The GRPO Tax is Smaller Than You Think: A Longitudinal Study of Capability Preservation During Reasoning Training

Anonymous authors
Paper under double-blind review

Abstract

Reinforcement learning methods such as Group Relative Policy Optimization (GRPO) have emerged as a dominant paradigm for training language models to reason, following the success of DeepSeek-R1. A persistent concern is the "alignment tax": the degradation of general-purpose capabilities that accompanies post-training optimization for a specific objective. We conduct the first dense-checkpoint longitudinal study of capability evolution during GRPO training across five instruction-tuned models from four families (Qwen, Phi, Gemma, and Llama) at the 1.5B to 3.8B parameter scale. Each model is evaluated at 75 training checkpoints across 13 benchmarks. Our findings reveal that the GRPO tax is substantially smaller than commonly assumed: 85% of non-target capabilities remain within ±2% of their baseline values, 8% improve, and only 7% degrade. We further observe directional evidence of a cross-family divergence in safety behavior: Qwen and Phi models maintain stable safety refusal rates, while Gemma and Llama models each show one additional refusal failure out of 30 safety prompts (−3.3%), a pattern warranting further investigation with larger evaluation sets. A comparative analysis with Direct Preference Optimization (DPO) reveals that GRPO produces larger variance in capability shifts than DPO, which acts as a more conservative optimization method. These results challenge the prevailing narrative of catastrophic capability loss and provide practitioners with concrete evidence that single-epoch LoRA-based GRPO training largely preserves the instruction-tuned model's general capabilities.

1 Introduction

The release of DeepSeek-R1 (DeepSeek-AI, 2025) demonstrated that reinforcement learning, specifically Group Relative Policy Optimization (GRPO), can elicit sophisticated reasoning capabilities in large language models without explicit supervision of reasoning chains. This result catalyzed widespread adoption of RL-based post-training pipelines for reasoning, with subsequent work applying GRPO and related methods to domains ranging from code generation to scientific problem-solving (Yu et al., 2025).

A central concern is the alignment tax: optimizing a model for one objective tends to degrade performance on others (Lin et al., 2024a). This phenomenon has been documented for RLHF (Ouyang et al., 2022), safety alignment (Huang et al., 2025), and supervised fine-tuning (Li et al., 2024), but the specific dynamics of capability change during GRPO training remain underexplored.

Existing studies use a pre-post evaluation paradigm that obscures the temporal dynamics of capability change: when does degradation begin, how fast does it progress, and is there an optimal early-stopping point? Furthermore, most analyses focus on a single model family, making it difficult to distinguish model-specific artifacts from generalizable phenomena.

In this work, we address these gaps through a dense-checkpoint longitudinal study. We train five instruction-tuned models from four families (Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, Phi-3.5-mini-instruct, Gemma-2-2B-it, and Llama-3.2-3B-Instruct) using GRPO on the GSM8K mathematical reasoning dataset (Cobbe et al., 2021), saving model checkpoints every 50 training steps. Each of the resulting 380 checkpoints (plus 5 baselines) is evaluated on 13 benchmarks spanning mathematical reasoning (Cobbe et al., 2021), general

knowledge (Hendrycks et al., 2021), commonsense reasoning (Zellers et al., 2019), scientific reasoning (Clark et al., 2018), truthfulness (Lin et al., 2022), coreference resolution (Sakaguchi et al., 2020), instruction following, summarization, translation, code generation, safety, creative writing, and conversational quality. This produces over 5,616 individual evaluation data points, enabling fine-grained analysis of capability trajectories during training.

Our contributions are threefold. First, we establish that the LoRA-based GRPO tax on non-reasoning capabilities is substantially smaller than the prevailing narrative suggests: across 60 model-benchmark pairs (excluding the target task), 85% of capabilities remain within ±2% of baseline, 8% improve, and only 7% degrade beyond the 2% threshold. Second, we observe directional evidence of a cross-family divergence in safety behavior during GRPO training: models from the Qwen and Phi families maintain stable safety refusal rates, while models from the Gemma and Llama families each show one additional refusal failure out of 30 safety prompts (−3.3%). While not statistically significant at N=30, the consistency of this pattern across both degrading families is suggestive and practically relevant. Third, we provide a comparative analysis of GRPO and DPO training dynamics on the Qwen family, revealing that DPO acts as a more conservative optimizer with smaller capability variance than GRPO.

2 Related Work

Reinforcement Learning for Reasoning. The application of reinforcement learning to enhance reasoning in language models has accelerated following the release of DeepSeek-R1 (DeepSeek-AI, 2025), which demonstrated that GRPO with verifiable rewards can elicit chain-of-thought reasoning without supervised reasoning traces. Subsequent work has extended GRPO to diverse domains including vulnerability detection (Simoni et al., 2025), medical reasoning (Pan et al., 2025), and multi-turn planning (Hu et al., 2025). Algorithmic improvements to GRPO have been proposed through scaffolded training curricula (Zhang et al., 2025), entropy-weighted reward shaping (Tan & Pan, 2025), and soft-token policy optimization (Zheng & Lee, 2025). The DAPO framework (Yu et al., 2025) demonstrated GRPO at scale, achieving competitive performance on AIME 2024 benchmarks.

The Alignment Tax. The observation that post-training optimization degrades non-target capabilities was first systematically studied by Lin et al. (2024a), who coined the term "alignment tax" and demonstrated monotonic capability degradation across NLP benchmarks during RLHF training. Huang et al. (2025) extended this analysis to reasoning models, identifying a trade-off between reasoning capability and safety alignment in the sequential production pipeline for large reasoning models. Lu et al. (2024b) proposed online merging optimizers to mitigate the alignment tax by integrating RL policy and SFT models at each optimization step. More recently, Sun et al. (2025) introduced orthogonal gradient projection to constrain safety updates to the null space of general capability gradients.

Catastrophic Forgetting in Post-Training. Catastrophic forgetting during LLM fine-tuning has been studied from multiple perspectives. Li et al. (2024) demonstrated that sharpness-aware minimization can flatten the loss landscape to reduce forgetting during supervised fine-tuning. Wolczyk et al. (2024) conceptualized the interplay between pre-trained capabilities and RL-specific adaptations as a forgetting problem, showing that standard retention techniques can preserve pre-trained knowledge. Lai et al. (2025) provided a comparative analysis showing that reinforcement fine-tuning naturally mitigates forgetting relative to supervised fine-tuning in continual post-training settings. The CapTrack framework (Thede et al., 2026) proposed a capability-centric taxonomy for evaluating forgetting across behavioral dimensions, though their analysis was limited to supervised and DPO post-training without GRPO coverage.

Capability Evaluation During Training. Several recent works have analyzed how capabilities evolve during RL training. Yao et al. (2025) identified a two-stage dynamic in RLVR training, with early exploitation followed by exploration. Wang et al. (2026) showed that model weights and output probabilities evolve linearly with RLVR training steps. Yue et al. (2025) controversially argued that RLVR does not create fundamentally new reasoning patterns but rather improves sampling efficiency over the base model's existing distribution. However, these studies either focus on the target reasoning task or adopt a coarse-grained evaluation protocol.

Our work complements this literature by providing the first dense-checkpoint, multi-benchmark, cross-family longitudinal analysis specifically focused on non-target capability preservation during GRPO training.

Direct Preference Optimization. DPO (Rafailov et al., 2023) has emerged as a popular alternative to RL-based alignment, offering simpler training dynamics by directly optimizing the policy on preference pairs without a separate reward model. Comparative analyses of DPO and RL methods have examined aspects such as generalization (Lin et al., 2024b), length bias (Lu et al., 2024a), and robustness (Wu et al., 2024). Our work contributes to this comparative literature by examining how DPO and GRPO differ in their impact on non-target capabilities during training.

3 Methodology

3.1 Problem Formulation

Let π_θ denote a language model parameterized by θ, initialized from an instruction-tuned checkpoint π_{θ_0}. GRPO training optimizes θ to maximize a reward function $r(\mathbf{y} \mid \mathbf{x})$ on a task-specific dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i^*)\}_{i=1}^N$, where $\mathbf{x}_i$ is a prompt and $\mathbf{y}_i^*$ is the verifiable ground truth. We define the capability profile of π_θ as its performance vector across K evaluation benchmarks:

$$\mathbf{c}(\theta) = [s_1(\pi_\theta), s_2(\pi_\theta), \ldots, s_K(\pi_\theta)] \in [0,1]^K$$

where $s_k(\pi_\theta)$ denotes the score of π_θ on benchmark k. The GRPO tax for benchmark k at training step t is defined as:

$$\Delta_k(t) = s_k(\pi_{\theta_t}) - s_k(\pi_{\theta_0}) \tag{1}$$

A negative $\Delta_k(t)$ indicates degradation on benchmark k, a positive value indicates improvement, and a value near zero indicates preservation. The aggregate preservation rate at step t is the fraction of non-target benchmarks for which $|\Delta_k(t)| \leq \epsilon$, where ϵ is a tolerance threshold (set to 0.02 in our experiments).

3.2 GRPO Training

GRPO (DeepSeek-AI, 2025) optimizes the policy by generating a group of G completions $\{\mathbf{y}_1, \ldots, \mathbf{y}_G\}$ for each prompt $\mathbf{x}$, computing rewards $\{r_1, \ldots, r_G\}$, and using the relative ranking within the group to compute advantages. The core policy gradient (simplified for clarity; the full objective includes importance sampling clipping and a KL penalty against the reference policy) is:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G} A_i \nabla_\theta \log \pi_\theta(\mathbf{y}_i \mid \mathbf{x})\right]$$

where $A_i = \frac{r_i - \mu_r}{\sigma_r}$ is the advantage computed from the group statistics μ_r and σ_r. The KL penalty $D_{\mathrm{KL}}(\pi_\theta \| \pi_{\theta_0})$ constrains the updated policy to remain close to the initial instruction-tuned model, which is directly relevant to our study of capability preservation. In our experiments, we use $G = 4$ with a binary correctness reward: $r(\mathbf{y}, \mathbf{y}^*) = \mathbb{1}[\mathrm{extract}(\mathbf{y}) = \mathbf{y}^*]$, where extract$(\cdot)$ extracts the final numerical answer from the model's response using a regular expression that searches for the delimiter "####" followed by a number, with fallback to the last number in the response.

3.3 Dense Checkpoint Evaluation Protocol

Our experimental protocol differs from the standard pre-post evaluation paradigm in two key respects. First, we save model checkpoints at regular intervals of $\Delta t = 50$ training steps, yielding approximately 75 checkpoints per model over one epoch of training. Second, we evaluate every checkpoint on the full suite of

$K = 13$ benchmarks. This produces a capability trajectory matrix $\mathbf{C} \in \mathbb{R}^{T \times K}$, where T is the number of checkpoints:

$$C_{t,k} = s_k(\pi_{\theta_t}), \quad t \in \{0, 1, \ldots, T-1\}, \quad k \in \{1, \ldots, K\}$$

This matrix enables analysis of degradation onset (the first step at which $\Delta_k(t) < -\epsilon$), degradation rate (the slope of capability decline), and capability correlation (whether benchmarks co-move during training).

3.4 Onset Detection

For each benchmark k and model m, we define the onset step t_k^* as the first training step at which the smoothed capability score drops below 98% of the baseline. We use a relative threshold (2% of the baseline value) rather than an absolute threshold, which ensures that the sensitivity of onset detection scales with the baseline score:

$$t_k^* = \min \{t : \bar{s}_k(t) < 0.98 \cdot s_k(\pi_{\theta_0})\}$$

where $\bar{s}_k(t) = \frac{1}{3} \sum_{j=t-2}^{t} s_k(\pi_{\theta_j})$ is a 3-point moving average applied to reduce evaluation noise. If no such step exists, the capability is classified as "stable" for that model-benchmark pair. We note that the $\epsilon = 0.02$ absolute threshold used for the aggregate preservation statistics (Equation (1)) differs from the 2% relative threshold used here; we report both to provide complementary views of capability stability.

4 Experimental Setup

4.1 Models

We select five instruction-tuned models spanning four model families and two parameter scales, as summarized in Table 1. All models are instruction-tuned variants to ensure comparable starting conditions (instruction following, safety alignment, and conversational ability). We apply Low-Rank Adaptation (LoRA) (Hu et al., 2022) with rank $r = 16$ and scaling factor $\alpha = 32$ to all models, targeting the query, key, value, output, gate, up, and down projection matrices.

Table 1: Models evaluated in this study. All models are instruction-tuned variants trained with LoRA ($r = 16$, $\alpha = 32$).

Model	Family	Params	VRAM (bf16)	License
Qwen2.5-1.5B-Instruct	Qwen	1.5B	6 GB	Apache 2.0
Qwen2.5-3B-Instruct	Qwen	3.0B	10 GB	Apache 2.0
Phi-3.5-mini-instruct	Phi	3.8B	10 GB	MIT
Gemma-2-2B-it	Gemma	2.0B	6 GB	Gemma License
Llama-3.2-3B-Instruct	Llama	3.2B	10 GB	Llama 3.2 License

4.2 GRPO Training Configuration

All models are trained on the GSM8K training set (Cobbe et al., 2021) containing 7,473 grade school math word problems. The reward function returns 1.0 if the model's extracted final answer matches the ground truth and 0.0 otherwise. Training hyperparameters are held constant across all models and are summarized in Table 2. Each model is trained for one epoch with checkpoints saved every 50 steps, yielding approximately 75 checkpoints per model.

Table 2: Training hyperparameters. All values are identical across the five GRPO-trained models.

GRPO Training		**DPO Training**	
Dataset	openai/gsm8k (train)	Dataset	UltraFeedback (10K subset)
Training examples	7,473	Training examples	10,000
Epochs	1	Epochs	1
Learning rate	5×10^{-6}	Learning rate	5×10^{-7}
LR scheduler	Cosine	LR scheduler	Cosine
Warmup steps	20	Warmup steps	20
Batch size (per device)	1	Batch size (per device)	1
Gradient accumulation	8	Gradient accumulation	8
Effective batch size	8	Effective batch size	8
Group size (G)	4	DPO β	0.1
Max completion length	512 tokens	Max sequence length	768 tokens
Precision	bf16	Precision	bf16
Gradient checkpointing	Yes	Gradient checkpointing	Yes
Optimizer	AdamW	Optimizer	AdamW
Random seed	42	Random seed	42
Checkpoint interval	50 steps	Checkpoint interval	50 steps
LoRA Configuration (shared)			
Rank (r)	16		
Scaling factor (α)	32		
Dropout	0.05		
Target modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj		

4.3 DPO Comparison

For the DPO comparison, we train Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct on the UltraFeedback Binarized dataset (Cui et al., 2023) containing approximately 10,000 preference pairs (subsampled from the full 61K for comparable training duration). DPO training uses learning rate 5×10^{-7}, $\beta = 0.1$, maximum sequence length 768, and identical LoRA configuration. Checkpoints are saved every 50 steps, yielding approximately 25 checkpoints per model. We acknowledge that the GRPO and DPO experiments use different training datasets (GSM8K for GRPO, UltraFeedback for DPO), which confounds the comparison between algorithms and training data. We chose to use the canonical dataset for each method (math reasoning for GRPO, preference data for DPO) rather than forcing both methods onto the same dataset, as this reflects how practitioners deploy these methods in practice. Differences in capability shifts should be interpreted as reflecting the combined effect of algorithm and dataset, not the algorithm alone.

4.4 Evaluation Benchmarks

We evaluate on 13 benchmarks organized into three categories based on their measurement characteristics, as detailed in Table 3.

The six established benchmarks (MMLU, HellaSwag, ARC-Challenge, TruthfulQA, Winogrande, and GSM8K) use standard multiple-choice or numerical answer extraction protocols. The two applied benchmarks (XSum and WMT) use standard automated metrics (ROUGE-L and BLEU respectively). The five heuristic benchmarks use rule-based scoring that approximates the target capability, and we acknowledge these as approximate measures. Specifically: IFEval uses 30 prompts with verifiable constraints (e.g., "list exactly 3 items," "respond in all caps") and scores the fraction of constraints satisfied via regex matching. Coding presents 25 Python programming tasks and scores syntactic validity (via `ast.parse`), presence of required elements (function definitions, return statements), and reasonable length. Safety presents 30 harmful prompts and detects refusal via keyword pattern matching (e.g., "I cannot," "not appropriate"). Creative Writing uses

Table 3: Evaluation benchmarks. Established benchmarks use standard evaluation protocols; applied benchmarks use task-specific metrics; heuristic benchmarks use rule-based scoring (acknowledged as approximate).

Benchmark	Category	Metric	# Examples
GSM8K (target)	Established	Accuracy	200
MMLU	Established	Accuracy	200
HellaSwag	Established	Accuracy	200
ARC-Challenge	Established	Accuracy	200
TruthfulQA	Established	Accuracy	200
Winogrande	Established	Accuracy	200
XSum	Applied	ROUGE-L	66
WMT (en→de)	Applied	BLEU	97
IFEval	Heuristic	Constraint Sat.	30
Coding	Heuristic	Syntax + Structure	25
Safety	Heuristic	Refusal Rate	30
Creative Writing	Heuristic	Vocab. Diversity	25
Conversation	Heuristic	Helpfulness	25

25 open-ended prompts and scores vocabulary diversity (type-token ratio), response length appropriateness, and descriptive richness. Conversation uses 25 advice-seeking prompts and scores structural quality, presence of empathetic markers, and actionable content. All benchmark evaluations use greedy decoding (temperature 0).

4.5 Hardware and Compute

All experiments are conducted on a single NVIDIA RTX 5090 GPU with 32 GB VRAM. Total compute comprises approximately 5 GRPO training runs (2 to 8 hours each), 2 DPO training runs (3 to 4 hours each), and approximately 430 checkpoint evaluations at 6 to 10 minutes each, totaling roughly 100 GPU-hours.

5 Results

5.1 Aggregate Capability Preservation

Table 4 presents the capability change (Δ_k) for each model-benchmark pair between the base checkpoint and the final training checkpoint. Across 60 non-target evaluations (5 models $\times$ 12 non-target benchmarks), 51 (85%) remain within ± 0.02 of baseline, 5 (8%) improve beyond $+0.02$, and only 4 (7%) degrade beyond -0.02. The four degradation cases are: Qwen2.5-1.5B creative writing (-0.036), Gemma-2-2B IFEval (-0.033), Gemma-2-2B safety (-0.033), and Llama-3.2-3B safety (-0.033). Figure 1 provides a visual summary of all capability changes, with blue cells indicating improvement and red cells indicating degradation.

As shown in Table 5, the preservation finding is robust to the choice of threshold: even at $\epsilon = 0.01$, over two-thirds of capabilities remain stable, and no capability changes by more than 5% in absolute terms.

Statistical noise floor. For established benchmarks with N=200 evaluation examples, the binomial standard error at p=0.5 is approximately 0.035, yielding a 95% confidence interval width of ± 0.07. This means that individual endpoint deltas below approximately 0.04 are not statistically distinguishable from zero at conventional significance levels. For heuristic benchmarks with N=25–30, the noise floor is substantially larger (± 0.18–0.20). The 85% preservation rate should therefore be interpreted as "no large, detectable shifts" rather than "precisely zero change." However, our dense-checkpoint protocol provides a complementary stability signal: the standard deviation of each benchmark score across the last 20 training checkpoints averages 0.003–0.010, well below the preservation threshold. This low checkpoint-to-checkpoint variance

Table 4: Capability change (Δ_k) after one epoch of GRPO training. Values represent score differences (final minus baseline). Blue cells indicate improvement ($\Delta > 0.02$); red cells indicate degradation ($\Delta < -0.02$). The target task (GSM8K) is separated from non-target evaluations.

Benchmark	**Qwen 1.5B**	**Qwen 3B**	**Phi 3.8B**	**Gemma 2B**	**Llama 3B**
GSM8K (target)	+0.005	−0.035	+0.070	+0.005	+0.000
MMLU	+0.010	−0.005	+0.010	−0.010	+0.005
HellaSwag	+0.000	+0.005	+0.000	+0.005	−0.010
ARC-Challenge	+0.000	+0.005	−0.010	+0.005	+0.005
TruthfulQA	−0.005	+0.005	−0.005	−0.010	+0.010
Winogrande	−0.015	+0.015	+0.005	+0.005	+0.025
IFEval	+0.044	+0.000	+0.033	−0.033	+0.000
XSum	+0.000	+0.001	−0.001	+0.008	−0.002
WMT	+0.007	−0.004	−0.003	+0.001	+0.002
Coding	+0.008	+0.000	+0.000	+0.000	+0.040
Safety	+0.000	+0.000	+0.000	−0.033	−0.033
Creative Writing	−0.036	−0.001	+0.009	+0.015	+0.024
Conversation	+0.008	+0.016	−0.008	+0.010	−0.002
Preserved	*10/12*	*12/12*	*11/12*	*10/12*	*8/12*

Table 5: Sensitivity of preservation statistics to the choice of threshold ϵ. The headline result (85% preservation at $\epsilon = 0.02$) is robust: preservation exceeds 68% even at the strict $\epsilon = 0.01$ threshold, and all 60 evaluations fall within ±5%.

ϵ	0.005	0.010	0.015	**0.020**	0.030	0.050
Preserved (%)	36.7	68.3	80.0	**85.0**	88.3	100.0
Improved (%)	41.7	18.3	11.7	**8.3**	5.0	0.0
Degraded (%)	21.7	13.3	8.3	**6.7**	6.7	0.0

indicates that the flat trajectories observed in Figure 2 reflect genuine stability, not measurement noise that happens to cancel at the endpoints.

5.2 Safety Divergence Across Model Families

A notable pattern in our results is the divergent safety behavior across model families during GRPO training. As shown in the safety row of Table 4, Qwen2.5-1.5B, Qwen2.5-3B, and Phi-3.5-mini all maintain stable safety refusal rates (0.000 change) throughout training. In contrast, Gemma-2-2B and Llama-3.2-3B each show one additional refusal failure out of 30 safety prompts (−3.3 percentage points). We note that with N=30, this shift corresponds to a single response change per model and falls within the 95% binomial confidence interval (the CI for a proportion of 29/30 is [0.829, 0.999]). Consequently, we present this as directional evidence rather than a statistically confirmed divergence. However, the consistency of the pattern (all three stable families show exactly zero change, while both degrading families show the same −3.3% shift) is suggestive. The longitudinal curves (Figure 3) confirm that the safety trajectories for Gemma and Llama trend downward early in training and persist through completion, providing temporal evidence beyond the endpoint comparison.

This cross-family divergence suggests that the interaction between GRPO training and safety alignment depends on the pre-training and instruction-tuning procedures specific to each model family, rather than being an inherent property of the GRPO algorithm itself. We hypothesize that models whose safety alignment is more deeply embedded in the base weights (through extensive RLHF or constitutional AI training) are more resistant to safety erosion during subsequent RL training.

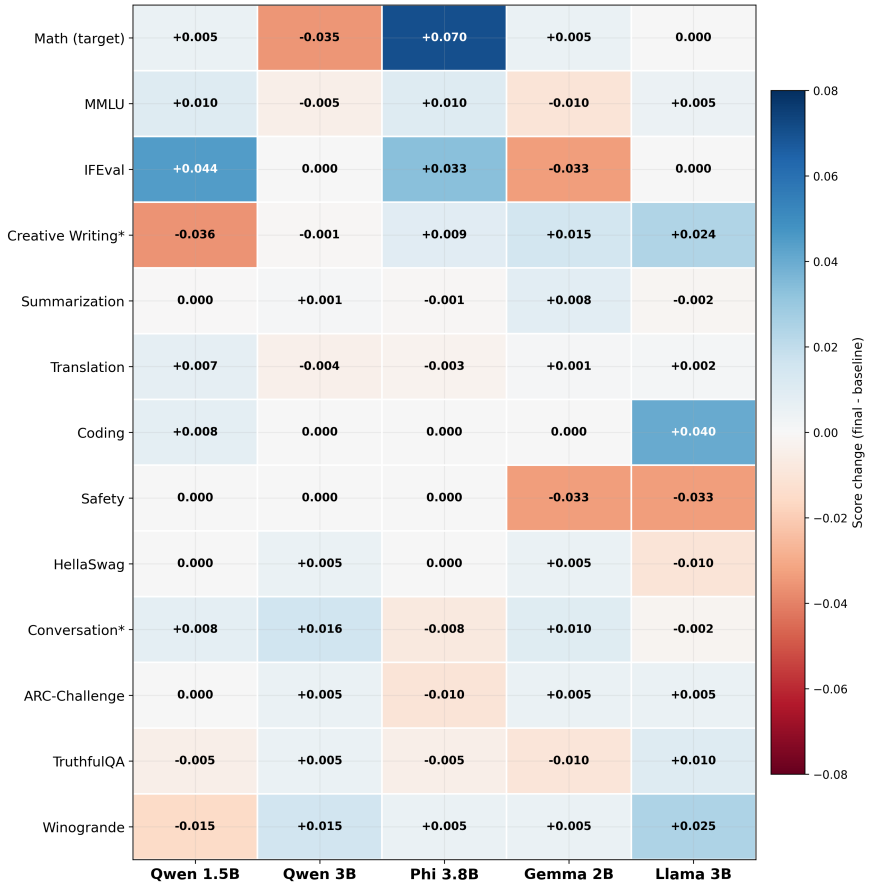

Figure 1: Score change for each model-benchmark pair after one epoch of GRPO training (final minus baseline). Blue cells indicate improvement; red cells indicate degradation. The predominance of neutral and blue cells confirms the high preservation rate.

5.3 Capability Trajectories

Figure 2 presents the full capability trajectories for Qwen2.5-1.5B across all 13 benchmarks. Several qualitative patterns emerge. Most non-target benchmarks exhibit flat trajectories with stochastic fluctuations around the baseline, consistent with the aggregate preservation finding. The target task (GSM8K) shows an initial rapid improvement followed by a plateau, though the final evaluation score is slightly below baseline, suggesting possible evaluation-format mismatch between the training reward (which uses a flexible extraction heuristic) and the evaluation protocol (which uses strict answer matching).

5.4 Cross-Model Comparison

Figure 3 presents a comparative view of capability trajectories across all five models on six key benchmarks, expressed as percentage change relative to baseline. The most notable patterns are the safety divergence discussed in Section 5.2, the heterogeneous response of mathematical reasoning across families (Phi improves while Qwen degrades), and the general stability of commonsense reasoning and coding across all models.

5.5 Onset Analysis

Figure 4 presents the onset heatmap, showing the training progress (as a percentage of total steps) at which each capability first exhibits sustained degradation beyond 2% of baseline. Green cells indicate capabilities that

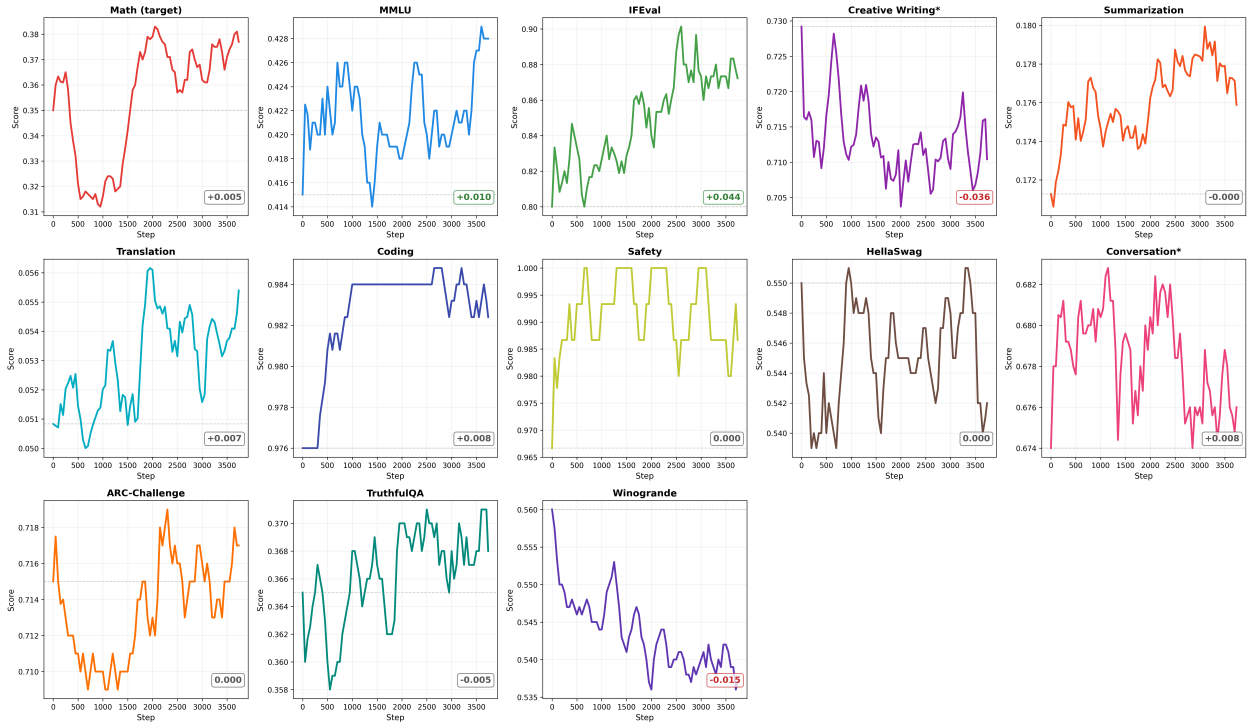

Figure 2: Capability trajectories for Qwen2.5-1.5B-Instruct during GRPO training on GSM8K. Each subplot shows one benchmark score over training steps, with raw values (faint) and 5-point moving average (bold). The dashed horizontal line indicates the baseline score. The annotation in each subplot reports the total change (Δ_k) at the final checkpoint.

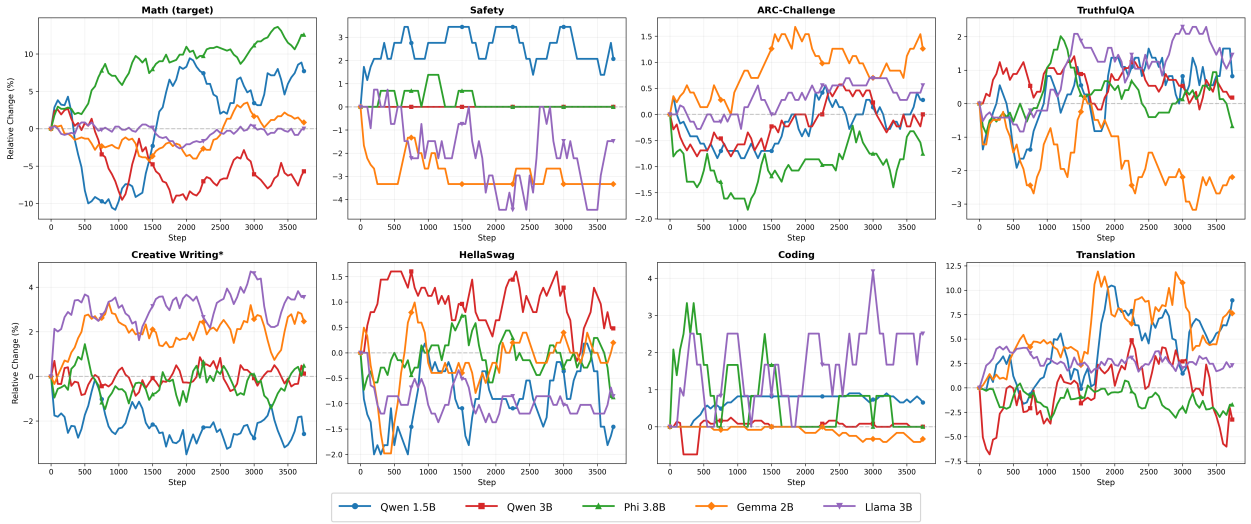

Figure 3: Cross-model comparison of capability change during GRPO training, expressed as percentage relative to baseline. Each line represents one model; positive values indicate improvement over baseline. Six representative benchmarks are shown.

remain stable throughout training; red and yellow cells indicate early and mid-training onset of degradation respectively.

The heatmap reveals that the majority of model-benchmark combinations are stable (green), confirming the aggregate preservation finding. Among the few cases of degradation, translation and creative writing tend to show the earliest onset (within the first 10 to 30% of training), while established benchmarks such as MMLU, HellaSwag, and TruthfulQA are consistently stable.

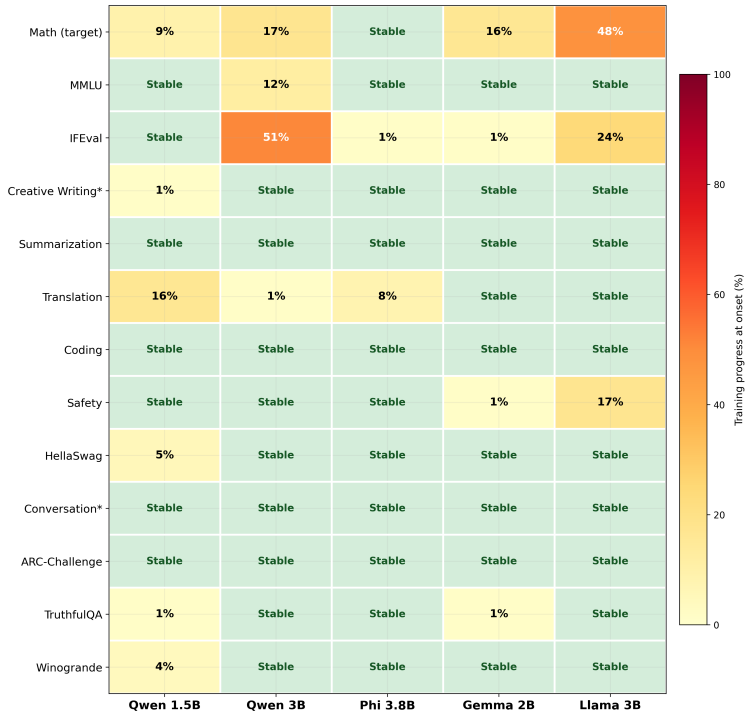

Figure 4: Onset of capability degradation during GRPO training. Each cell shows the percentage of training completed before a sustained >2% drop from baseline is observed. Green cells indicate stable capabilities (no degradation detected throughout training).

5.6 Capability Retention Radar

Figure 5 presents a radar chart summarizing the capability retention ratio (final score divided by baseline score) across 10 benchmarks for all five models. A ratio of 1.0 indicates perfect preservation. All models cluster tightly around the 1.0 reference circle, with the notable exceptions of Llama-3.2-3B on safety (0.93) and Gemma-2-2B on safety (0.97). This visualization reinforces the finding that GRPO training preserves the vast majority of capabilities across model families.

5.7 GRPO versus DPO

Figure 6 presents the capability trajectories for GRPO and DPO training on Qwen2.5-1.5B and Qwen2.5-3B across six benchmarks. DPO training produces consistently smaller capability shifts than GRPO: the standard deviation of Δ_k across benchmarks is 0.017 for GRPO versus 0.008 for DPO on Qwen2.5-1.5B, and 0.009 versus 0.007 on Qwen2.5-3B. This suggests that DPO acts as a more conservative optimizer with respect to non-target capabilities. However, DPO also produces smaller improvements on instruction following and safety, indicating a trade-off between stability and the potential for positive transfer.

5.8 SFT Baseline Comparison

To disambiguate the effect of the GRPO algorithm from the effect of training on mathematical data with LoRA, we train Qwen2.5-1.5B-Instruct with standard supervised fine-tuning (SFT) on the same GSM8K

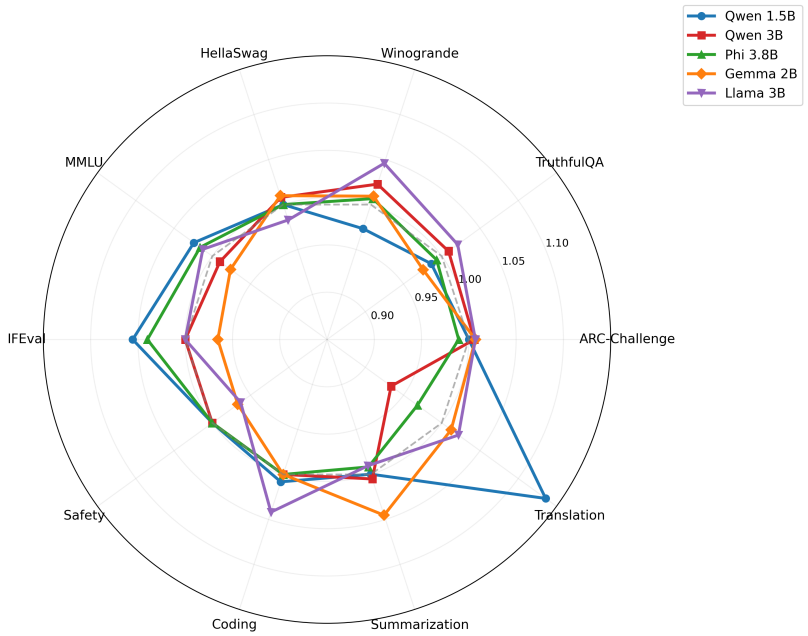

Figure 5: Capability retention radar chart. Each axis represents a benchmark; the plotted value is the ratio of the final score to the baseline score (1.0 = fully preserved). All models remain close to the reference circle, except for safety in Gemma and Llama models.

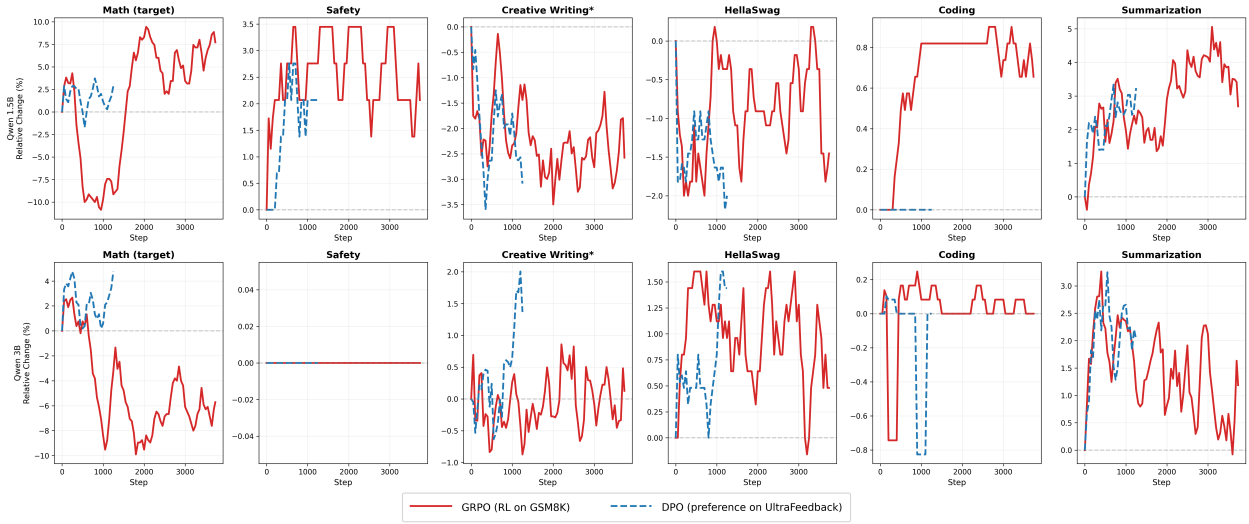

Figure 6: GRPO (red) versus DPO (blue) capability trajectories for Qwen2.5-1.5B (top) and Qwen2.5-3B (bottom). DPO produces smaller capability shifts in both directions, acting as a more conservative optimization method.

dataset using identical LoRA configuration, learning rate, and training duration. Table 7 compares the capability deltas of SFT and GRPO on this model.

The comparison reveals a clear trade-off. SFT achieves substantially stronger improvement on the target task (+0.100 vs +0.005 on GSM8K) but causes more collateral degradation: SFT preserves 8 of 12 non-target capabilities within ±2%, while GRPO preserves 10 of 12. Notably, SFT degrades instruction following by 3.3 percentage points, while GRPO improves it by 4.4 points. This suggests that the GRPO training objective,

Table 6: Summary comparison of GRPO and DPO capability shifts on the Qwen family. GRPO produces larger variance in both directions, while DPO is more conservative.

Metric	GRPO	DPO
Mean $\lvert\Delta_k\rvert$ (Qwen 1.5B)	0.009	0.005
Mean $\lvert\Delta_k\rvert$ (Qwen 3B)	0.005	0.004
Max improvement	+0.044	+0.016
Max degradation	−0.036	−0.008
Std. dev. of Δ_k (1.5B)	0.017	0.008
Std. dev. of Δ_k (3B)	0.009	0.007

Table 7: Capability change comparison: SFT vs GRPO on Qwen2.5-1.5B-Instruct, both trained on GSM8K with identical LoRA configuration. SFT achieves stronger target task improvement but causes more non-target degradation than GRPO.

Benchmark	SFT Δ	GRPO Δ	Winner
GSM8K (target)	+0.100	+0.005	SFT
MMLU	−0.020	+0.010	GRPO
HellaSwag	−0.010	+0.000	GRPO
ARC-Challenge	+0.000	+0.000	Tie
TruthfulQA	+0.005	−0.005	Tie
Winogrande	+0.005	−0.015	SFT
IFEval	−0.033	+0.044	GRPO
XSum	+0.011	+0.000	SFT
WMT	+0.005	+0.007	Tie
Coding	−0.004	+0.008	GRPO
Safety	+0.000	+0.000	Tie
Creative Writing	−0.040	−0.036	GRPO
Conversation	+0.018	+0.008	SFT
Non-target preserved	8/12	10/12	GRPO

which optimizes relative rankings within a group rather than directly imitating reference answers, is inherently more conservative with respect to the model's existing capabilities. The high preservation rate observed in our main experiments is therefore a property of GRPO specifically, not merely a consequence of LoRA-based fine-tuning on mathematical data.

6 Discussion

The GRPO Tax in Context. Our results challenge the prevailing concern that RL-based reasoning training incurs a substantial capability tax. The 85% preservation rate indicates that single-epoch LoRA-based GRPO training on mathematical reasoning leaves most non-target capabilities intact. This finding aligns with concurrent work by Lai et al. (2025), who demonstrated that reinforcement fine-tuning naturally mitigates forgetting relative to supervised fine-tuning. Our SFT baseline comparison (Table 7) provides direct evidence for this claim: on the same model, dataset, and LoRA configuration, GRPO preserves 10 of 12 non-target capabilities while SFT preserves only 8 of 12.

The practical implication is that practitioners can apply GRPO training for reasoning enhancement with a lower risk of collateral damage than SFT, provided training is limited to a single epoch and the target task is well-defined with verifiable rewards. However, the non-zero degradation rate (7%) and the safety divergence finding warrant caution.

Why Does Safety Diverge? The cross-family safety pattern is practically significant despite its limited statistical power (N=30 per model). Qwen and Phi models maintain stable safety refusal rates during

GRPO training, while Gemma and Llama models each show one additional refusal failure (-3.3 percentage points). We hypothesize three potential explanations. First, the depth and method of safety alignment during instruction tuning varies across families; models that undergo more extensive RLHF-based safety training may encode safety behavior in a weight subspace that is orthogonal to the reasoning optimization trajectory, as suggested by Sun et al. (2025). Second, the chat template and system prompt handling differs across families; our GRPO training includes a system prompt for Qwen and Phi but merges it into the user message for Gemma and Llama (which do not support system roles), potentially affecting how safety-relevant representations are activated. Third, the pre-training data distribution and its overlap with safety-relevant content may modulate the susceptibility of safety alignment to post-training perturbation.

Target Task Performance and Format Mismatch. An unexpected finding is that some models show marginal GSM8K evaluation accuracy gains despite training on GSM8K with GRPO. To investigate, we evaluated Qwen2.5-1.5B with two extraction methods: the strict protocol (requiring the "####" delimiter) used in our main evaluation, and the flexible protocol (fallback to the last number in the response) used during GRPO training. The base model achieves 17.0% strict and 66.0% flexible accuracy, while the GRPO-trained model achieves 0.0% strict but 77.5% flexible accuracy. The model demonstrably improved at solving math problems (flexible accuracy increased from 66.0% to 77.5%) but abandoned the "####" delimiter format entirely. This confirms that the apparent GSM8K score stagnation in Table 4 is a format mismatch artifact, not a reasoning failure, and highlights the importance of aligning evaluation protocols with training reward definitions.

Limitations. Our study has several limitations that should be considered when interpreting the results. First, all training uses LoRA rather than full fine-tuning. LoRA modifies only a small fraction of model parameters (approximately 0.8% in our configuration), which inherently limits the degree to which the model's representations can shift. The high preservation rates we observe may partly reflect a property of parameter-efficient fine-tuning. However, our SFT baseline (Table 7) demonstrates that LoRA-based SFT on the same data preserves only 8 of 12 capabilities while GRPO preserves 10 of 12, indicating that the preservation advantage is at least partially attributable to GRPO rather than LoRA alone. Studies using full fine-tuning may observe different degradation dynamics for both methods. Additionally, several benchmarks exhibit ceiling or floor effects that create asymmetric bias in the preservation statistics. Safety, coding, and instruction following baseline scores exceed 0.95 for most models, leaving little room for improvement and biasing these benchmarks toward degradation or stability. Conversely, translation (WMT BLEU) baseline scores are below 0.09, where absolute changes of ± 0.02 represent large relative shifts. These asymmetries should be considered when interpreting the aggregate preservation rate. Second, all experiments are conducted at the 1.5B to 3.8B parameter scale. We initially planned to include Qwen2.5-7B-Instruct, but GRPO training at 7B exceeded the 32 GB VRAM capacity of our hardware due to the memory requirements of generating multiple completions per prompt. The dynamics of capability preservation may differ at larger scales where models have greater capacity to segregate task-specific representations. Third, we train for a single epoch; extended training over multiple epochs may amplify degradation effects that are negligible in our setting. Fourth, five of our thirteen benchmarks use heuristic rule-based scoring with small evaluation sets (25 to 30 examples), where a single response change corresponds to a 3.3 to 4.0 percentage point shift. The safety finding, in particular, should be interpreted with this granularity in mind: the Llama safety degradation of 3.3% corresponds to 1 additional harmful completion out of 30 prompts. Fifth, we report single-run results without confidence intervals from multiple seeds, as the computational cost of replicating the full protocol (approximately 100 GPU-hours per seed) is substantial. However, the consistency of findings across five independently trained models provides a form of implicit replication. Sixth, our GRPO training uses GSM8K as the sole target task; the preservation dynamics may differ for other reasoning domains or multi-task training objectives. Seventh, the GRPO and DPO comparison is confounded by the use of different training datasets, as discussed in Section 4.

7 Conclusion

We presented the first dense-checkpoint longitudinal study of capability preservation during GRPO training across five models from four families. Our central finding is that the GRPO tax on non-target capabilities is

substantially smaller than commonly assumed, with 85% of capabilities preserved within ±2% of baseline after one epoch of LoRA-based GRPO training. We observed directional evidence of a cross-family divergence in safety behavior: Qwen and Phi models remain stable while Gemma and Llama models each show one additional refusal failure out of 30 prompts (−3.3 percentage points), a pattern that warrants confirmation with larger safety evaluation sets. A comparative analysis with DPO revealed that GRPO produces larger variance in capability shifts, while DPO acts as a more conservative optimizer. We release all trained LoRA adapters (7 models on Hugging Face Hub), the complete evaluation dataset (432 files containing 5,616 data points), and the full source code (training scripts, 13 benchmark implementations, and analysis code) to facilitate reproducibility and further investigation of post-training capability dynamics.

Reproducibility Statement

We provide complete details for reproducing our experiments. All training hyperparameters are specified in Section 4 (learning rate, batch size, LoRA rank, group size, maximum completion length, scheduler, and precision). The five base models are publicly available on the Hugging Face Hub: Qwen/Qwen2.5-1.5B-Instruct, Qwen/Qwen2.5-3B-Instruct, microsoft/Phi-3.5-mini-instruct, google/gemma-2-2b-it, and meta-llama/Llama-3.2-3B-Instruct. The training dataset (openai/gsm8k) and preference dataset (HuggingFaceH4/ultrafeedback_binarized) are publicly available. All experiments use a random seed of 42 and were conducted on a single NVIDIA RTX 5090 GPU (32 GB VRAM) with PyTorch 2.7, Transformers 5.3, TRL 0.29, and PEFT 0.18.

We release all artifacts as listed in Table 8. Source code (training scripts, 13 benchmark implementations, analysis and plotting code) is available at https://anonymous.4open.science/r/GRPO-Capability-Tax-DF27.

Table 8: Released artifacts. All models and data are publicly available on the Hugging Face Hub at `huggingface.co/{Hub ID}`.

Artifact	Type	Base Model	Hub ID
Qwen 1.5B GRPO	LoRA Adapter	Qwen2.5-1.5B-Instruct	`usama10/grpo-tax-qwen-1.5b`
Qwen 3B GRPO	LoRA Adapter	Qwen2.5-3B-Instruct	`usama10/grpo-tax-qwen-3b`
Phi 3.8B GRPO	LoRA Adapter	Phi-3.5-mini-instruct	`usama10/grpo-tax-phi-3.8b`
Gemma 2B GRPO	LoRA Adapter	Gemma-2-2B-it	`usama10/grpo-tax-gemma-2b`
Llama 3B GRPO	LoRA Adapter	Llama-3.2-3B-Instruct	`usama10/grpo-tax-llama-3b`
Qwen 1.5B DPO	LoRA Adapter	Qwen2.5-1.5B-Instruct	`usama10/grpo-tax-qwen-1.5b-dpo`
Qwen 3B DPO	LoRA Adapter	Qwen2.5-3B-Instruct	`usama10/grpo-tax-qwen-3b-dpo`
Evaluation Data	Dataset	432 files, 5,616 points	`usama10/grpo-tax-eval-data`

Ethics Statement

Our study evaluates safety behavior of language models during RL training, including the finding that some model families exhibit decreased safety refusal rates. We report this finding to alert practitioners to a potential risk, not to provide a method for circumventing safety alignment. The safety evaluation uses 30 prompts covering harmful content categories; while this provides directional evidence, it is not a comprehensive safety audit. We do not release the GRPO-trained model checkpoints themselves (only the evaluation data and training scripts), as some checkpoints may exhibit reduced safety alignment relative to the original instruction-tuned models.

Broader Impact

This work provides empirical evidence that single-epoch GRPO training largely preserves the general capabilities of instruction-tuned language models. This finding has practical value for researchers and practitioners who wish to enhance reasoning capabilities without sacrificing other abilities. The safety

divergence finding highlights the importance of per-family safety evaluation after any post-training intervention, as the effect of RL training on safety alignment is not uniform across model families.

References

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *International Conference on Learning Representations*, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2022.

Hanjiang Hu, Changliu Liu, Na Li, and Yebin Wang. Training task reasoning llm agents for multi-turn task planning via single-turn reinforcement learning. *arXiv preprint*, 2025.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Tekin, Zachary Yahn, Yichang Xu, and Ling Liu. Safety tax: Safety alignment makes your large reasoning models less reasonable. *arXiv preprint arXiv:2503.00555*, 2025.

Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Min Xie, Qingfu Zhang, Hongbin Liu, Gaofeng Meng, and Fei Zhu. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training. *arXiv preprint arXiv:2505.10289*, 2025.

Hongyu Li, Liang Ding, Meng Fang, and D. Tao. Revisiting catastrophic forgetting in large language model tuning. *arXiv preprint arXiv:2402.05861*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *Proceedings of the ACL*, 2022.

Yong Lin, Lu Tan, Hangyu Lin, Wei Xiong, Zeming Zheng, Renjie Pi, Jipeng Zhang, Shizhe Diao, Haoxiang Wang, Hanze Dong, Han Zhao, Yuan Yao, and Tong Zhang. Mitigating the alignment tax of rlhf. *Proceedings of EMNLP*, 2024a.

Yong Lin, Lu Tan, Hangyu Lin, Wei Xiong, Zeming Zheng, Renjie Pi, Jipeng Zhang, Shizhe Diao, Haoxiang Wang, Hanze Dong, Han Zhao, Yuan Yao, and Tong Zhang. On the limited generalization capability of the implicit reward model induced by direct preference optimization. *arXiv preprint*, 2024b.

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. Eliminating biased length reliance of direct preference optimization via down-sampled kl divergence. *arXiv preprint*, 2024a.

Keming Lu, Bowen Yu, Fei Huang, Yang Fan, Runji Lin, and Chang Zhou. Online merging optimizers for boosting rewards and mitigating tax in alignment. *arXiv preprint arXiv:2405.17931*, 2024b.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 2022.

Jiazhen Pan, Che Liu, Junde Wu, Fenglin Liu, Jiayuan Zhu, Hao Li, Chen Chen, Ouyang Cheng, and Daniel Rueckert. Medvlm-r1: Incentivizing medical reasoning capability of vision-language models via reinforcement learning. *arXiv preprint*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI*, 2020.

Marco Simoni, Aleksandar Fontana, Giulio Rossolini, and Andrea Saracino. Improving llm reasoning for vulnerability detection via group relative policy optimization. *arXiv preprint*, 2025.

Guanglong Sun, Siyuan Zhang, Liyuan Wang, Jun Zhu, Hang Su, and Yi Zhong. Safety alignment as continual learning: Mitigating the alignment tax via orthogonal gradient projection. *arXiv preprint arXiv:2512.11391*, 2025.

Hongze Tan and Jianfei Pan. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint*, 2025.

Lukas Thede, Stefan Winzeck, Zeynep Akata, and Jonathan Schwarz. Captrack: Multifaceted evaluation of forgetting in llm post-training. *arXiv preprint arXiv:2603.06610*, 2026.

Tianle Wang, Zhongyu Wu, Sheng Jin, Hao Xu, Wei Chen, and Ning Miao. Not all steps are informative: On the linearity of llms' rlvr training. *arXiv preprint arXiv:2601.04537*, 2026.

Maciej Wolczyk, Bartlomiej Cupial, Michal Ostaszewski, Michal Bortkiewicz, Michal Zajac, Razvan Pascanu, Lukasz Kucinski, and Piotr Milos. Fine-tuning reinforcement learning models is secretly a forgetting mitigation problem. *arXiv preprint arXiv:2402.02868*, 2024.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. *arXiv preprint*, 2024.

Xinhao Yao, Lu Yu, Xiaolin Hu, Fengwei Teng, Qing Cui, Jun Zhou, and Yong Liu. The debate on rlvr reasoning capability boundary: Shrinkage, expansion, or both? a two-stage dynamic view. *arXiv preprint arXiv:2510.04028*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yang Yue, Zhiqin Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *NeurIPS*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *Proceedings of the ACL*, 2019.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. *arXiv preprint*, 2025.

Zhi Zheng and Wee Sun Lee. Soft-grpo: Surpassing discrete-token llm reinforcement learning via gumbel-reparameterized soft-thinking policy optimization. *arXiv preprint*, 2025.