{RL} is optimized using the objective function defined in (3).

$$J(\theta) = \mathbb{E}_{p\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|p)}\Big[r_\phi(p, y) - \beta\,\text{KL}\big(\pi_\theta(\cdot|p) \,\|\, \pi_{\text{SFT}}(\cdot|p)\big)\Big] \qquad (3)$$

Algorithm 1 Merge-and-Reinitialize for PEFT–PPO

Require: SFT adapter $\mathcal{A}_{\text{SFT}}^{\text{ad}}$, base model θ_0, output dir $\mathcal{O}$
Ensure: Trained policy π_θ
$\hat{\theta}_{\text{SFT}} \leftarrow \text{Merge}(\theta_0, \mathcal{A}_{\text{SFT}}^{\text{ad}})$; save to $\mathcal{O}/\text{merged}$ ▷ fold SFT into base
$\ell \leftarrow \text{LoraConfig}(r, \alpha, \text{dropout}{=}0)$ ▷ fresh, zero-init adapter
$\pi_\theta \leftarrow \text{InitPolicy}(\hat{\theta}_{\text{SFT}}, \ell)$; $\pi_{\text{ref}} \leftarrow \text{InitPolicy}(\hat{\theta}_{\text{SFT}})$
snapshot $\leftarrow$ copy of trainable parameters of π_θ
for $t = 1$ **to** T_{PPO} **do**
  Sample $(p_b, y_b)_{b=1}^{B}$ from π_θ; $r_b \leftarrow r_\phi(p_b, y_b)$
  Whiten r_b, clip to $[-3, 3]$, compute advantages via GAE
  if $\overline{\rho}_{\text{mini-batch}} > 5$ **then skip** mini-batch
  end if
  Update π_θ with PPO in `float32`
  if any parameter of π_θ is NaN or Inf **then**
    $\pi_\theta \leftarrow$ snapshot; reset optimizer moments ▷ rollback
  else
    snapshot $\leftarrow$ copy of trainable parameters of π_θ
  end if
end for
return π_θ

PPO [9] is applied with GAE [21] ($\lambda = 0.95, \gamma = 1.0$), a clipping ratio $\epsilon = 0.2$, and an adaptive KL coefficient [20] targeting a mean per-token KL of 6.0 nats, implemented using TRL [13]. Following standard RLHF practice [8], [20], the reverse (mode-seeking) KL divergence, $\text{KL}(\pi_\theta \| \pi_{\text{SFT}})$, is used as a penalty term. The current policy is represented by π_θ and the frozen SFT prior is represented by π_{SFT}. The divergence is estimated on a per-token basis using sampled roll-outs. Three systemic instabilities specific to the SLM scale are addressed at this stage.

a) Failure Mode I–Gradient Flow Obstruction in PEFT: In some PEFT implementations, adapter parameters are incorrectly set as non-trainable when used in the PPO training loop [13]. As a result, the policy generates roll-outs and computes the loss, but its underlying distribution remains unchanged because the gradient flow is blocked. This is reduced using a merge-and-reinitialize method mentioned in Algorithm 1. The SFT adapter is merged into the base model weights using the weight-merging operation (Merge$(\cdot)$). Then, a new zero-initialized LoRA adapter is attached using (InitPolicy$(\cdot)$). The policy is initialized from the SFT distribution while keeping all required parameters trainable. The merged weights $\hat{\theta}_{\text{SFT}}$ are also used as the frozen reference policy π_{ref}, ensuring that $\pi_{\text{ref}} \equiv \pi_{\text{SFT}}$ throughout training.

b) Failure Mode II–Numerical Instability in Reduced Precision: PPO importance ratios $\rho_t = \exp(\log \pi_\theta - \log \pi_{\text{SFT}})$, can suffer from catastrophic cancellation when computed in bfloat16 precision. This occurs because the difference between two similar log-probabilities loses numerical precision due to the limited seven-bit mantissa. In models with fewer than 200 million parameters, this precision limitation can cause probability-ratio values to exceed 10^6 during the first optimization steps, resulting in hardware-level exceptions. This issue is addressed by using float32 precision for all tensors in the PPO loop, including the policy, reference model, value head, and reward model.

c) Failure Mode III–Distributional Collapse: Long-tailed reward distributions combined with unclipped KL penalties can cause the policy to generate degenerate outputs. This occurs when the advantage estimates significantly exceed the clipping range ϵ, causing the optimizer to move the policy toward regions where the reference model assigns very low probability. This mechanism includes reward whitening with 3σ clipping to limit the advantage estimates, an importance-ratio threshold of 5 to skip unstable mini-batches, and a weight-rollback mechanism. The weight-rollback mechanism saves the trainable parameters before each optimizer step and restores the previous state when NaN or Inf values are detected.

TABLE I
MODEL PARAMETERS AND PER-CORPUS SFT PERPLEXITY (LOWER IS BETTER): EACH SFT CHECKPOINT IS A LORA ADAPTER TRAINED ON 10K SAMPLES FOR 5 EPOCHS. **BOLD** MARKS THE BEST PER-CORPUS VALUE. SMOLLM2 RECORDS THE STRONGEST SFT BASELINE ACROSS DOMAINS.

Family	Model	Params	L	SFT PPL ↓		
				TS	CNN	WT
Pythia [3]	Pythia-70M	70.4M	6	51.4	70.3	115.1
	Pythia-160M	162M	12	13.5	29.4	53.5
	Pythia-410M	405M	24	6.5	16.2	25.4
SmolLM2 [4]	SmolLM2-135M	135M	30	7.0	18.8	24.4
	SmolLM2-360M	361M	32	**5.3**	**12.7**	**16.7**

IV. EXPERIMENTAL SETUP

A. Models

Five small language models from two distinct architecture families were used to analyze the interaction between architecture and parameter count during RL-based alignment, as presented in Table I. The selected models range from 70M to 410M parameters, providing broad coverage of model sizes within the SLM range. The Pythia family [3] includes three deduplicated variants (70M, 160M, and 410M) that use a GPT-NeoX Byte-Pair Encoding (BPE) tokenizer and are pre-trained on the Pile dataset [23]. The SmolLM2 family [4] consists of 135M and 360M parameter models based on a Llama-style architecture with Rotary Positional Embeddings (RoPE), SwiGLU activations, and grouped-query attention. These models generate fifteen unique configuration pairs suitable for on-device deployment [5].

B. Datasets

Three corpora with different linguistic characteristics were selected: TinyStories ($\mathcal{D}_{\text{TS}}$) [2] for simple narratives, CNN/DailyMail ($\mathcal{D}_{\text{CNN}}$) [11] for formal journalism, and Wikitext-103 ($\mathcal{D}_{\text{WT}}$) [12] for technical content. For each corpus, 10,000 examples were used for SFT. Preference pairs for reward-model training were generated using three degradation methods: truncation (cutting the chosen text in the middle of a clause), sentence shuffling (randomly changing the sentence order), and cross-example mismatch (using a continuation from a different reference text). This approach ensures a consistent and reproducible synthetic training signal. Table I summarizes the model parameters and per-model SFT perplexity.

C. Training Hyperparameters

A fixed set of hyperparameters was used across all configurations without model-specific tuning. Optimization was performed using AdamW [24]. For SFT, a learning rate of

2×10^{-5} was used with a batch size of 8×4, a 6% linear warm-up, and cosine decay over 5 epochs. Reward model training used a learning rate of 1×10^{-5} and a batch size of 8×2 over 2 epochs. PPO optimization utilized a learning rate of 5×10^{-6}, a rollout batch size of 32, and a mini-batch size of 4 over 250 steps. The PPO configuration used a clipping range $\epsilon = 0.2$, GAE parameters $\lambda = 0.95$ and $\gamma = 1.0$, and an adaptive KL penalty β with a target of 6.0 nats. LoRA rank r was adjusted according to model capacity: $r = 8$ for Pythia-70M, $r = 16$ for Pythia-160M and SmolLM2-135M, and $r = 32$ for Pythia-410M and SmolLM2-360M. To ensure numerical stability, float32 precision was used for all tensors in the PPO loop, while bfloat16 precision was used for SFT and reward-model training.

D. Evaluation Metrics

Evaluation was conducted using 200 held-out prompts for each configuration. Perplexity (PPL) was computed on the generated continuations, with padding tokens masked to avoid inflated scores [12]. The reward score is computed as the mean output of r_ϕ on the model-generated responses. The reward gain Δ is defined as the signed difference between the PPO and SFT reward scores, $\bar{r}_{\text{PPO}} - \bar{r}_{\text{SFT}}$. The win rate is estimated as the probability that a PPO response receives a higher score than an SFT response. It is computed as $\Phi(\Delta/\sqrt{\sigma^2_{\text{PPO}} + \sigma^2_{\text{SFT}}})$, where Φ represents the standard normal Cumulative Distribution Function (CDF). Lexical diversity was evaluated using Distinct-1 and Distinct-2, while n-gram overlap was assessed using ROUGE-1 and ROUGE-L [26]. Due to the additive invariance of the Bradley–Terry loss, absolute reward scores are compared only within each configuration, while a shared reward model is used for comparisons across different models.

E. Computational Environment and Reproducibility

All experiments were conducted using two NVIDIA RTX A6000 GPUs (48 GB each, CUDA 12.1) and required approximately 16 GPU-hours. A post-hoc cross-verification script was used to verify all 339 scalar result fields against the raw evaluation outputs. An interactive verification application is also provided with the released code.

V. Results Analysis

A. Main Performance Analysis

The main results for all fifteen configurations are presented in Table II. PPL and reward-model scores are reported for both the SFT and PPO checkpoints, together with the reward difference (Δ) and analytical win rates. The largest positive reward gains are observed for Pythia-410M and SmolLM2-360M models on TinyStories ($\Delta = +1.355$ and $+0.724$, with win rates of 59.9% and 59.7%, respectively). On the other hand, Pythia-70M shows near-zero or negative changes across all domains and remains close to the identity diagonal in Fig. 2.

The reliability of these gains is assessed using a two-sided z-test on $n = 200$ held-out prompts for each configuration, with a 95% Confidence Interval (CI) reported for Δ. Three configurations show statistically significant improvements: Pythia-410M on TinyStories (CI $[+0.61, +2.10]$, $p < 0.001$), SmolLM2-360M on TinyStories (CI $[+0.32, +1.13]$, $p < 0.001$), and SmolLM2-360M on Wikitext-103 (CI $[+0.04, +0.50]$, $p < 0.05$). These results confirm that the observed gains are not due to sampling variation. The only significant regression is observed for Pythia-410M on Wikitext-103 ($\Delta = -1.043$, $p < 0.001$). Its SFT prior has a PPL of 25.4, which is above the fluency threshold and supports the capacity-headroom hypoth-

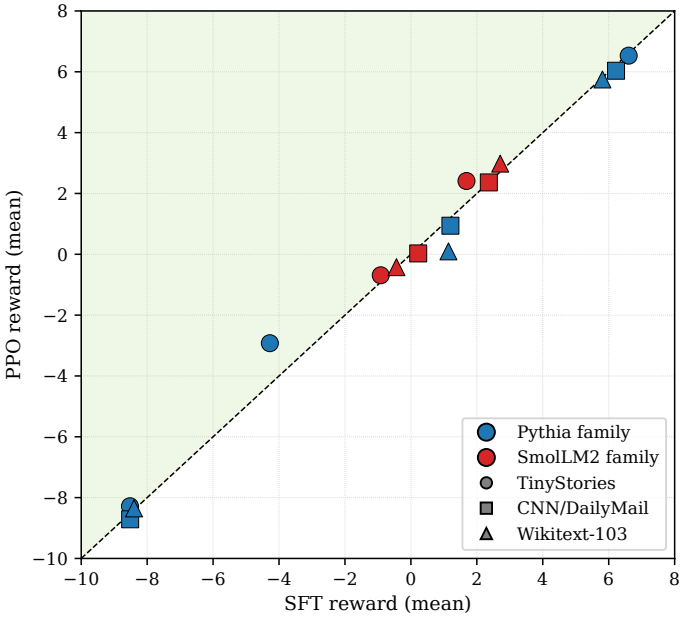

Fig. 2. **SFT versus PPO reward across all 15 configurations**. Each marker represents the mean reward for one configuration, where color indicates the architecture family and shape represents the corpus. Markers above the dashed identity line indicate improvement by PPO, while the shaded upper-triangular region represents the "PPO improves" area. A clear rightward shift is observed for Pythia-410M and SmolLM2-360M on TinyStories, while Pythia-70M remains close to the identity line.

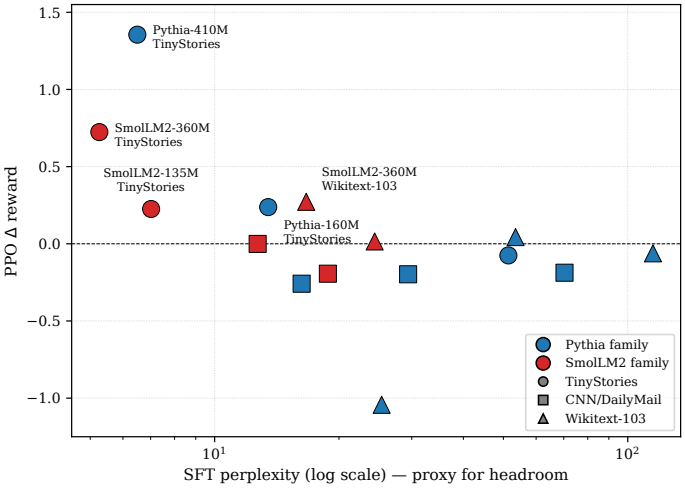

Fig. 3. **Capacity-headroom hypothesis:** PPO gain is associated with a fluent SFT prior rather than raw parameter count. The figure shows SFT perplexity (log scale, x-axis) against PPO reward delta (y-axis) for all 15 configurations. A clear negative relationship is observed: PPO improves performance when the SFT prior is already fluent (PPL $\lesssim 20$) but provides no improvement when the prior is weak.

TABLE II

MAIN RESULTS: PERPLEXITY (↓ IS BETTER) AND REWARD-MODEL SCORE (↑ IS BETTER) FOR SFT AND PPO CHECKPOINTS ACROSS 5 ARCHITECTURES × 3 CORPORA. $\Delta = \bar{r}_{\text{PPO}} - \bar{r}_{\text{SFT}}$. WIN % IS AN ANALYTICAL ESTIMATE $\Phi(\Delta/\sqrt{\sigma^2_{\text{PPO}} + \sigma^2_{\text{SFT}}})$. REWARD SCORES ARE REPORTED ON PER-CONFIGURATION SCALES; ABSOLUTE VALUES ARE NOT COMPARABLE ACROSS ROWS. **BOLD** MARKS THE BEST Δ AND WIN % PER FAMILY.

Model	Dataset	SFT Baseline		PPO-Aligned			
		PPL ↓	Reward ↑	PPL ↓	Reward ↑	Δ ↑	Win % ↑
Pythia-70M	TinyStories	51.4	$+6.61 \pm 1.63$	51.2	$+6.53 \pm 1.42$	-0.075	**48.6**
	CNN/DailyMail	70.3	$+6.22 \pm 1.21$	70.5	$+6.04 \pm 1.23$	-0.187	45.7
	Wikitext-103	115.1	$+5.81 \pm 1.24$	116.7	$+5.75 \pm 1.30$	$-$**0.062**	**48.6**
Pythia-160M	TinyStories	13.5	-8.52 ± 2.39	13.5	-8.28 ± 2.46	$+$**0.238**	**52.8**
	CNN/DailyMail	29.4	-8.52 ± 1.31	29.4	-8.71 ± 1.19	-0.198	45.6
	Wikitext-103	53.5	-8.40 ± 2.65	53.2	-8.35 ± 2.34	$+0.044$	50.5
Pythia-410M	TinyStories	6.5	-4.28 ± 4.14	7.3	-2.92 ± 3.48	$+$**1.355**	**59.9**
	CNN/DailyMail	16.2	$+1.20 \pm 1.76$	17.1	$+0.94 \pm 1.79$	-0.259	45.9
	Wikitext-103	25.4	$+1.14 \pm 2.89$	27.5	$+0.10 \pm 2.84$	-1.043	39.9
SmolLM2-135M	TinyStories	7.0	-0.92 ± 2.26	7.4	-0.69 ± 1.96	$+$**0.226**	**53.0**
	CNN/DailyMail	18.8	$+0.22 \pm 1.90$	19.2	$+0.03 \pm 1.90$	-0.194	47.1
	Wikitext-103	24.4	-0.44 ± 1.53	25.1	-0.42 ± 1.41	$+0.015$	50.3
SmolLM2-360M	TinyStories	5.3	$+1.69 \pm 2.25$	5.3	$+2.41 \pm 1.89$	$+$**0.724**	**59.7**
	CNN/DailyMail	12.7	$+2.36 \pm 1.09$	12.8	$+2.36 \pm 1.05$	-0.001	50.0
	Wikitext-103	16.7	$+2.71 \pm 1.28$	16.9	$+2.98 \pm 1.06$	$+0.272$	56.5

esis. The intervals are calculated from the complete per-prompt reward distribution rather than from repeated seeds, providing a direct measure of reliability within each configuration.

B. Capacity-Headroom Hypothesis

Figure 3 shows a monotonic negative relationship between SFT perplexity and PPO reward gain, with an inflection point at $\text{PPL}_{\text{SFT}} \approx 20$. This pattern supports the *capacity-headroom hypothesis*, which suggests that PPO effectiveness depends on both a fluent SFT prior and a discriminative reward signal, rather than on parameter count alone. Under this framework, a fluent prior keeps the generated samples within the training distribution of the reward model. In contrast, a weak prior generates a gradient that is difficult to distinguish from noise, which explains the lack of policy improvement for Pythia-70M across all domains. The hypothesis is supported by the results in Table II. The configurations with $\text{PPL}_{\text{SFT}} > 20$ show negligible or negative reward gains, while all reward gains above $+0.2$ occur only below this threshold. This boundary is presented as an empirical decision rule based on the evaluated corpora and the 70–500M parameter range, rather than as a universal threshold.

C. Comparison with SOTA Instruction-Tuned Baselines

A comparison with publicly available instruction-tuned baselines is presented in Table III. Domain-specific SFT consistently achieves lower perplexity than SmolLM2-360M-Instruct and Qwen2.5-0.5B-Instruct on all tested datasets. At the 360M parameter scale, the PPO-aligned checkpoint achieves the highest reward scores on both TinyStories ($+2.41$) and Wikitext-103 ($+2.98$). On Wikitext-103, it also reduces most of the lexical-diversity gap with the instruction-tuned baselines (0.310 vs. Qwen 0.282 and SmolLM2-360M-Instruct 0.331), while using several orders of magnitude less training data.

D. Text Quality and Diversity

Text diversity and reference-overlap metrics are shown in Table IV. PPO-aligned D-1 remains within ± 0.04 of the SFT

TABLE III

COMPARISON WITH PUBLICLY AVAILABLE SOTA SLM BASELINES: ALL MODELS ARE EVALUATED USING THE SAME REWARD MODEL FOR EACH DATASET. PPL IS COMPUTED USING REFERENCE TEXT; R DENOTES THE MEAN REWARD-MODEL SCORE; D1 IS DISTINCT-1 DIVERSITY. THE *proposed (SFT)* ROWS USE ONLY 5 EPOCHS OF LORA SFT ON 10K EXAMPLES, WHILE THE *proposed (PPO)* ROWS INCLUDE THE COMPLETE RLHF PIPELINE. **BOLD** INDICATES THE BEST VALUE IN EACH COLUMN WITHIN EACH PARAMETER CLASS, I.E., THE LOWEST PPL AND THE HIGHEST R AND D1.

Class	Model	Training Regime	TinyStories			CNN/DailyMail			Wikitext-103		
			PPL↓	R↑	D1	PPL↓	R↑	D1	PPL↓	R↑	D1
135M class	SmolLM2-135M-Instruct [4]	instruction-tuned (2T tok)	8.5	$-$**0.52**	**0.195**	19.8	$+$**0.35**	**0.283**	34.3	-0.79	**0.297**
	SmolLM2-135M (proposed, SFT)	LoRA SFT, 5 ep, 10K ex	**7.0**	-0.92	0.147	**18.8**	$+0.22$	0.248	**24.4**	-0.44	0.230
	SmolLM2-135M (proposed, PPO)	+ 250-step PPO RLHF	7.4	-0.69	0.172	19.2	$+0.03$	0.250	25.1	$-$**0.42**	0.269
360M class (incl. 500M)	SmolLM2-360M-Instruct [4]	instruction-tuned (4T tok)	6.6	$+1.35$	0.220	14.7	$+$**3.08**	**0.329**	24.3	$+2.58$	**0.331**
	Qwen2.5-0.5B-Instruct [6]	instruction-tuned (18T tok)	7.2	$+1.32$	**0.229**	19.9	$+2.58$	0.245	25.8	$+1.83$	0.282
	SmolLM2-360M (proposed, SFT)	LoRA SFT, 5 ep, 10K ex	**5.3**	$+1.69$	0.135	**12.7**	$+2.36$	0.243	**16.7**	$+2.71$	0.227
	SmolLM2-360M (proposed, PPO)	+ 250-step PPO RLHF	**5.3**	$+$**2.41**	0.156	12.8	$+2.36$	0.247	16.9	$+$**2.98**	0.310

TABLE IV
TEXT DIVERSITY AND REFERENCE-OVERLAP METRICS FOR THE SFT PRIOR AND PPO-ALIGNED POLICY ACROSS ALL FIFTEEN CONFIGURATIONS. **BOLD** INDICATES IMPROVEMENT IN THE PPO POLICY COMPARED WITH THE SFT PRIOR.

Model	Dataset	SFT Prior				PPO Policy			
		D-1	D-2	R-1	R-L	D-1	D-2	R-1	R-L
Pythia-70M	TinyStories	0.051	0.123	0.172	0.143	0.050	0.120	0.166	0.138
	CNN/DailyMail	0.130	0.329	0.184	0.115	0.124	0.310	0.178	0.111
	Wikitext-103	0.084	0.189	0.118	0.095	0.081	0.186	0.110	0.087
Pythia-160M	TinyStories	0.103	0.320	0.241	0.161	**0.104**	**0.329**	**0.246**	**0.163**
	CNN/DailyMail	0.195	0.526	0.237	0.130	**0.201**	**0.543**	0.236	**0.131**
	Wikitext-103	0.143	0.399	0.171	0.119	0.132	0.366	**0.172**	**0.120**
Pythia-410M	TinyStories	0.119	0.456	0.351	0.202	0.110	0.390	0.323	0.194
	CNN/DailyMail	0.225	0.632	0.263	0.139	0.213	0.609	**0.274**	**0.145**
	Wikitext-103	0.165	0.541	0.219	0.139	0.144	0.493	0.216	**0.140**
SmolLM2-135M	TinyStories	0.147	0.513	0.310	0.181	**0.172**	**0.579**	0.253	0.154
	CNN/DailyMail	0.248	0.682	0.246	0.129	**0.250**	0.678	0.232	0.123
	Wikitext-103	0.230	0.638	0.185	0.120	**0.269**	**0.675**	0.149	0.102
SmolLM2-360M	TinyStories	0.135	0.489	0.355	0.212	**0.156**	**0.516**	0.335	0.205
	CNN/DailyMail	0.243	0.666	0.256	0.136	**0.247**	**0.670**	0.249	0.133
	Wikitext-103	0.227	0.651	0.206	0.138	**0.310**	**0.731**	0.141	0.102

TABLE V
QUALITATIVE COMPARISON ON TINYSTORIES (PYTHIA-410M). A PROMPT FROM THE EVALUATION SET IS FOLLOWED BY TRUNCATED SFT AND PPO CONTINUATIONS. THE SFT CONTINUATION BECOMES REPETITIVE, WHILE THE PPO CONTINUATION MAINTAINS TOPICAL RELEVANCE.

Prompt: "Once upon a time, there was a mama and her little child. The child was only three years old…"
SFT: "…boy thought it was great, but he was very shy. So, the mama said, 'no, the laundry must not be done here. Let's go and clean the house'. The little boy said, 'I can't do the laundry. I am a little boy' *[unresolved loop]*" **PPO:** "…boy was very excited and ran to the laundry machine. He ran to the clothes and washed them carefully. He did not know that he would be doing it quickly and carefully. The little boy was so excited to *[continues on-topic]*"
Per-configuration rewards: SFT -4.28*, PPO* -2.92*;* Δ *reward* $+1.355$.

prior in fourteen of the fifteen configurations. The only exception is SmolLM2-360M on Wikitext-103, where D-1 increases by $+0.083$ together with the largest reward gain in this capacity class. Within the SmolLM2 family, D-1 improves across all configurations. In the Pythia family, a small reduction in D-1 is observed when the reward signal is uninformative, indicating a shift toward the SFT distribution rather than policy collapse. ROUGE-1 and ROUGE-L remain stable across most configurations. The largest reduction is observed for the two SmolLM2 models on Wikitext-103, where increased diversity is associated with lower reference overlap. No evidence of mode collapse is observed in any configuration. A representative example is shown in Table V, where the SFT continuation becomes repetitive, while the PPO continuation remains relevant to the topic.

E. Ablation: Effect of the Stabilization Mechanism

The impact of the proposed engineering methods is evaluated through an ablation study using the Pythia-70M model on the TinyStories dataset. Three configurations are compared: (a) Naive PEFT, where the SFT adapter is directly passed to the trainer; (b) Manual unfreeze, where LoRA parameters are made trainable without using a reference model; and (c) the proposed stabilization mechanism. Configuration (a) produces a reward delta of zero because no gradient flows through the LoRA parameters. Configuration (b) fails during the first mini-batch due to numerical instability and the absence of a reference distribution. Only configuration (c) completes the 250-step training run without catastrophic failure. Each mechanism addresses a specific failure mode: merge-and-reinitialize restores gradient flow in (a), float32 precision with an importance-ratio cap ($\bar{\rho} \leq 5$) prevents overflow in (b), and reward whitening with 3σ clipping and weight rollback prevents policy collapse, keeping all fifteen runs within 15% of the SFT perplexity.

VI. DISCUSSION

Engineering Recommendations. To ensure the stability of SLM-scale training, three essential methods are implemented. Merge-and-reinitialization is applied when TRL is combined with LoRA-adapted policies, while float32 precision is recommended for models with fewer than 200M parameters, and weight-rollback is used to prevent unrecoverable policy collapse. The convergence process is mainly controlled by an adaptive KL target of 6.0 nats and an importance-ratio cap of $\bar{\rho} \leq 5$, where $\bar{\rho}$ is the mini-batch mean of ρ_t. The importance-ratio cap is the most sensitive parameter for models with fewer than 200M parameters. Since the loop remains stable across the tested learning-rate and clipping-ratio ranges, the same configuration is used for all fifteen cases. The merge operation is performed once before training with $\mathcal{O}(|\theta|)$ complexity and is independent of the number of PPO steps. At larger scales, the same gradient-flow property can be achieved by re-enabling the adapter gradients after the PEFT round-trip.

Deployment Decision Rule. The capacity-headroom hypothesis provides a practical guideline for SLM agents in adaptive and human–machine systems. A 70–500M parameter agent can be reliably aligned when (i) the supervised prior is sufficiently fluent (PPL $<$ 20) to keep samples within the training distribution of the reward model, and (ii) the reward signal is sufficiently discriminative to provide a useful gradient. When PPL $\in [20, 50]$, the expected improvement is limited and regression may occur, as observed for Pythia-410M on Wikitext-103. In this range, computational resources may be better used to improve SFT data quality or increase the LoRA rank. When PPL exceeds 50, PPO is unlikely to provide improvement. PPO may drift or collapse if either condition is not satisfied, while the safety mechanism can reduce but not fully resolve the underlying capacity limitation. These findings connect the empirical results with the broader concept of hybrid and cybernetically regulated computational-intelligence systems.

Forward Compatibility with Multi-Turn Agents. The empirical claims of this paper are limited to the single-turn form of the agent MDP, where the action space is reduced to the vocabulary $\mathcal{V}$, the horizon consists of one turn, and the reward is provided at the terminal state. A forward-compatibility framework for the multi-turn case is released with the fifteen checkpoints. It includes a parser for four sentinel-delimited action types, a set of deterministic and verifiable tools, and a multi-turn PPO variant that applies the safety mechanism at each turn. This framework is released as an unvalidated, forward-looking component rather than an evaluated contribution, and no multi-

turn experiments are reported. The three identified failure modes remain necessary conditions for any form of the agent MDP, which motivates releasing the framework as a basis for future work on tool-call reward design and per-turn safety analysis.

Limitations and Future Work. Five boundary conditions are considered to define the scope of this study. First, synthetic preference pairs may limit the discriminative ability of the reward model, while human-annotated pairs could provide a stronger signal for complex corpora. Second, the 250-step PPO training budget is intentionally limited and may not capture the full improvements possible with longer training schedules. Third, the Bradley–Terry objective is additively invariant, which makes absolute reward values incomparable across different configurations. Fourth, the evaluation is limited to generative quality, while downstream task performance and human preference studies are needed for a complete evaluation of agent utility. Finally, the empirical evaluation is limited to the single-turn form of the agent MDP. The released forward-compatibility framework supports future extension of the safety mechanism to multi-turn interactions, where tool invocations and external observations increase the state space. Since DPO does not use the explicit reward model and safety mechanism investigated in this paper, a controlled empirical comparison would address a different research question and is considered a direct extension of this work.

VII. Conclusion

This paper investigates the applicability of PPO for aligning small language model agents. Three reproducible failure modes at the 70–500M scale were identified: silent LoRA gradient freezing, bfloat16 importance-ratio overflow, and reward-driven policy collapse. These issues were addressed using a three-layer cybernetic safety mechanism that transforms the PPO update from an unstable process into a more reliable engineering approach. The resulting checkpoints achieve competitive performance compared with SOTA instruction-tuned baselines while using significantly less training data. The results support the capacity-headroom hypothesis across fifteen (model, corpus) configurations, showing that PPO effectiveness at the SLM scale depends on both a fluent SFT prior and a discriminative reward signal, rather than on the absolute number of model parameters. It remains an open research question whether this boundary also applies to group-relative methods such as GRPO and REINFORCE Leave-One-Out (RLOO), models at the 1B parameter scale, and multi-turn agent MDPs. The released checkpoints, datasets, scripts, and forward-compatibility framework are expected to facilitate these future investigations.

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[2] R. Eldan and Y. Li, "TinyStories: How Small Can Language Models Be and Still Speak Coherent English?" *arXiv:2305.07759*, 2023.

[3] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff *et al.*, "Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling," in *International Conference on Machine Learning*. PMLR, 2023, pp. 2397–2430.

[4] L. B. Allal, A. Lozhkov, E. Bakouch, G. M. Blázquez, G. Penedo, L. Tunstall, A. Marafioti, H. Kydlíček, A. P. Lajarín, V. Srivastav *et al.*, "SmolLM2: When Smol Goes Big–Data-Centric Training of a Small Language Model," *arXiv:2502.02737*, 2025.

[5] Z. Liu, C. Zhao, F. Iandola, C. Lai, Y. Tian, I. Fedorov, Y. Xiong, E. Chang, Y. Shi, R. Krishnamoorthi *et al.*, "MobileLLM: Optimizing Sub-billion Parameter Language Models for On-Device Use Cases," *arXiv:2402.14905*, 2024.

[6] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei *et al.*, "Qwen2.5 Technical Report," *arXiv:2412.15115*, 2024.

[7] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep Reinforcement Learning from Human Preferences," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[8] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training Language Models to Follow Instructions with Human Feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv:1707.06347*, 2017.

[10] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," *Advances in Neural Information Processing Systems*, vol. 36, pp. 53 728–53 741, 2023.

[11] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond," in *Proceedings of the SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 280–290.

[12] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer Sentinel Mixture Models," *arXiv:1609.07843*, 2016.

[13] L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, and S. Huang, "TRL: Transformer Reinforcement Learning," *GitHub*, 2022. [Online]. Available: https://github.com/huggingface/trl

[14] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to Summarize with Human Feedback," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3008–3021, 2020.

[15] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open Foundation and Fine-Tuned Chat Models," *arXiv:2307.09288*, 2023.

[16] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela, "KTO: Model Alignment as Prospect Theoretic Optimization," *arXiv:2402.01306*, 2024.

[17] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu, and D. Guo, "DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models," *arXiv:2402.03300*, 2024.

[18] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," *International Conference on Learning Representations (ICLR)*, vol. 1, no. 2, p. 3, 2022.

[19] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," *Advances in Neural Information Processing Systems*, vol. 36, pp. 10 088–10 115, 2023.

[20] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-Tuning Language Models from Human Preferences," *arXiv:1909.08593*, 2019.

[21] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," *arXiv:1506.02438*, 2015.

[22] N. Jain, P.-y. Chiang, Y. Wen, J. Kirchenbauer, H.-M. Chu, G. Somepalli, B. Bartoldson, B. Kailkhura, A. Schwarzschild, A. Saha *et al.*, "NEFTune: Noisy Embeddings Improve Instruction Finetuning," in *International Conference on Learning Representations*, vol. 2024, 2024, pp. 17 566–17 591.

[23] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima *et al.*, "The Pile: An 800gb Dataset of Diverse Text for Language Modeling," *arXiv:2101.00027*, 2020.

[24] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv:1711.05101*, 2017.

[25] R. A. Bradley and M. E. Terry, "Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[26] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *Text Summarization Branches Out*, 2004, pp. 74–81.