TRACING REPETITIVE DEGENERATION TO THE POST-TRAINING PIPELINE

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have achieved strong performance across diverse tasks. However, LLMs occasionally suffer from output repetition, which degrades output quality and wastes computational resources. This issue is widely observed, but difficult to reliably reproduce, leaving its underlying mechanisms underexplored. In this work, we present a controlled empirical study of how post-training choices are associated with this phenomenon. Firstly, to enable controlled analysis, we present the Self-Reinforced Loop Approach (*SRLA*), which amplifies output repetition and makes this otherwise sporadic phenomenon reproducible. Using SRLA, we find that models trained with recent reward-based post-training (*e.g.*, GRPO) are more susceptible to repetition than conventional ones. This observation motivates our intuition that output repetition is closely associated with post-training choices, which we validate through cross-model comparisons. A weaker but consistent increase remains observable in models trained on GRPO-distilled data. Analyzing token-level distributions, we identify three crucial factors, including selected-token lock-in, insufficient candidate token diversity, and weak EOS activation. Motivated by these findings, we propose two mitigations that significantly reduce failure rates by an average of 86%. These findings bridge post-training and repetitive generation behaviors, and provide practical guidance for developing more robust models.

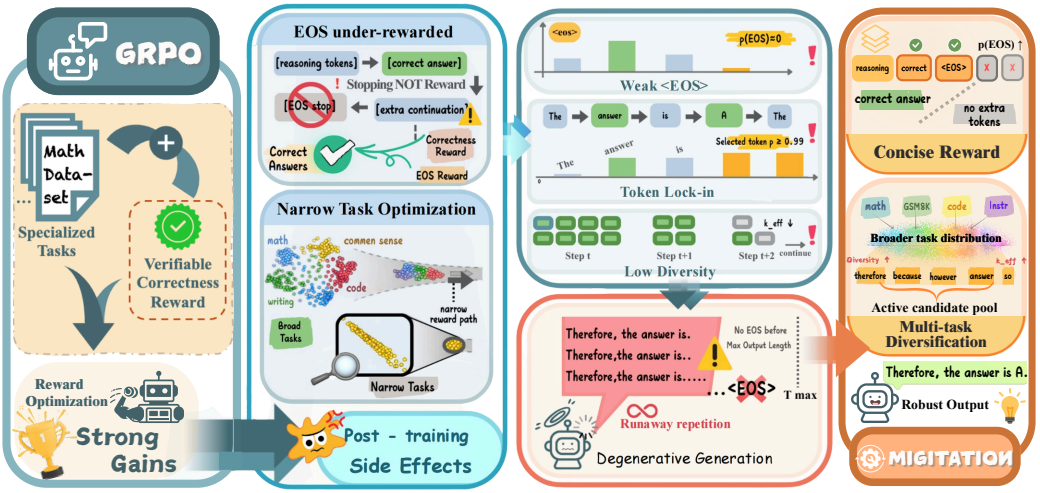


Figure 1: **Overview of repetitive degeneration after GRPO-based post-training.** Weak termination incentives and narrow-task optimization are associated with token lock-in, reduced candidate diversity, and weak EOS activation. Conciseness rewards and multi-task training target these failure modes.

1 INTRODUCTION

Large Language Models (LLMs) have been deployed across a range of applications, from multi-turn dialogue and question answering (Li et al., 2024; Xiong et al., 2024) and document summarization (Tian et al., 2024; Hu et al., 2024) to code generation (Wu et al., 2024; Han et al., 2024) and multi-step

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

reasoning (Servantez et al., 2024; Jiang et al., 2024). With the scaling of deployments, unpredictable repetition in LLMs has attracted growing attention due to its impact on the quality of generated text. Prior studies have shown that repetition can be induced under a range of explicit elicitation settings, including instruction-following exploits (Nasr et al., 2023; Bai et al., 2024) and optimization-based adversarial attacks (Li et al., 2026; Dong et al., 2025; Geiping et al., 2024).

However, these settings mainly study deliberately induced repetition and do not capture the sporadic repetition failures that arise during benign, non-adversarial use. Additionally, as post-training methods such as supervised fine-tuning (SFT) (Ouyang et al., 2022), Direct Preference Optimization (DPO) (Rafailov et al., 2023; Wang et al., 2024a; Shi et al., 2024), and Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Zhang & Zuo, 2025; Barta et al., 2025) have become increasingly adopted, repetition has also been noted in recent model reports, including Llama (Grattafiori et al., 2024), Qwen (Qwen et al., 2025; Yang et al., 2025), and DeepSeek (Bi et al., 2024). Related mechanism analyses have attributed repetition to generation-level factors such as decoding strategies (Zhu et al., 2024; Nguyen et al., 2025), context-driven token behaviors such as in-context copying (Yan et al., 2024; Mahaut & Franzon, 2025), and internal model components such as repetition neurons and attention-sink dynamics (Wang et al., 2024b; Yona et al., 2025). These studies provide post-hoc accounts of how repetition manifests once it occurs, but they do not identify what gives rise to such failures in the first place.

To bridge these gaps, we first introduce the Self-Reinforced Loop Approach (SRLA), a prompt-construction method that exploits self-reinforcing dynamics, which are the model's repetitive distributions under benign instructions. With SRLA, the repetitive phenomenon can be reproducibly elicited in a controlled, non-adversarial stress-test setting, facilitating further analysis. Then we assess models from the DeepSeek (Liu et al., 2024), Llama (Grattafiori et al., 2024), and Qwen (Qwen et al., 2025) families, including official releases, community-released models, and models trained in this work. The results reveal that different post-training models exhibit varying degrees of repetitive degradation, among which GRPO-based post-training is the most susceptible. While SFT-only models remain stable and provide a consistent baseline, introducing GRPO markedly increases repetition, with failure rates reaching **100%** $\uparrow$ in the most severe settings. The vulnerability also persists under distillation, with GRPO-distilled data transferring the degeneration tendency.

We therefore conduct a token-level analysis to investigate how post-training alters generation dynamics, finding selected-token lock-in, reduced candidate diversity, and weak EOS activation. Considering that GRPO improves reasoning through optimization against length-agnostic verifiable rewards with a group-relative advantage estimator, we propose two targeted mitigations for the degradation associated with this training setup. The first mitigation introduces a conciseness reward to explicitly align response termination with task completion. By encouraging the model to stop once the task is complete, this reward increases p_{eos} by $158\% \uparrow$ and reduces repetition failure rates by **81.10%** $\downarrow$. To further counteract the narrowing effect of task-specific optimization, the second method we adopt is multi-task diversification. Instead of training on a narrow task, we rebalance the training composition across four datasets covering three task types, which increases the average candidate-set size by $22.5\% \uparrow$ and reduces repetition failure rates by **91.46%** $\downarrow$.

Our work makes three main contributions. First, we introduce SRLA, a prompt-construction method that makes otherwise sporadic repetition failures reproducible for a controlled study under a fixed, non-adversarial stress-test condition. Second, we present a focused empirical study of GRPO-related repetition across Llama and Qwen models under a shared elicitation setting, finding greater susceptibility after GRPO post-training and a weaker but consistent increase after training on GRPO-distilled data. Third, we characterize the failure pattern at the token level and propose two targeted mitigations, conciseness reward and multi-task diversification, which reduce failure rates by 81.10% and 91.46%, respectively. These findings suggest that post-training can exacerbate repetitive degeneration, and offer concrete guidance for designing more robust post-training pipelines.

2 Related Work

Repetition is a widely observed degeneration mode in LLMs, manifesting across model scales, decoding settings, and downstream tasks Holtzman et al. (2019); Li et al. (2023); Ivgi et al. (2024); Mahaut & Franzon (2025). More broadly, repetitive loops long predate current post-training pipelines, and earlier neural generation work linked degeneration to both decoding choices and likelihood-based

training, motivating nucleus sampling and unlikelihood training Holtzman et al. (2019); Welleck et al. (2019). Within modern LLMs, recent studies analyze repetition from several complementary angles: decoding-time methods for managing generation degeneration Zhu et al. (2024); Nguyen et al. (2025), reinforcement effects arising from repeated patterns in training data or context Yan et al. (2024); Li et al. (2023), distinct mechanisms underlying naturally occurring and in-context-induced repetition Mahaut & Franzon (2025), and fallback behavior under epistemic uncertainty Ivgi et al. (2024). Repetition or abnormally long generation can also be deliberately induced under adversarial settings for training-data extraction or denial-of-service Nasr et al. (2023); Bai et al. (2024); Geiping et al. (2024); Li et al. (2026); Dong et al. (2025). Our focus is different: we study sporadic repetitive degeneration that already occurs in benign use through non-adversarial stress testing, because its sporadic occurrence makes systematic comparisons across models and training stages difficult without controlled elicitation.

Mechanism Analyses and Post-Training Effects. Beyond surface-level descriptions, several recent studies examine repetition from an internal or post-training perspective. At the component level, repetition has been traced to repetition neurons Wang et al. (2024b); Hiraoka & Inui (2025). Related work studies attention-sink dynamics and their emergence Gu et al. (2025), while repeated-token divergence has been linked to disruptions in an attention-sink circuit Yona et al. (2025). Mechanistic interpretability work further identifies repetition-related features for token-level and paragraph-level repetition through SAE-based activation analysis Yao et al. (2025). Complementary analyses examine how fine-tuning, RLHF, and instruction formatting affect output diversity or mode collapse O'Mahony et al. (2024); Kirk et al. (2024); Yun et al. (2025). Together, these studies characterize internal repetition mechanisms and broader diversity changes during alignment, but they do not provide a controlled comparison of repetitive-degeneration susceptibility across models trained with SFT, GRPO, or GRPO-distilled data.

3 SRLA FOR REPETITIVE DEGENERATION

In this section, we first introduce the Self-Reinforced Loop Approach (SRLA), a prompt-construction method for eliciting self-reinforcing repetitive continuations, and describe the bilingual evaluation set produced by this procedure. Building on SRLA, we conduct a token-level analysis of model outputs to examine how prompts affect generation dynamics under controlled decoding.

3.1 SELF-REINFORCED LOOP APPROACH (SRLA)

Prior work shows that repeated substrings can induce pattern continuation, with the probability of continuation increasing monotonically with repetition count (Yan et al., 2024; Mahaut & Franzon, 2025). Repetitive continuations typically receive low perplexity, suggesting that such outputs may remain likely under the model distribution rather than solely determined by decoding (Holtzman et al., 2019; Welleck et al., 2019; Fu et al., 2021). Inspired by these observations, we hypothesize that there are inherent self-reinforcing continuation dynamics in the LLMs, which can be exploited to elicit repetitive patterns. We operationalize this hypothesis through SRLA, which reincorporates a short repetitive seed elicited from the seed model's response into an otherwise task-formatted base prompt, producing a prompt that is not optimized for any particular target model and thus enables controlled comparison across models. Specifically, we start with a seed model π_θ. Let $p_\theta(x_t \mid x_{<t})$ denote the native next-token distribution of π_θ at autoregressive step t, with corresponding logits ℓ_t. We define the controlled decoding distribution $\tilde{p}_\theta(x_t \mid x_{<t})$ by excluding EOS from the support of $p_\theta(\cdot \mid x_{<t})$, applying the temperature τ scaling to the remaining logits, and renormalizing over the non-EOS vocabulary:

$$\tilde{p}_\theta(v \mid x_{<t}) = \frac{\exp(\ell_t(v)/\tau)\mathbf{1}[v \neq \mathrm{EOS}]}{\sum_{v' \in \mathcal{V}} \exp(\ell_t(v')/\tau)\mathbf{1}[v' \neq \mathrm{EOS}]}.$$

Then, we elicit the repetitive continuation from π_θ through the above controlled decoding conditioned on base prompt x. For each base prompt x, we sample a continuation $z = (z_1, \ldots, z_m)$ from $\tilde{p}_\theta$ until a repetitive segment is detected. We extract the detected repetitive seed r_x from z and append it to the corresponding input x, forming $x_{\mathrm{SRLA}} = x \oplus r_x$. This construction preserves the original task context while exposing the model to a locally repetitive suffix, where $\oplus$ denotes string concatenation under the original prompt template.

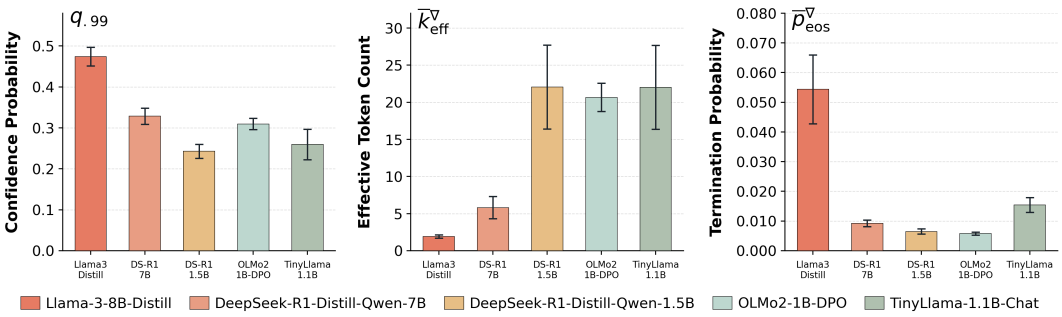


Figure 2: Token-level statistics on D_{SRLA} for five models with intermediate RDR: selected-token lock-in $q_{.99}$, effective candidate-set size k^R_{eff}, and EOS activation p^R_{eos}. Error bars: prompt variability.

Repeating this procedure over all $x \in \mathcal{D}_{\text{base}}$ yields the SRLA evaluation set $\mathcal{D}_{\text{SRLA}} = \{x \oplus r_x : x \in \mathcal{D}_{\text{base}}, r_x \text{ is detected}\}$. Since the repetitive seeds are extracted with an auxiliary controlled decoder, we further check whether the resulting SRLA prompts remain compatible with the native distributions of the evaluated models. The perplexity validation in Appendix A shows that $\mathcal{D}_{\text{SRLA}}$ is not merely an artifact of the seed-extraction procedure. We next describe the concrete instantiation of SRLA and evaluate model behavior on the resulting benchmark.

3.2 Evaluation on the SRLA Evaluation Set

We instantiate SRLA with Qwen2.5-7B (Qwen et al., 2025) as the seed model π_θ. We then generate controlled-decoding sequences over the base-prompt dataset $\mathcal{D}_{\text{base}}$, which is designed to mitigate language-specific confounds, with a detailed analysis provided in Appendix A. The base prompts $\mathcal{D}_{\text{base}}$ are formulated in a multiple-choice format. For seed extraction, each base prompt is fed into π_θ with temperature $\tau = 0.5$, and the generated sequence is truncated at $T_{\max} = 512$. After detecting repetitive segments from these trajectories, we append the extracted seeds to their corresponding base prompts to form $\mathcal{D}_{\text{SRLA}}$, which introduces a locally self-reinforcing suffix.

We feed the evaluation set $\mathcal{D}_{\text{SRLA}}$ into each model and evaluate its outputs under another fixed decoding configuration: temperature $\tau_{\text{dec}} = 0.7$, top-p sampling with threshold $p_{\text{top}} = 0.9$, and maximum generation length $T_{\max} = 4096$. To reduce stochastic variation, we performed vanilla decoding $N_g = 3$ times. We evaluate each model using the repetitive degeneration rate (RDR), defined as:

$$\text{RDR} = \frac{1}{|\mathcal{D}_{\text{SRLA}}|} \sum_{i=1}^{|\mathcal{D}_{\text{SRLA}}|} \mathbb{I}(x_i), x_i \in \mathcal{D}_{\text{SRLA}},$$

where $\mathbb{I}$ is a designed indicator of repetitive degeneration. Detailed criteria about $\mathbb{I}$ are provided in Appendix B. Table 1 reports RDR for several models across the $\mathcal{D}_{\text{SRLA}}$, appendix B.2 for a comprehensive comparison with the baseline question group. These results show that the same SRLA construction elicits heterogeneous response strengths across evaluated models, ranging from negligible degeneration to severe repetitive non-termination. This response heterogeneity motivates the token-level characterization in the next section, where we examine the distributional profile associated with stronger SRLA responses.

3.3 Token-Level Characterization of Repetitive Degeneration

SRLA exposes repetitive degeneration at the output level, but the surface failure alone does not explain the underlying mechanism. We explore this through three sequence-level measures from the model's native next-token distribution: $q_{.99}$ for *Selected-token Lock-in*, $\overline{k}^{\mathcal{R}}_{\text{eff}}$ for *Low Candidate Diversity*, and $\overline{p}^{\mathcal{R}}_{\text{eos}}$ for *Weak EOS Activation*.

Given an SRLA prompt x and generated sequence $y_{1:T}$, let $\mathcal{R}$ denote the *Runaway* window, instantiated as the final 25% of decoding steps.

Selected-token Lock-in. We measure whether generation follows near-deterministic continuations under the native next-token distribution. Higher $q_{.99}$ indicates stronger token lock-in to high-probability continuations.

$$q_{.99} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{p_\theta(y_t \mid x, y_{<t}) \geq 0.99\} \quad (1)$$

$$\overline{k}_{\text{eff}}^{\mathcal{R}} = \frac{1}{|\mathcal{R}|}\sum_{t\in\mathcal{R}} \exp(H(p_\theta(\cdot \mid x, y_{<t}))) \quad (2)$$

$$\overline{p}_{\text{eos}}^{\mathcal{R}} = \frac{1}{|\mathcal{R}|}\sum_{t\in\mathcal{R}} p_\theta(< \text{EOS} > \mid x, y_{<t}) \quad (3)$$

Low Candidate Diversity. To capture whether the *Runaway* window leaves few viable alternatives, we compute the effective candidate-set size. where $H(\cdot)$ denotes Shannon entropy. Lower $\overline{k}_{\text{eff}}^{\mathcal{R}}$ indicates fewer effective alternatives to escape repetitive continuation.

Weak EOS Activation. For termination, we measure EOS activation in the same *Runaway* window. Lower $\overline{p}_{\text{eos}}^{\mathcal{R}}$ indicates weaker termination tendency during repetitive continuation, prone to runaway.

Analysis. For token-level analysis, we select five models that exhibit intermediate degradation under SRLA. This excludes the near fully-failure endpoint and the zero RDR endpoints, allowing us to analyze differential degradations rather than trivial full failure or no response cases. Figure 2 summarizes the token-level results under SRLA, with more analysis and details in Appendix B.3.

Table 1: Global sample-weighted Response Degradation Rate on $\mathcal{D}_{\text{SRLA}}$; lower is better.

Model	RDR (%)
Qwen2.5-Math-7B-Instruct Yang et al. (2024)	**91.87**
DeepSeek-R1-Distill-Qwen-1.5B Guo et al. (2025)	9.38
DeepSeek-R1-Distill-Llama-8B Guo et al. (2025)	7.11
DeepSeek-R1-Distill-Qwen-7B Guo et al. (2025)	4.22
TinyLlama-1.1B-Chat Zhang et al. (2024)	3.31
OLMo2-1B-DPO OLMo et al. (2024)	1.20
Qwen2.5-3B-Instruct Qwen et al. (2025)	0.00
Llama-3-8B-Instruct Grattafiori et al. (2024)	0.00

The $q_{.99}$ shows that models with stronger degradation tend to be more locked into high-probability continuations. This identifies concentrated token continuation as one component of repetitive degeneration, although selected-token lock-in alone does not fully explain the differences across models. $\overline{k}_{\text{eff}}^{\mathcal{R}}$ examines whether effective alternatives remain available. Models with severe degradation have much smaller effective candidate sets, while milder cases retain broader continuation spaces. This suggests that candidate diversity captures how strongly generation is constrained once repetition is sustained. On the termination side, $\overline{p}_{\text{eos}}^{\mathcal{R}}$ characterizes EOS activation during repetitive continuation. Across the selected models, EOS activation remains weak during repetitive continuation. This indicates that the model distribution does not provide a sufficiently strong stopping signal when repetition is maintained.

Taken with the RDR differences, these results suggest that the self-reinforcing dynamic is not uniformly exploited across models. This shifts the question from what repetitive degeneration looks like to why some models are more likely to enter this runaway state under the same self-reinforcing pressure. To this end, the next section addresses where this model-level sensitivity comes from.

4 Post-Training Effects on Repetitive Degeneration

The analysis above exposes heterogeneous sensitivity to the self-reinforcing dynamic. Since this dynamic is examined under the same setting but varies across models, the variation points to model-side factors such as training data and training procedures. In this section, we focus on the post-training stage because it reshapes generation, including continuation and termination preferences. We therefore organize the evaluation suite into stage-wise contrasts that isolate post-training effects.

4.1 Model Suite and Post-training Contrasts

We construct an evaluation suite $\mathcal{M}$ of six models spanning two model architectures and multiple post-training configurations. Each model instance is denoted M with superscripts indicating the family and subscripts indicating the post-training method. The complete identification and training details are provided in Appendix C. To ensure consistency, all models are evaluated on the dataset used in Section 3, $\mathcal{D}_{\text{SRLA}}$. We organize the models into three contrasts $\mathcal{S}_1$-$\mathcal{S}_3$, covering non-GRPO baselines, GRPO after SFT, and GRPO-distilled data.

$\mathcal{S}_1$**: Non-GRPO baselines.** To first establish whether supervised fine-tuning alone introduces repetitive degeneration across model architectures, we select two SFT-only models spanning different base architectures. For $M_{\text{SFT}}^{\text{Qwen}}$, we start from the Qwen2.5-7B base model and perform instruction-following SFT on the Databricks-Dolly-15k dataset (Conover et al., 2023), with dataset details provided in Appendix C. $M_{\text{Instruct}}^{\text{Llama}}$ is the Meta-Llama-3-8B-Instruct (Grattafiori et al., 2024), which completes its alignment pipeline using only supervised fine-tuning. Neither model includes any reward-based optimization, and they serve as non-GRPO baselines for all subsequent comparisons.

$\mathcal{S}_2$**: GRPO after SFT.** Following the standard SFT-to-reward-optimization recipe in LLM post-training Ouyang et al. (2022); Team et al. (2025), and its recent GRPO instantiations Shao et al. (2024); Guo et al. (2025); Yang et al. (2024), we apply a GRPO stage on each SFT-only model in $\mathcal{S}_1$. Both models, $M_{\text{S-GRPO}}^{\text{Qwen}}$ and $M_{\text{S-GRPO}}^{\text{Llama}}$, are trained on the combined GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) datasets, which provide verifiable rewards suitable for the GRPO objective. Both runs use identical GRPO settings, including a learning rate of 10^{-6}, a PPO mini-batch size of 64, a per-GPU micro-batch size of 8, eight rollouts per prompt, a maximum response length of 4096, the `low_var_kl` setting, and two training epochs. No repetition-based filtering, SRLA-based sample selection, or model-specific filtering is applied. Details are provided in Appendix C. Since the models and preceding SFT stage are kept fixed within each pair, this contrast isolates the effect of the GRPO stage on generation behavior.

$\mathcal{S}_3$**: GRPO-distilled data.** $\mathcal{S}_3$ tests whether repetitive degeneration can propagate through distilled supervision without directly applying GRPO to the student model. $M_{\text{Distill}}^{\text{Llama}}$ is the official DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025), which fine-tunes Llama-3-8B on reasoning traces distilled from DeepSeek-R1. $M_{\text{Distill}}^{\text{Qwen}}$ is the official DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025), which fine-tunes Qwen2.5-7B on reasoning traces distilled from the same teacher. Since neither student model is exposed to the direct reward signal, this setup tests whether repetitive degeneration can emerge from data transfer rather than from applying GRPO to the student model itself.

4.2 Repetitive Degeneration Rates across Post-training Contrasts

Non-GRPO Baselines Remain Stable. Before any reward-based, both $M_{\text{SFT}}^{\text{Qwen}}$ and $M_{\text{Instruct}}^{\text{Llama}}$ exhibit minimal repetitive degeneration, with RDR of 2.59% and 0.00%, respectively. This establishes a low non-GRPO baseline and suggests that SFT or instruction tuning alone does not systematically induce repetition failures.

Table 2: Global sample-weighted Repetitive Degeneration Rate (RDR, %) comparing SFT/Instruct baselines against GRPO-aligned extensions across Qwen and Llama architectures; lower is better.

Model	Suite	RDR (%)	Δ (pp)
$M_{\text{SFT}}^{\text{Qwen}}$	baseline	2.59	—
$M_{\text{SFT-GRPO}}^{\text{Qwen}}$	GRPO	100.00	+97.41
$M_{\text{Instruct}}^{\text{Llama}}$	baseline	0.00	—
$M_{\text{SFT-GRPO}}^{\text{Llama}}$	GRPO	3.14	+3.14

GRPO after SFT Amplifies Repetition. Adding a GRPO stage after SFT increases the repetitive degeneration rate within both model families. For Qwen, RDR rises from 2.59% for $M_{\text{SFT}}^{\text{Qwen}}$ to $100.00\% \uparrow$ for $M_{\text{S-GRPO}}^{\text{Qwen}}$, with $\Delta = +97.41\%$. For Llama, RDR rises from 0.00% for $M_{\text{Instruct}}^{\text{Llama}}$ to $3.14\% \uparrow$ for $M_{\text{S-GRPO}}^{\text{Llama}}$, as shown in Table 2.

The absolute RDR differs across model families, but the within-family direction is consistent. Since the model family and preceding SFT stage are fixed within each pair, the increase can be associated with the added GRPO objective.

Additional robustness experiments on the KL coefficient and PPO mini-batch size are conducted separately reported in Appendix C.2, and the results support the same conclusion.

Distilled Supervision Preserves the Degeneration Tendency. The degeneration tendency associated with GRPO can also appear, in weaker form, through distilled supervision, even when GRPO is not directly applied to the student model.

As shown in Table 3, RDR rises from 0.00% to 7.11% on the Llama side and from 2.59% to 4.22% on the Qwen side.

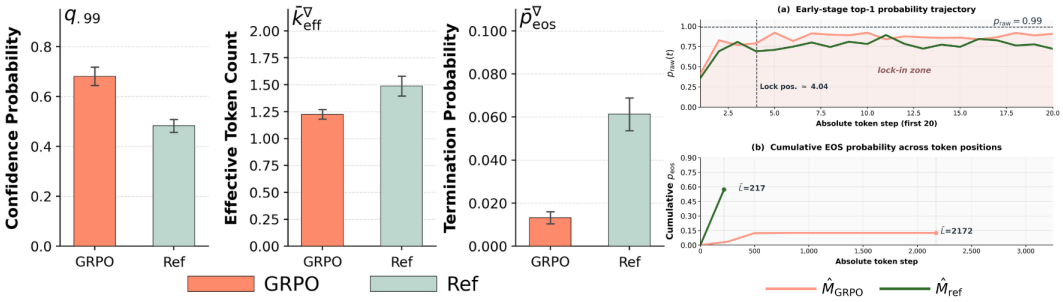


Figure 3: **Token-level traces and diagnostics for $\mathcal{D}_{\text{eval}}$.** The left panels summarize selected-token lock-in ($q_{.99}$), runaway-window candidate diversity ($\bar{k}^{\mathcal{R}}_{\text{eff}}$), and EOS activation ($\bar{p}^{\mathcal{R}}_{\text{eos}}$). The right panels show early selected-token confidence $p_{\text{raw}}(t)$ and cumulative EOS probability. The shaded region marks the lock-in zone; the dashed line marks the average lock-in position.

Since the student models are trained with supervised fine-tuning rather than direct reward optimization, these results suggest that repetitive degeneration can remain observable after distillation.

Across the three contrasts, the non-GRPO baselines remain stable, while adding GRPO after SFT increases RDR in both Qwen and Llama families. Although the absolute increase is much larger for Qwen than for Llama, the same qualitative trend appears across different model architectures. This cross-family consistency suggests that the observed degeneration is not tied to a single backbone. GRPO-distilled SFT further shows that the degeneration can be transmitted through distilled data.

Takeaway. These results trace model-level sensitivity to GRPO-related post-training, with evidence from both direct GRPO optimization across model families and GRPO-distilled supervision.

5 Mitigations

Section 4.2 shows that repetitive degeneration is associated with GRPO-related post-training. This motivates improving the GRPO training process, but RDR alone does not indicate which part of the generation dynamics should be targeted. We therefore first construct a failure-prone subset from $\mathcal{D}_{\text{SRLA}}$ that more easily triggers failures across models. This subset is used to analyze the token-level behavior of the failure to design mitigation.

Table 3: Global sample-weighted Repetitive Degeneration Rate (RDR, %) comparing SFT/Instruct baselines against models fine-tuned on GRPO-distilled data across Qwen and Llama architectures; lower is better.

Model	Suite	RDR (%)	Δ (pp)
$M^{\text{Llama}}_{\text{Instruct}}$	baseline	0.00	—
$M^{\text{Llama}}_{\text{Distill}}$	distilled data	7.11	+7.11
$M^{\text{Qwen}}_{\text{SFT}}$	baseline	2.59	—
$M^{\text{Qwen}}_{\text{Distill}}$	distilled data	4.22	+1.63

5.1 Failure-prone Mode Analysis

We instantiate $\mathcal{D}_{\text{eval}}$ as a failure-prone subset of $\mathcal{D}_{\text{SRLA}}$, selecting prompts that more easily trigger degeneration across models. We analyze two representative Qwen-family models on $\mathcal{D}_{\text{eval}}$. $\hat{M}_{\text{GRPO}}$ denotes the Qwen2.5-7B model first applying supervised fine-tuning and then applying GRPO. M_{ref} denotes Qwen2.5-7B-Instruct, used as a non-degenerate reference model. This comparison is used to analyze the token-level behavior of the failure, while the attribution to GRPO-related post-training is established by the controlled contrasts in Section 4.2. All outputs are independently regenerated on $\mathcal{D}_{\text{eval}}$ and analyzed using the token-level measures defined in Section 3.3.

Figure 3 shows the generation traces on $\mathcal{D}_{\text{eval}}$, including early selected-token confidence $p_{\text{raw}}(t)$ and cumulative EOS probability, and also reports the three token-level measures. The traces show that $\hat{M}_{\text{GRPO}}$ enters high-confidence continuation early. Its cumulative EOS probability plateaus at 0.12, and average output length exceeds $2{,}000$, compared with 217.9 for the reference model. This suggests that the failure-prone subset exposes a generation process where continuation becomes dominant while termination remains weak. The same figure further compares $\hat{M}_{\text{GRPO}}$ and M_{ref} using the three token-level measures.

Supplementary token-level statistics for GRPO failure cases and reference-model outputs on the same prompts are reported in Appendix D.1.

On $\mathcal{D}_{\text{eval}}$, $\hat{M}_{\text{GRPO}}$ exhibits more high-confidence selected-token steps than the non-degenerate reference. The degenerate models also show fewer effective candidate tokens in the runaway window and near-zero EOS probabilities. These results indicate selected-token lock-in, reduced candidate diversity, and weak EOS activation in the failure cases.

To examine EOS behavior beyond the runaway-window average, Table 4 reports EOS probability over both the full sequence and the runaway window $\mathcal{R}$.

Across all models, EOS probability increases inside $\mathcal{R}$ compared with the full-sequence average. The increase is much smaller for $\hat{M}_{\text{GRPO}}$ than for M_{ref}. This supports the view that repetitive degeneration is tied to persistent under-activation of EOS.

Table 4: Averaged dynamic EOS activation patterns, $\mathbb{E}[p_{\text{eos}}]_{\mathcal{R}}$ denotes the mean stop signal constrained within the runaway window $\mathcal{R}$.

Metric	GRPO	DPO	Ref
(1) $\mathbb{E}[p_{\text{eos}}]$ (Full Seq.)	0.0005	0.0007	**0.0055**
(2) $\mathbb{E}[p_{\text{eos}}]_{\mathcal{R}}$	0.0019	0.0028	**0.0212**
Shift Δ (2) - (1)	+0.0014	+0.0021	**+0.0157**

5.2 From Analysis to Mitigation Targets

The failure-prone analysis points to two mitigation targets: *EOS suppression* and *distributional concentration*. EOS suppression reflects the stop token remains under-activated during repetition. Distributional concentration reflects high-confidence continuation with fewer effective alternatives.

EOS Suppression. Preference optimization is known to introduce length biases that favor longer outputs (Singhal et al., 2023; Bu et al., 2025; Park et al., 2024). Recent work further shows that such optimization can suppress EOS emission probability during generation (Gupta et al., 2024). This is consistent with our failure cases, where p_{eos} remains persistently low. This motivates a termination-oriented reward that encourages the model to stop after completing the task.

Distributional Concentration. Optimization on a narrow task distribution can focus reward pressure on limited behavioral modes and sharpen the next-token distribution (Gao et al., 2023; Chen et al., 2026; Fan et al., 2026). This can reduce candidate diversity as the model exploits narrow reward signals (Qwen et al., 2025). This motivates a diversification-oriented training design that broadens the optimization signal across task types.

5.3 Mitigation Design

Conciseness Reward. The first intervention targets EOS suppression by making task completion explicitly visible to the GRPO objective. Under a correctness-only reward, a response can receive the same reward even if it continues after producing the correct answer. This leaves unnecessary continuation weakly penalized. For $M^{\text{Qwen}}_{\text{Concise}}$, we therefore augment the reward with a conciseness term that is activated only when the response is correct. The total reward is defined as

$$r_{\text{total}} = r_{\text{acc}} + \lambda\, r_{\text{brev}}\, \mathbf{1}[r_{\text{acc}} > 0],$$

where r_{acc} is a binary correctness reward, r_{brev} assigns higher values to shorter responses, and λ controls the strength of the brevity term. The indicator preserves correctness as the primary objective, while discouraging excessive continued generation.

Multi-task Diversification. The second intervention targets distributional concentration by broadening the reward-bearing contexts used during GRPO training. When optimization is concentrated on a narrow task, the model may over-specialize to a small set of solution templates and continuation patterns. For $M^{\text{Qwen}}_{\text{Multi}}$, we therefore rebalance the training mixture across complementary task types. Specifically, 35% of the training data comes from English mathematics, 20% from Chinese mathematics, 15% from code generation tasks, and 30% from instruction-following tasks. This design keeps verifiable reasoning as the main training signal, while introducing broader behavioral contexts that reduce degradation to a single continuation mode. Full dataset sources are provided in Appendix F.

5.4 Mitigation Results

We compare the unmitigated GRPO model with the two mitigation variants, $M^{\text{Qwen}}_{\text{Concise}}$ and $M^{\text{Qwen}}_{\text{Multi}}$. Each prompt is run 3 times independently and evaluated using RDR together with the token-level metric defined in Section 3.3. Figure 4 and Table 5 report the outcomes and token-level metrics.

Table 5: Token-level mitigation indicators on $\mathcal{D}_{\text{eval}}$. **Bold** and underline denote the best and second-best values within each metric.

Metric	GRPO	Concise	Δ_{Concise}	Multi-task	Δ_{Multi}
Late-stage candidate set (k_{late})	1.225	1.180	-4%	**1.500**	**$+22\%$** ↑
Collapse rate ($p_{\text{raw}} \geq 0.99$)	0.681	0.720	$+6\%$	**0.442**	**-35%** ↓
Repetition rate (late 25%)	0.418	0.380	-9%	**0.109**	**-74%** ↓
Mean p_{eos} (late 25%)	0.0132	**0.0340**	**$+158\%$** ↑	0.0144	$+9\%$

Conciseness reward. As shown in Table 5, the conciseness reward increases late-window p_{eos} by 158%, indicating stronger EOS activation. The distributional concentration remains largely unchanged relative to the unmitigated GRPO model, but RDR is reduced by 81.10%. This suggests that directly counteracting EOS suppression can partially restore timely termination.

Multi-task diversification. The multi-task mitigation significantly affects candidate diversity when decoding. As shown in Table 5, the runaway window effective candidate set increases to 1.500, corresponding to a 22.5% increase. This broader token-level diversity is accompanied by a reduction of repetition from 41.8% to 10.9%, and a slight increase in EOS probability from 1.32% to 1.44%. As a result, RDR is reduced by 91.46%, dropping from the 16.51% GRPO baseline to 1.41%.

Standard capability check. To test whether these mitigations materially degrade ordinary task performance, we evaluate the DPO baseline together with $M^{\text{Qwen}}_{\text{Concise}}$ and $M^{\text{Qwen}}_{\text{Multi}}$ on MMLU, GSM8K, IFEval, and HumanEval. Appendix G shows that both mitigation models remain broadly comparable to the DPO baseline.

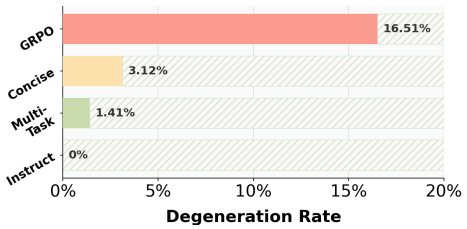


Figure 4: RDR on $\mathcal{D}_{\text{eval}}$ across four Qwen variants, ordered by decreasing crash rate. Red denotes degeneration cases; hatching denotes up to 20% RDR.

Takeaway. The conciseness reward mainly strengthens EOS activation, while multi-task diversification mainly broadens candidate diversity and reduces concentration. The two mitigations reduce repetitive degeneration through different paths.

6 Conclusion

We study repetitive degeneration in LLMs as a post-training-related failure rather than a purely decoding-time artifact. To make this sporadic failure reproducible, we introduce SRLA to elicit self-reinforcing repetition under benign task prompts. Building on SRLA, we use controlled post-training contrasts to trace model-level sensitivity to GRPO-related training. Our analysis shows that this sensitivity is reflected in a continuation-termination imbalance during generation. We further show that this failure can be mitigated by modifying the GRPO training process. A conciseness reward improves termination behavior, while multi-task diversification reduces over-concentrated continuation. These findings connect repetitive degeneration to concrete post-training choices and provide practical guidance for more robust reward-based training.

AI use statement

Generative AI tools were used to assist with manuscript editing, consistency checks, discussion of result interpretations, and preparation of plotting code. The authors reviewed the resulting suggestions and verified the text, code, and figures incorporated into the manuscript. All reported numerical results were derived from experimental outputs and subsequent analyses. The authors take full responsibility for the final manuscript and accompanying artifacts.

Ethics statement

This work studies repetitive generation to improve the reliability and computational efficiency of language models. SRLA can elicit long repetitive outputs and could be misused to increase inference costs. Evaluations in this study use a fixed generation limit of 4,096 tokens per response. The proposed mitigations aim to reduce repetitive degeneration.

Reproducibility Statement

The construction of SRLA and the evaluation protocol are described in Section 3 and Appendices A–B. Model identifiers, training details, and GRPO hyperparameters are provided in Appendix C. Section 5 describes the mitigation experiments, with dataset composition and capability evaluation reported in Appendices F–G. To support reproducibility, we will publicly release the code with the camera-ready version, including scripts for SRLA construction and evaluation, together with the corresponding configuration files and the dataset splits used in our experiments.

References

Yang Bai, Ge Pei, Jindong Gu, Yong Yang, and Xingjun Ma. Special characters attack: Toward scalable training data extraction from large language models. *arXiv preprint arXiv:2405.05990*, 2024.

Botond Barta, Endre Hamerlik, Milán Nyist, Masato Ito, and Judit Acs. Enhancing AMR parsing with group relative policy optimization. In Hao Fei, Kewei Tu, Yuhui Zhang, Xiang Hu, Wenjuan Han, Zixia Jia, Zilong Zheng, Yixin Cao, Meishan Zhang, Wei Lu, N. Siddharth, Lilja Øvrelid, Nianwen Xue, and Yue Zhang (eds.), *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pp. 99–105, Vienna, Austria, August 2025. Association for Computational Linguistics. ISBN 979-8-89176-286-2. doi: 10.18653/v1/2025.xllm-1.11. URL https://aclanthology.org/2025.xllm-1.11/.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

Yuyan Bu, Liangyu Huo, Yi Jing, and Qing Yang. Beyond excess and deficiency: Adaptive length bias mitigation in reward models for RLHF. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3091–3098, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.169. URL https://aclanthology.org/2025.findings-naacl.169/.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026.

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

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Jianshuo Dong, Ziyuan Zhang, Qingjie Zhang, Tianwei Zhang, Hao Wang, Hewu Li, Qi Li, Chao Zhang, Ke Xu, and Han Qiu. An engorgio prompt makes large language model babble on. In *International Conference on Learning Representations*, volume 2025, pp. 67280–67307, 2025.

Mingyuan Fan, Weiguang Han, Daixin Wang, Cen Chen, Zhiqiang Zhang, and Jun Zhou. When sharpening becomes collapse: Sampling bias and semantic coupling in rl with verifiable rewards. *arXiv preprint arXiv:2601.15609*, 2026.

Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. A theoretical analysis of the repetition problem in text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 12848–12856, 2021.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International conference on machine learning*, pp. 10835–10866. PMLR, 2023.

Jonas Geiping, Alex Stein, Manli Shu, Khalid Saifullah, Yuxin Wen, and Tom Goldstein. Coercing llms to do and reveal (almost) anything. *arXiv preprint arXiv:2402.14020*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations*, volume 2025, pp. 97114–97144, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Taneesh Gupta, Rahul Madhavan, Xuchao Zhang, Chetan Bansal, and Saravan Rajmohan. Refa: Reference free alignment for multi-preference optimization. *arXiv preprint arXiv:2412.16378*, 2024.

Hojae Han, Jaejin Kim, Jaeseok Yoo, Youngwon Lee, and Seung-won Hwang. ArchCode: Incorporating software requirements in code generation with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13520–13552, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.730. URL https://aclanthology.org/2024.acl-long.730/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Tatsuya Hiraoka and Kentaro Inui. Repetition neurons: How do language models produce repetitions? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 483–495, 2025.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Qisheng Hu, Geonsik Moon, and Hwee Tou Ng. From moments to milestones: Incremental timeline summarization leveraging large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7232–7246, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.390. URL https://aclanthology.org/2024.acl-long.390/.

Maor Ivgi, Ori Yoran, Jonathan Berant, and Mor Geva. From loops to oops: Fallback behaviors of language models under uncertainty. *arXiv preprint arXiv:2407.06071*, 2024.

Weisen Jiang, Han Shi, Longhui Yu, Zhengying Liu, Yu Zhang, Zhenguo Li, and James Kwok. Forward-backward reasoning in large language models for mathematical verification. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6647–6661, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.397. URL https://aclanthology.org/2024.findings-acl.397/.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. In *International Conference on Learning Representations*, volume 2024, pp. 20620–20653, 2024.

Huayang Li, Tian Lan, Zihao Fu, Deng Cai, Lemao Liu, Nigel Collier, Taro Watanabe, and Yixuan Su. Repetition in repetition out: Towards understanding neural text degeneration from the data perspective. *Advances in Neural Information Processing Systems*, 36:72888–72903, 2023.

Xingyu Li, Xiaolei Liu, Cheng Liu, Yixiao Xu, Kangyi Ding, Bangzhou Xin, and Jia-Li Yin. Loopllm: Transferable energy-latency attacks in llms via repetitive generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31770–31777, 2026.

Zekun Li, Zhiyu Chen, Mike Ross, Patrick Huber, Seungwhan Moon, Zhaojiang Lin, Xin Luna Dong, Adithya Sagar, Xifeng Yan, and Paul A Crook. Large language models as zero-shot dialogue state tracker through function calling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8688–8704, 2024.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Matéo Mahaut and Francesca Franzon. Repetitions are not all alike: distinct mechanisms sustain repetition in language models. *arXiv preprint arXiv:2504.01100*, 2025.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent llm outputs. In *International Conference on Learning Representations*, volume 2025, pp. 70333–70366, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Laura O'Mahony, Leo Grinsztajn, Hailey Schoelkopf, and Stella Biderman. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, volume 2, pp. 2, 2024.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4998–5017, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.297. URL https://aclanthology.org/2024.findings-acl.297/.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin

Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115v2.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Sergio Servantez, Joe Barrow, Kristian Hammond, and Rajiv Jain. Chain of logic: Rule-based reasoning with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2721–2733, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.159. URL https://aclanthology.org/2024.findings-acl.159/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Wentao Shi, Mengqi Yuan, Junkang Wu, Qifan Wang, and Fuli Feng. Direct multi-turn preference optimization for language agents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2312–2324, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.138. URL https://aclanthology.org/2024.emnlp-main.138/.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*, 2023.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Yuanhe Tian, Fei Xia, and Yan Song. Dialogue summarization with mixture of experts based on large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7143–7155, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.385. URL https://aclanthology.org/2024.acl-long.385/.

Tianduo Wang, Shichen Li, and Wei Lu. Self-training with direct preference optimization improves chain-of-thought reasoning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11917–11928, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.643. URL https://aclanthology.org/2024.acl-long.643/.

Weichuan Wang, Zhaoyi Li, Defu Lian, Chen Ma, Linqi Song, and Ying Wei. Mitigating the language mismatch and repetition issues in llm-based machine translation via model editing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15681–15700, 2024b.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.

Jason Wu, Eldon Schoop, Alan Leung, Titus Barik, Jeffrey Bigham, and Jeffrey Nichols. UICoder: Finetuning large language models to generate user interface code through automated feedback. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7511–7525, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.417. URL https://aclanthology.org/2024.naacl-long.417/.

Guanming Xiong, Junwei Bao, and Wen Zhao. Interactive-kbqa: Multi-turn interactions for knowledge base question answering with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10561–10582. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.569. URL http://dx.doi.org/10.18653/v1/2024.acl-long.569.

Jianhao Elliott Yan, Jin Xu, Chiyu Song, Chenming Wu, Yafu Li, and Yue Zhang. Understanding in-context learning from repetitions. In *International Conference on Learning Representations*, volume 2024, pp. 49968–49989, 2024.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Junchi Yao, Shu Yang, Jianhua Xu, Lijie Hu, Mengdi Li, and Di Wang. Understanding the repeat curse in large language models from a feature perspective. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 7787–7815, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.406. URL https://aclanthology.org/2025.findings-acl.406/.

Itay Yona, Ilia Shumailov, Jamie Hayes, and Yossi Gandelsman. Interpreting the repeated token phenomenon in large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72535–72555. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/yona25a.html.

Longfei Yun, Chenyang An, Zilong Wang, Letian Peng, and Jingbo Shang. The price of format: Diversity collapse in llms. In *EMNLP (Findings)*, pp. 15454–15468, 2025.

Jixiao Zhang and Chunsheng Zuo. GRPO-LEAD: A difficulty-aware reinforcement learning approach for concise mathematical reasoning in language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5642–5654, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.287. URL https://aclanthology.org/2025.emnlp-main.287/.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

Wenhong Zhu, Hongkun Hao, Zhiwei He, Yiming Ai, and Rui Wang. Improving open-ended text generation via adaptive decoding. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62386–62404. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhu24d.html.

A SRLA EVALUATION SET DETAILS

A.1 DATASET SPECIFICATIONS AND SAMPLE STATISTICS

To ensure a rigorous empirical evaluation of microscopic token-level behaviors, we strictly standardize the volumes of our benchmark suites. Specifically, $\mathcal{D}_{\text{base}}$ and $\mathcal{D}_{\text{SRLA}}$ are structurally identical in scale, with **each dataset independently containing exactly** 996 **prompts**.

A.2 Perplexity Validation of SRLA Prompts

Because SRLA uses a controlled-decoding procedure to extract repetitive seeds, one potential concern is that the constructed evaluation prompts may be specific to this auxiliary decoder rather than reflecting inputs that are naturally compatible with the evaluated models. To examine this issue, we measure the perplexity of each SRLA prompt under the native next-token distributions of both the seed model and the downstream evaluated models.

Recall that each SRLA prompt is constructed as $x_{\mathrm{SRLA}} = x \oplus r_x$, where x denotes the original base prompt, r_x denotes the extracted repetitive seed, and $\oplus$ denotes string concatenation under the original prompt template. For a model m with native next-token distribution p_m, we compute the perplexity of x_{SRLA} as

$$\mathrm{PPL}_m(x_{\mathrm{SRLA}}) = \exp\left(-\frac{1}{|x_{\mathrm{SRLA}}|}\sum_{i=1}^{|x_{\mathrm{SRLA}}|}\log p_m\left((x_{\mathrm{SRLA}})_i \mid (x_{\mathrm{SRLA}})_{<i}\right)\right).$$

This validation uses the models' original distributions without applying the controlled decoder used during seed extraction. Therefore, low perplexity indicates that the resulting SRLA prompts remain likely under standard model behavior. Table 6 reports the aggregated perplexity statistics by dataset group. Across the evaluated models, SRLA prompts exhibit low perplexity, suggesting that $\mathcal{D}_{\mathrm{SRLA}}$ remains compatible with each model's native distribution. This provides additional evidence that the repetition induced by SRLA is not merely an artifact of controlled decoding, these prompts naturally drive the models to generate repetitive outputs with high probability.

Table 6: Aggregated language-specific perplexity (PPL) across models, reported using the median, mean, and IQR. The largest value within each language group and column is shown in bold.

Group	Median PPL	Mean PPL	IQR
CHI_MCQ	**10.77**	**21.43**	**7.10**
CHI_Seed	3.09	2.77	0.42
CHI_SRLA	2.64	2.23	0.38
ENG_MCQ	**9.97**	**10.30**	**2.08**
ENG_Seed	2.47	2.38	0.13
ENG_SRLA	4.85	4.73	0.45

B Definition of the Repetitive Degeneration Indicator

We evaluate each model using the repetitive degeneration rate (RDR), defined as

$$\mathrm{RDR} = \frac{\mathrm{Rep.}}{\mathrm{Valid}},$$

where $\mathrm{Rep.}$ denotes the number of cases in which the generation reaches the maximum token budget ($T_{\max}$) and terminates in repetition, and Valid denotes the total number of valid test cases.

B.1 Algorithmic Criteria for Repetitive Degradation Detection

To rigorously quantify the Repetitive Degeneration Rate (RDR) without introducing human annotation bias, we establish a deterministic, token-level string-matching protocol to identify degenerated trajectories. The detection heuristic specifically targets the terminal behavior of the generated sequences to isolate stable, self-reinforcing loops from standard contextual alignment. Formally, for any generated sequence exceeding a minimum viable length constraint, the verification is applied exclusively to the final quarter of the sequence. Within this localized window, a token-level sliding-window search is executed to identify repeating sub-sequences under the following strict threshold configuration:

- **n-gram Size (ngram_size $= 6$):** The sequence is decomposed into continuous segments of 6 tokens.
- **Minimum Frequency (min_repeats $= 4$):** A specific 6-gram pattern must re-occur at least 4 distinct times within the trailing window to be labeled as entering a structural, self-reinforcing loop.

A trajectory is formally marked as *degraded* if and only if it satisfies this joint criteria, which ensure that the RDR metric strictly captures structural behavioral *degeneration*, rather than artifactual

statistical collisions induced purely by extended sequence lengths. Crucially, this configuration has been verified via manual spot-checks, demonstrating the accuracy with human-perceived structural degeneration. Sensitivity analysis shows that larger n-gram sizes or higher frequencies yield no accuracy gain, while smaller thresholds introduce heavy noise by misclassifying normal text as degeneration.

B.2 Granular RDR Evaluation Across Datasets

Table 7: Global sample-weighted Repetitive Degeneration Rate (RDR, %) across $\mathcal{D}_{\text{SRLA}}$ and $\mathcal{D}_{\text{Base_mcq}}$, where Δ measures the absolute margin ($\mathcal{D}_{\text{SRLA}}$ − Base_mcq); lower is better.

Model	RDR (%)		Δ
	$\mathcal{D}_{\text{SRLA}}$	Base_mcq	
Qwen2.5-Math-7B-Instruct Yang et al. (2024)	**91.87**	**25.74**	**66.13**
DeepSeek-R1-Distill-Qwen-1.5B Guo et al. (2025)	9.38	0.99	8.39
DeepSeek-R1-Distill-Llama-8B Guo et al. (2025)	7.11	0.00	7.11
DeepSeek-R1-Distill-Qwen-7B Guo et al. (2025)	4.22	0.00	4.22
TinyLlama-1.1B-Chat Zhang et al. (2024)	3.31	0.99	2.32
OLMo2-1B-DPO OLMo et al. (2024)	1.20	0.00	1.20
Qwen2.5-3B-Instruct Qwen et al. (2025)	0.00	0.00	0.00
Llama-3-8B-Instruct Grattafiori et al. (2024)	0.00	0.00	0.00

To guaranty robust deployment and evaluate the behavioral boundaries of audited large language models, we conducted a fine-grained statistical analysis on the text generation length constraints. Specifically, we investigate generation behaviors under two distinct question groups: the primary data set $\mathcal{D}_{\text{SRLA}}$ and a baseline multiple-choice question group, denoted Base _mcq, in table 7.

B.3 Supplementary Quantitative Analysis on Token-level EOS Probability Shifts

To further examine the behavior between text continuation and sequence termination, Table 8 reports the fine-grained EOS behavior from a complementary perspective. Across the models that exhibit intermediate degradation, Table 8 quantifies a persistent probability gap between token continuation and sequence termination. This uniform negative shift (Δ) suggests that once repetition is sustained, the model's native logit surface tends to remain locked in a state that favors continuation over termination.

C Training Details

C.1 Model Identification

Table 9 summarizes the eight models in $\mathcal{M}$, including the base model, post-training stages, training data, and whether the checkpoint is trained by us or adopted from a public release.

Table 8: Token-level alignment shift $\Delta = \overline{p}^{\mathcal{R}}_{\text{eos}} - \overline{p}_{\text{raw}}$ across intermediate-degradation models, measuring the margin between token lock-in and termination intent within the runaway window $\mathcal{R}$.

Metric	Llama-3-8B Distill	DS-R1-Distill 7B	DS-R1-Distill 1.5B	OLMo2-1B DPO	TinyLlama 1.1B
(1) $\overline{p}_{\text{raw}}$	0.8368 ± 0.0127	0.7130 ± 0.0104	0.6119 ± 0.0126	0.6386 ± 0.0111	0.5809 ± 0.0285
(2)$\overline{p}^{\mathcal{R}}_{\text{eos}}$	0.0544 ± 0.0116	0.0092 ± 0.0011	0.0065 ± 0.0009	0.0058 ± 0.0005	0.0154 ± 0.0025
Shift Δ (2) − (1)	-0.7825 ± 0.0172	-0.7038 ± 0.0105	-0.6054 ± 0.0127	-0.6328 ± 0.0111	-0.5655 ± 0.0286

Table 9: Identification of the models in $\mathcal{M}$ introduced in Section 4.1.

Model	Base	Post-training	Training data	Source
$M_{\text{SFT}}^{\text{Qwen}}$	Qwen2.5-7B	SFT	Dolly-15k	Ours
$M_{\text{Instruct}}^{\text{Llama}}$	Meta-Llama-3-8B-Instruct	SFT	—	Public
$M_{\text{S-GRPO}}^{\text{Qwen}}$	Qwen2.5-7B	SFT → GRPO	Dolly-15k; GSM8K + MATH	Ours
$M_{\text{Distill}}^{\text{Llama}}$	Meta-Llama-3-8B	GRPO Distill	DeepSeek-R1 Trajectory	Public (Guo et al., 2025)
$M_{\text{Distill}}^{\text{Qwen}}$	Qwen2.5-7B	GRPO Distill	DeepSeek-R1 Trajectory	Public (Guo et al., 2025)

Table 10: URLs for public model checkpoints.

Model	URL
$M_{\text{Instruct}}^{\text{Llama}}$	`https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct`
$M_{\text{Distill}}^{\text{Llama}}$	`https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B`
$M_{\text{Distill}}^{\text{Qwen}}$	`https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B`

Dataset links.

- Dolly-15k:`https://huggingface.co/datasets/databricks/databricks-dolly-15k`
- GSM8K:`https://huggingface.co/datasets/openai/gsm8k`
- MATH-lighteval:`https://huggingface.co/datasets/DigitalLearningGmbH/MATH-lighteval`

Model Links. URLs for public models used in Section 4.1 are listed in Table 10.

C.2 Shared GRPO Settings and Supplementary Controls

Direct comparisons of SFT→ GRPO in Section 4.2 use shared core GRPO settings across the Qwen and Llama model families, with learning rate 10^{-6}, PPO mini-batch size 64, micro-batch size per-GPU 8, eight rollouts per prompt, maximum response length 4096, `low` setting _var _kl and two training epochs. In these runs, repetition-based filtering, SRLA-based sample selection, or target-model-specific filtering is not applied.

To probe whether the observed increase is tied to one specific hyperparameter choice, we additionally evaluate a Qwen DPO→GRPO pipeline while varying only the KL coefficient or PPO mini-batch size. Table 11 shows that all four perturbations remain above the DPO baseline, supporting the interpretation that the observed increase is associated with the tested GRPO-related stage rather than only one of these hyperparameter settings.

Table 11: Supplementary hyperparameter perturbation controls on a Qwen DPO→GRPO pipeline. Training data, reward design, filtering policy, and two-epoch training length are fixed; only the KL coefficient or PPO mini-batch size is varied.

Model	Global RDR (%)
$M_{\text{DPO}}^{\text{Qwen}}$	14.76
$M_{\text{mixed-GRPO}}^{\text{Qwen}}$	100.00
$M_{\text{mixed-GRPO}(\beta_{\text{KL}}\times 0.1)}^{\text{Qwen}}$	49.80
$M_{\text{mixed-GRPO}(\beta_{\text{KL}}\times 10)}^{\text{Qwen}}$	57.13
$M_{\text{mixed-GRPO}(\text{mb}\times 0.5)}^{\text{Qwen}}$	48.39
$M_{\text{mixed-GRPO}(\text{mb}\times 2)}^{\text{Qwen}}$	23.59

D Auxiliary Token-Level Diagnostics

We additionally examine three auxiliary token-level diagnostics that complement the runaway-window measures defined in Section 3.3 along two orthogonal axes. The first two, $r_{\text{eos}>.01}$ and $I_{\text{eos}>.1}$, broaden the analysis of EOS activation by extending $\overline{p}_{\text{eos}}^{\mathcal{R}}$ from a within-window mean to (i) a full-sequence frequency view of weak activation, and (ii) a binary indicator of whether EOS is ever appreciably activated. The third, q_{nar}, complements

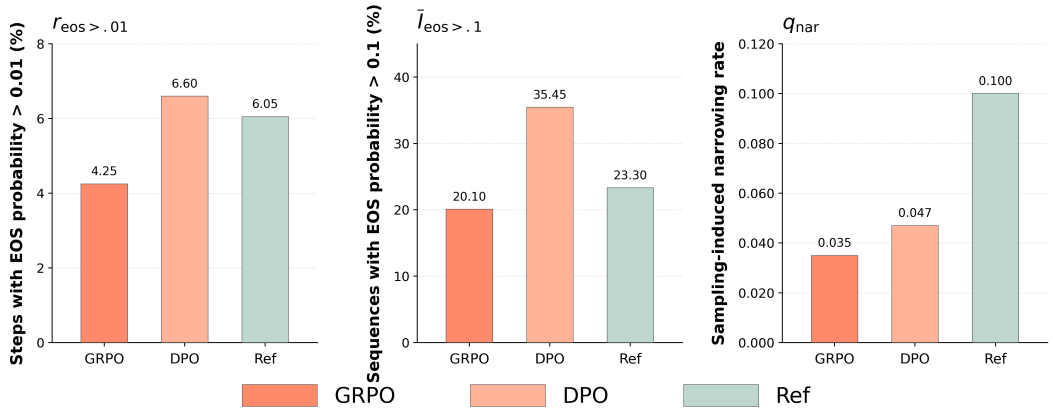


Figure 5: Auxiliary token-level diagnostics. $r_{\text{eos}>.01}$ measures the fraction of decoding steps for which $p_{\text{eos}}(t) > 0.01$. $I_{\text{eos}>.1}$ indicates whether the model ever assigns $p_{\text{eos}}(t) > 0.1$. q_{nar} measures the fraction of decoding steps for which $p_{\text{raw}}(t) < 0.90$, $p_{\text{nucleus}}(t) \geq 0.95$, and nucleus filtering retains exactly one token.

the candidate-diversity measure $\overline{k}^{\mathcal{R}}_{\text{eff}}$ by shifting from a within-window mean to the full-sequence fraction of steps with a critically narrow candidate set. Together, these statistics report how often each failure-related signal occurs across the entire sequence, complementing how strong it is within the runaway window. Note that the selected-token lock-in measure $q_{.99}$ is already defined as a full-sequence fraction in Section 3.3.

$$r_{\text{eos}>.01} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\left[p_{\text{eos}}(t) > 0.01\right],$$

where T is the generation length and $p_{\text{eos}}(t)$ denotes the raw probability assigned to the EOS token at decoding step t. $r_{\text{eos}>.01}$ captures the global prevalence of weakly-activated EOS steps independent of where they occur, comparing with $\overline{p}^{\mathcal{R}}_{\text{eos}}$, which averages EOS probability inside the runaway window

Second, we record whether the model ever assigns a substantially larger probability to EOS:

$$I_{\text{eos}>.1} = \mathbf{1}\left[\max_{1\leq t\leq T} p_{\text{eos}}(t) > 0.1\right].$$

This binary statistic indicates whether the model ever enters a state in which termination receives appreciable probability mass. $I_{\text{eos}>.1}$ records whether such activation ever occurs at all over the full sequence, comapring with $\overline{p}^{\mathcal{R}}_{\text{eos}}$ measures averaged activation strength inside the runaway window

Third, we report an auxiliary narrowing statistic q_{nar}, defined as the fraction of decoding steps satisfying the following joint condition:

$$q_{\text{nar}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\left[p_{\text{raw}}(t) < 0.90 \ \wedge \ p_{\text{nucleus}}(t) \geq 0.95 \ \wedge \ k(t) = 1\right],$$

where $p_{\text{raw}}(t)$ and $p_{\text{nucleus}}(t)$ denote the selected token's probabilities before and after nucleus filtering and renormalization, respectively, and $k(t)$ is the number of tokens retained by nucleus filtering.

Figure 5 reports these auxiliary diagnostics.

D.1 Additional Token-level Statistics

Table 12 reports supplementary token-level statistics for cases in which $\hat{M}_{\text{GRPO}}$ exhibits repetitive degeneration and compares its outputs with those of M_{ref} on the same prompts, complementing the

mechanism analysis on $\mathcal{D}_{\text{eval}}$ in Section 5.1.Beyond the primary metrics discussed in the main text ($q_{.99}$, $\overline{k}^{\mathcal{R}}_{\text{eff}}$, and $\overline{p}_{\text{raw}}$), these sequence-level diagnostics further characterize the temporal onset, path concentration, and state space of repetitive degeneration.

Sequence length and onset timing. The average output length $\overline{L}$ of $\hat{M}_{\text{GRPO}}$ fully saturates the maximum sequence limit 100%, whereas M_{ref} utilizes merely 6.1% of this capacity (247.9 tokens). While the absolute repetition onset positions remain comparable (200.0 vs. 95.3), the relative onset ratio is drastically smaller for $\hat{M}_{\text{GRPO}}$ (0.049 vs. 0.327). This juxtaposition indicates that degenerate patterns emerge early within a truncated sequence that actively fails to self-terminate in $\hat{M}_{\text{GRPO}}$.

Table 12: Token-level aggregate statistics for GRPO failure cases and reference-model outputs on the same prompts.

Metric	GRPO	Ref
$\overline{L}$	4096.0	247.9
Rep. onset pos.	200.0	95.3
Rep. onset ratio	0.049	0.327
Lock pos.	4.04	6.54
$\overline{p}_{\text{raw}}$	0.993	0.757
$q_{.99}$	0.968	0.368
Mean PPL	1.043	1.596

Selected-token concentration and lock-in. The lock-in position (4.04 vs. 6.54) indicates that $\hat{M}_{\text{GRPO}}$ enters a near-deterministic continuation regime within the opening decoding steps. Driven by this, $\overline{p}_{\text{raw}}$ reaches 0.993 and $q_{.99}$ plateaus at 0.968, confirming that almost every decoding step assigns a top-1 confidence exceeding 0.99. Conversely, M_{ref} exhibits substantially broader distribution dispersion ($\overline{p}_{\text{raw}} = 0.757$, $q_{.99} = 0.368$). This low-entropy trajectory under $\hat{M}_{\text{GRPO}}$ is further validated by its minimal mean perplexity (1.043 vs. 1.596), proving that the repetitive path remains highly favored.

Collectively, these aggregate statistics confirm the core findings in Section 5.1 the failure-prone subset exposes a generation regime where path continuation becomes single-mindedly dominant while self-termination remains systematically under-activated.

E Scale-Invariant Justification for the Terminal Window Fraction

To evaluate token-level microscopic dynamics without introducing length-dependent artifacts, we employ a relative trailing fraction (tail_fraction $= 0.25$) rather than a fixed token-length window. A fixed-window paradigm underestimates degradation in hyper-extended sequences. Since generated sequences vary drastically in length, our relative 25% terminal window dynamically scales with the sequence, ensuring a consistent and high-fidelity auditing of late-stage token behaviors across sequences.

F Multi-task Dataset Composition

Table 13 lists the datasets used for the multi-task diversification variant $M^{\text{Qwen}}_{\text{Multi}}$.

G Standard Capability Evaluation of Mitigation Models

To check whether the mitigation models trade off ordinary task performance for lower repetitive degeneration, we evaluate the DPO baseline together with $M^{\text{Qwen}}_{\text{Concise}}$ and $M^{\text{Qwen}}_{\text{Multi}}$ using `lm-evaluation-harness` (Biderman et al., 2024). We report MMLU, GSM8K, IFEval, and HumanEval in Table 14.

Table 13: Training data composition for $M_{\text{Multi}}^{\text{Qwen}}$.

Task	Proportion	Dataset	Source
English math	35%	MATH-lighteval	https://huggingface.co/datasets/DigitalLearningGmbH/MATH-lighteval
Chinese math	20%	GSM8K_zh	https://huggingface.co/datasets/meta-math/GSM8K_zh
Code generation	15%	MBPP	https://huggingface.co/datasets/google-research-datasets/mbpp
Instruction following	30%	RLVR-IFeval	https://huggingface.co/datasets/allenai/RLVR-IFeval

Table 14: Standard capability evaluation for the mitigation models relative to the DPO baseline. Scores are reported in percentage points.

Model	MMLU	Δ	GSM8K	Δ	IFEval	Δ	HumanEval	Δ
$M_{\text{DPO}}^{\text{Qwen}}$	58.08	—	89.99	—	21.26	—	48.78	—
$M_{\text{Multi}}^{\text{Qwen}}$	58.12	+0.04	90.30	+0.31	21.26	0.00	50.00	+1.22
$M_{\text{Concise}}^{\text{Qwen}}$	58.10	+0.02	89.69	-0.30	20.89	-0.37	45.12	-3.66

H Additional Positioning Against Prior Work

Prior work studies repetition through decoding-time dynamics, context-induced reinforcement, and internal mechanisms. Complementary alignment studies examine diversity or mode collapse after fine-tuning and preference optimization. In contrast, we provide a black-box, prompt-only protocol for controlled cross-stage comparison, and connect the resulting token-level signatures to targeted post-training interventions.

White-box mechanism analyses. Prior SAE- or circuit-based studies localize internal features associated with repetition (Yao et al., 2025; Yona et al., 2025), whereas SRLA is a black-box prompt-only protocol designed for controlled cross-model comparison.

Decoding-time controls. Decoding-focused methods improve generation quality by changing inference-time sampling or filtering (Zhu et al., 2024; Nguyen et al., 2025), whereas we keep decoding fixed and study training-stage effects.

In-context and adversarial induction. Work on in-context repetition or adversarially induced long outputs studies how repetition can be reinforced or intentionally triggered (Yan et al., 2024; Dong et al., 2025). In contrast, SRLA uses fixed non-adversarial prompts to expose benign but unstable degeneration across models.

Alignment and diversity effects. Broader alignment analyses examine diversity loss, mode collapse, or fine-tuning effects (O'Mahony et al., 2024; Kirk et al., 2024; Yun et al., 2025). Our focus is narrower: repetitive degeneration under a shared elicitation setting, together with GRPO-side mitigation experiments.

LIMITATIONS

This work has several limitations. First, SRLA is designed as a controlled elicitation method rather than a complete model evaluation benchmark. It focuses on prompts that expose self-reinforcing repetitive continuations, and therefore does not cover all possible forms of repetition that may appear in open-ended dialogue, long-form writing, code generation, or multi-turn interaction.

Second, our model suite covers several representative open-weight model families and post-training configurations, but it does not exhaust all architectures, scales, data mixtures, or proprietary training pipelines. The controlled contrasts are intended to isolate GRPO-related effects within the available settings. Other implementations of reward-based optimization may use different reward models, training schedules, filtering rules, or stopping criteria, which could affect the degree of repetitive degeneration.

Third, the token-level measures used in this work summarize observable distributional behavior during generation. Metrics such as selected-token lock-in, effective candidate-set size, and EOS activation help characterize the failure state, but they do not provide a complete mechanistic account of the internal circuits or features responsible for repetition. Future work could combine SRLA with causal tracing, activation patching, or sparse feature analysis to study how these distributional signatures are represented internally.

Fourth, our mitigation experiments focus on two lightweight training modifications. The conciseness reward targets weak EOS activation, while multi-task diversification targets distributional concentration. These interventions are simple and effective in our setting, but they are not meant to cover the full design space of post-training safeguards. Other approaches, such as explicit EOS calibration, reward regularization, curriculum design, or decoding-time safeguards, may further reduce repetitive degeneration.

Finally, several implementation choices, including the RDR definition, the runaway-window statistic, and the selected decoding configuration, are fixed for consistency across experiments. Changing these choices may affect absolute values, although the main conclusions rely on relative contrasts under the same evaluation protocol. We leave broader evaluations across decoding settings, larger model families, and more diverse application domains to future work.