LACONIC: Length-Aware Constrained Reinforcement Learning for LLMs

Anonymous ACL submission

Abstract

Reinforcement learning (RL) can improve large language models (LLMs), but it often yields excessively long responses that increase latency and cost. Prior length-control methods typically rely on fixed heuristic reward shaping, which can misalign with the underlying task objective and require brittle tuning. We propose `LACONIC`, an RL method that enforces a target token budget during training by augmenting task reward with a length-based cost whose scale is adjusted adaptively. This yields strong length control while preserving task reward, and we provide theoretical support for the method. Across mathematical reasoning models from 1.5B to 32B parameters, `LACONIC` preserves or improves `pass@1` while reducing output length by over 50%. On Qwen3-32B, it lowers macro-average output length from 6431 to 2902 tokens, a 55% reduction, with macro-average `pass@1` changing only from 73.12% to 71.55%. It also maintains out-of-domain performance on general-knowledge and logic benchmarks with 44% fewer tokens. `LACONIC` integrates into standard RL tuning with no inference changes and minimal deployment overhead.

1 Introduction

Large language models (LLMs) such as GPT (OpenAI et al., 2024; OpenAI, 2025), Gemini (Comanici et al., 2025), DeepSeek (DeepSeek-AI, 2025), and Claude (Anthropic, 2025) are increasingly used in applications from software agents to enterprise analytics (Team et al., 2025; Li et al., 2025b; Jin et al., 2025; Feng et al., 2025). Their capabilities have been further enhanced by reinforcement learning based fine-tuning (Li et al., 2025a; Wu et al., 2025; Li et al., 2025b), which aligns pretrained models with task-specific rewards and has been pivotal in improving reasoning, generalization, and benchmark performance (Wang et al., 2024; Hsiao et al., 2025; Shi et al., 2025; Qu et al., 2025). However, RL-tuned language models often generate unnecessarily long thinking traces, especially on reasoning and mathematics tasks (Chen et al., 2025; Sui et al., 2025). In practice, excessive verbosity inflates training and inference time, increases memory pressure, and degrades user experience.

Recent work on length-aware LLMs has explored positional encoding, prompt engineering, and post-generation truncation (Li et al., 2025a; Wu et al., 2025; Li et al., 2025b). A straightforward method is to design new reward functions to incorporate response length signals into RL-tuning (Aggarwal and Welleck, 2025; Cheng et al., 2025; Huang et al., 2025; Yuan et al., 2025). These methods typically hard-code a length penalty or heuristic reward shaping that stays fixed throughout training. Fine-tuning with these rewards optimizes a surrogate objective that is misaligned with true task reward, and often demands per-setting hyperparameter tuning. The top panel in fig. 1 visualizes this by showing a sketch of the training process. The fixed

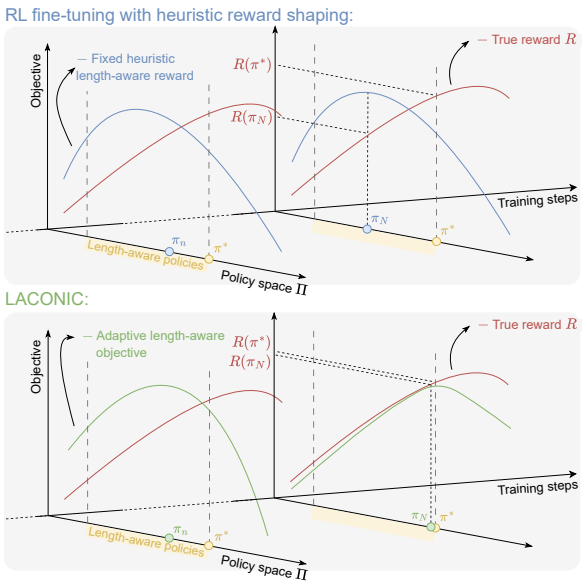


Figure 1: The top panel skethces RL-tuning with a fixed length-aware shaping objective (blue). As the heuristically shaped objective generally differs from the true task reward R (red), optimizing it may converge to a policy π_N that is suboptimal in R. The bottom panel sketches training with `LACONIC`. `LACONIC` adaptively updates the length-aware objective (green) so that it better aligns with the true task reward while achieving shorter outputs, yielding near-optimal policies.

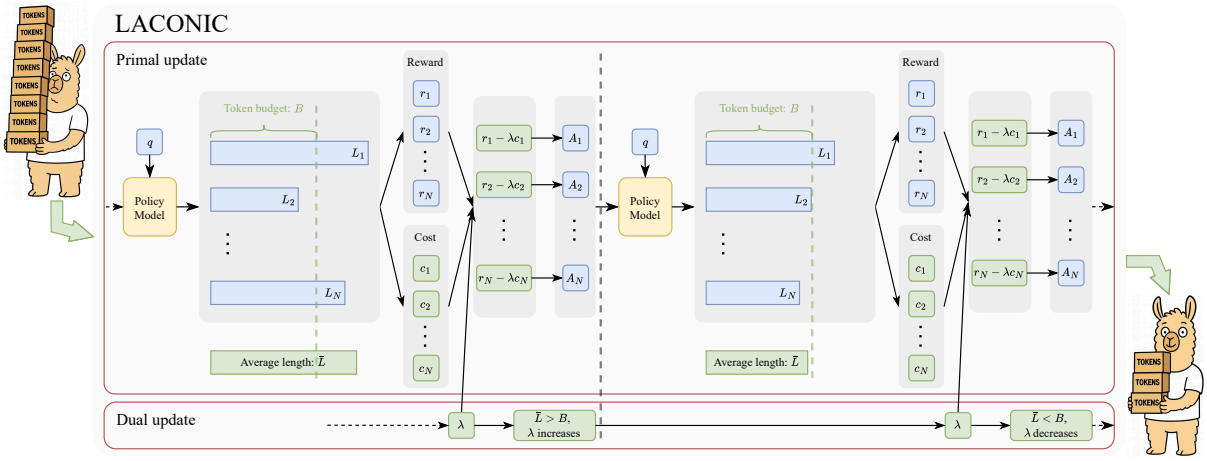


Figure 2: Illustration of LACONIC. LACONIC alternates two steps: (1) in a primal update, the policy model is updated on an augmented objective that trades off task reward r with a length-aware cost c scaled by the dual variable λ; (2) in a dual update, λ is adaptively updated to enforce a token budget constraint B by increasing when the average length $\bar{L}$ exceeds the budget B and decreasing otherwise. Together, these updates maximize task reward while meeting the budget on average.

heuristic objective (blue curve) differs from the true objective (red curve), so optimizing it yields policies with suboptimal task rewards.

In this paper, we address length control in RL-tuning by maximizing task reward subject to an average token budget constraint. We introduce `LACONIC` (**L**ength-**A**ware **Con**strained Pol**ic**y Optimization), a primal-dual algorithm that assigns each sampled response a length-aware cost proportional to its budget violation and combines it with task reward through an adaptively learned multiplier. We alternately update the policy and the multiplier: the policy uses the resulting signal for GRPO-style optimization, and the multiplier rises when the batch average exceeds the budget and falls otherwise. This feedback steers average output length towards the budget; once the model consistently stays within budget, the multiplier naturally relaxes to zero and training approaches standard RL-tuning, allowing the model to recover task rewards with shortened responses. As illustrated in fig. 1, `LACONIC` adopts an adaptive objective that dynamically aligns optimization with true rewards, steering the policy to the length-aware optimum.

We provide theoretical support for our approach: under standard assumptions, the idealized clipped-cost primal-dual dynamics converge to a feasible limit and admit a near-optimality bound.

We conduct extensive experiments to evaluate our method `LACONIC` and present the evaluation results in section 3. On two reasoning base models DeepScaleR-1.5B-Preview (Luo et al., 2025) and DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI et al., 2025), the experimental results show that `LACONIC`-tuned models can significantly outperform existing length control baselines L1 (Aggarwal and Welleck, 2025), (Arora and Zanette, 2025), and ThinkPrune (Hou et al., 2025) on pass@1 across common mathematics benchmarks, while using fewer tokens. `LACONIC` also preserves accuracy on benchmarks outside our RL-tuning domain while substantially reducing response length, and it continues to provide strong length control on larger Qwen3 models up to 32B parameters. Furthermore, we perform ablation analysis on the hyperparameters of our method in section 4, showing that `LACONIC` provides robust length control.

2 Methodology

2.1 Preliminary Background

RL-tuning treats text generation as a sequential decision process, where the language model π_θ chooses the next token and receives a sequence-level reward $r(q, o)$. Standard policy-gradient methods such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) optimize the expected reward

$$\max_\theta \mathbb{E}_{q\sim P(Q), o\sim\pi_\theta(\cdot|q)}[r(q, o)]. \tag{1}$$

In practice, GRPO uses a surrogate objective

$$\mathcal{J}(\theta) = \mathbb{E}_{q,\{o_i\}_{i=1}^G}\Big[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min(\rho_{i,t}A_{i,t}, \mathrm{clip}(\rho_{i,t}, 1-\varepsilon, 1+\varepsilon)A_{i,t}) - \beta D_{\mathrm{KL}}[\pi_\theta\|\pi_{\mathrm{ref}}]\Big]. \tag{2}$$

where $\rho_{i,t}$ is the likelihood ratio and $A_{i,t}$ is the group-relative advantage; clipping and KL regularization stabilize the update.

2.2 Formulation

To explicitly control response lengths, we extend standard RL-tuning to a constrained setting that maximizes task reward under an average token constraint B, a pre-specified budget reflecting deployment targets such as latency and computational resources in practice. Formally, letting $L(o)$ denote the length of response o, we address

$$\begin{aligned} &\max_{\theta} \mathbb{E}_{q\sim P(Q), o\sim\pi_\theta(\cdot|q)}[r(q,o)], \\ &\text{s.t. } \mathbb{E}_{q\sim P(Q), o\sim\pi_\theta(\cdot|q)}[L(o)] \le B. \end{aligned} \tag{3}$$

In eq. (3), we enforce a corpus-level average token budget rather than a strict per-sequence length constraint, as response lengths naturally vary across prompts (e.g., math olympiad problems typically require more tokens than simple arithmetic). This allows allocating more tokens to hard instances while still meeting the overall budget constraint.

A standard approach to eq. (3) is Lagrangian primal-dual optimization. Introduce a dual variable $\lambda \ge 0$ and Lagrangian $\mathcal{L}(\theta,\lambda) := \mathbb{E}[r(q,o)] - \lambda\left(\mathbb{E}[L(o)]/B - 1\right)$, then the constrained problem eq. (3) can be addressed by optimizing the Lagrangian over policy models θ while adjusting $\lambda \ge 0$, i.e., $\max_\theta \min_{\lambda\ge 0} \mathcal{L}(\theta,\lambda)$. Deferring the derivation to appendix A, we obtain the following primal and dual updates:

$$\theta_{t+1} \in \arg\max_{\theta} \mathbb{E}\left[r(q,o) - \lambda_t \frac{L(o)-B}{B}\right], \tag{4}$$

$$\lambda_{t+1} = \max\{\lambda_t + \eta\left(\mathbb{E}[L(o)]/B - 1\right), 0\}. \tag{5}$$

The primal update eq. (4) is analogous to standard RL-tuning in eq. (1). It maximizes expected reward augmented by a linear length-aware cost $\widetilde{c} := (L(o)-B)/B$ weighted by λ. However, this linear cost $\widetilde{c}$ can be problematic in practice: when $\lambda > 0$, the linear cost $\widetilde{c}$ consistently incentivizes shorter outputs, and drives the policy towards extremely short responses. This can cause unstable policy updates in training. We observe this behavior in practice and verify it in the ablation analysis in section 4.1 on DeepScaleR-1.5B. Appendix A provides more detailed analysis and additional experiments on other base models.

2.3 LACONIC: Length-Aware Constrained Policy Optimization

We propose LACONIC, a primal-dual method for length-aware RL tuning that preserves the principled constrained-optimization structure, while resolving the collapse to overly short outputs induced by the linear length cost $\widetilde{c}$.

Clipped cost. We introduce a clipped cost that is zero up to the budget. Specifically, for a prompt q and a response o generated by the policy model π_θ, we define

$$c(q,o) = \max\left\{\frac{L(o)-B}{B}, 0\right\}. \tag{6}$$

This clipped cost assigns equal cost to within-budget responses ($L(o) \le B$). Therefore, it removes artificial pressure to shorten responses that already meet the token budget. This design is empirically validated in section 4.1 and theoretically supported by the convergence and near-optimality analysis summarized below.

Primal updates. In the primal update, we update the policy model θ by holding λ_t fixed and solving eq. (4). This objective has the same form as standard RL-tuning in eq. (1) with a shaped reward. Therefore, we instantiate the primal update using the RL-tuning procedure (e.g., GRPO) by substituting the task reward r with the Lagrangian reward:

$$\ell_\lambda(q,o) = r(q,o) - \lambda \cdot c(q,o), \tag{7}$$

where we replace the linear cost with our clipped cost c to stabilize policy updates.

Specifically, for each prompt q, we sample a group of candidate outputs $\mathbf{o} = \{o_1, o_2, \ldots, o_G\}$ from the current policy model π_θ, and compute their task rewards $\mathbf{r} = \{r_1, r_2, \ldots, r_G\}$ and costs $\mathbf{c} = \{c_1, c_2, \ldots, c_G\}$ by eq. (6). We then compute the Lagrangian rewards $\boldsymbol{\ell}_\lambda = \{\ell_{\lambda,1}, \ell_{\lambda,2}, \ldots, \ell_{\lambda,G}\}$ by eq. (7). For each token $o_{i,t}$, we construct the GRPO-style advantage by normalizing Lagrangian rewards $\ell_{\lambda,i}$ within the group, i.e.,

$$A_{i,t} = \widetilde{\ell}_{\lambda,i} = \frac{\ell_{\lambda,i} - \text{mean}(\boldsymbol{\ell}_\lambda)}{\text{std}(\boldsymbol{\ell}_\lambda)}. \tag{8}$$

The policy model is optimized by maximizing the GRPO objective in eq. (2) where advantages are calculated by eq. (8).

Dual updates. In the dual update, we adjust the dual variable λ by estimating the expectation in eq. (5) with the empirical mean response length $\bar{L}$

Algorithm 1: LACONIC (Length-Aware Constrained Policy Optimization)

Input: initial policy model $\pi_{\theta_{\text{init}}}$; reward models r_φ; task prompts $\mathcal{D}$; token budget B; step size η; initial dual variable λ_{init}

policy model $\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
dual variable $\lambda \leftarrow \lambda_{\text{init}}$
for *iteration* = $1, \ldots, I$ **do**
  reference model $\pi_{\text{ref}} \leftarrow \pi_\theta$
  for *step* = $1, \ldots, M$ **do**
    Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
    Update the old policy model $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
    Sample G outputs $\{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)$ for each question $q \in \mathcal{D}_b$
    Compute rewards $\{r_i\}_{i=1}^G$ for each sampled output o_i by running r_φ
    Compute costs $\{c_i\}_{i=1}^G$ for each sampled output o_i by eq. (6)
    Compute Lagrangian rewards $\{\ell_{\lambda,i}\}_{i=1}^G$ for each sample output by eq. (7)
    Compute advantages $A_{i,t}$ for the t-th token of o_i by eq. (8)
    // Primal update
    Update the policy model π_θ by maximizing the GRPO-style objective
    // Dual update
    Update the dual variable λ by eq. (9)

Output: π_θ

of the current minibatch. This yields the practical dual update

$$\lambda \leftarrow \text{clip}\left(\lambda + \eta\left(\frac{\bar{L}}{B} - 1\right), 0, \Lambda\right), \qquad (9)$$

with the step size η and a λ-ceiling Λ. When the batch violates the token budget on average ($\bar{L} > B$), the update increases λ, raising the effective price of tokens in ℓ_λ. Longer responses then receive lower (often negative) advantages than shorter responses with similar task rewards, so the next primal update shifts the policy toward shorter outputs. When the batch falls within the budget, λ relaxes towards 0. Notably, when $\lambda = 0$, ℓ_λ reduces to the task reward r, and the next primal update is exactly a GRPO step. This feedback adapts λ to track the budget constraint throughout training as the policy and length distribution evolve.

We cap λ by a ceiling Λ to avoid over-penalizing length. For an indicator reward $r(q, o) \in \{0, 1\}$, an excessively large λ can make a within-budget incorrect response (with $\ell_\lambda = 0$) score above a correct but long response. A sufficient safeguard is to require $\ell_\lambda(q, o_c) > 0$ for any correct response o_c, which requires $\lambda < \frac{B}{L(o_c)-B}$ for $L(o_c) > B$. Using the worst case $L(o_c) \leq L_{\max}$, where $L_{\max}$ is the maximum response length cap, it is sufficient to set $\Lambda = \frac{B}{L_{\max}-B}$.

We present LACONIC in algorithm 1 and illustrate the workflow with two sample steps in fig. 2.

Theory summary. Appendix C analyzes an idealized version of LACONIC with exact primal maximization and exact dual expectations. Under standard monotonicity and continuity assumptions on the dual response function, the projected dual iterates converge to a fixed point $\lambda^\sharp$, and the induced limit policy $\pi^\sharp$ satisfies the original average-budget constraint in eq. (3). Moreover, if π^* denotes an optimal feasible policy, then its reward advantage over $\pi^\sharp$ is bounded by $\lambda^\sharp C(\pi^*)$, where $C(\pi^*)$ is the expected above-budget clipped cost of π^*. For indicator rewards with response-length cap $L_{\max}$ and λ-ceiling $\Lambda = \frac{B}{L_{\max}-B}$, this yields a worst-case gap of at most $B/L_{\max}$. In short, clipping removes unnecessary pressure on within-budget responses and stabilizes optimization, while remaining near-optimal when $B \ll L_{\max}$. The cost ablations in section 4.1 are consistent with this picture.

3 Experiment

3.1 Experimental Setup

Models and Datasets. We train on DeepScaleR-Preview-Dataset (Luo et al., 2025), a 40.3K-example math dataset sampled from AIME (pre-2023), AMC (pre-2023), Omni-MATH (Gao et al., 2024), and STILL (Min et al., 2024). Our base models are DeepScaleR-1.5B-Preview (Luo et al., 2025) (**DeepScaleR-1.5B**) and DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI et al., 2025) (**DeepSeek-1.5B**); for a larger-scale study, we also fine-tune Qwen3-32B (Yang et al., 2025).

Baselines. We compare against (i) GRPO (Shao et al., 2024), the RL-tuning algorithm underlying the released base checkpoints, rerun here under the same 4K training cap as LACONIC so that it serves as a matched-cap reference rather than a long-cap

Table 1: Evaluation results across four math benchmarks.

Model	AIME2024		MATH		Minerva		Olympiad		Macro Average	
	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens
DeepScaleR-1.5B	38.75	8140	85.92	3019	27.62	4134	51.94	5410	51.06	5176
+ GRPO	29.58	6122	85.28	1767	29.64	2630	**49.11**	3418	48.40 ↓2.66	3484 ↓33%
+ L1-Exact	22.02	4138	82.57	3734	28.74	3755	44.91	3987	44.56 ↓6.50	3904 ↓24%
+ L1-Max	25.12	**2979**	83.45	1912	28.35	**1701**	44.76	**2334**	45.42 ↓5.64	**2231** ↓57%
+ Efficient-Reasoning	32.71	6440	82.01	1585	27.11	1852	46.95	3230	47.19 ↓3.87	3277 ↓37%
+ ThinkPrune-Iter2k	**35.00**	4598	84.22	1802	29.24	1941	48.77	2935	49.71 ↓1.35	2819 ↓45%
+ LACONIC (B=2000)	34.79	3976	**85.61**	**1556**	**31.76**	1869	48.96	2448	**50.28** ↓0.78	2462 ↓52%
DeepSeek-1.5B	30.00	15218	82.90	5340	29.13	6783	44.80	12417	46.71	9940
+ GRPO	**25.83**	9090	81.57	2403	27.20	2655	43.52	5420	44.53 ↓2.18	4892 ↓51%
+ Efficient-Reasoning	22.08	8801	81.63	1909	27.57	2018	42.96	4799	43.56 ↓3.15	4382 ↓56%
+ ThinkPrune-Iter2k	23.33	5440	81.35	1762	27.35	1837	43.39	**3021**	43.85 ↓2.86	3015 ↓70%
+ LACONIC (B=1500)	24.17	**5140**	**81.88**	**1636**	**28.36**	**1638**	**44.11**	3241	**44.63** ↓2.08	**2914** ↓71%

Table 2: Scalability results on Qwen3-32B across four math benchmarks.

Methods	AIME2024		MATH		Minerva		Olympiad		Macro Average	
	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens
Qwen3-32B	80.00	13355	95.80	3855	46.32	4294	70.37	9220	73.12	6431
+ LACONIC (B=3000)	77.50	5072	95.20	1596	47.06	1418	66.44	3521	71.55 ↓1.57	2902 ↓55%

GRPO setting; (ii) L1-Exact and L1-Max (Aggarwal and Welleck, 2025); (iii) Efficient-Reasoning (Arora and Zanette, 2025); and (iv) ThinkPrune-Iter (Hou et al., 2025). We also report direct evaluation of the released base checkpoints as long-cap reference points.

Training. We train DeepScaleR-1.5B with $B = 2000$ for 300 steps, DeepSeek-1.5B with $B = 1500$ for 500 steps, and Qwen3-32B with $B = 3000$ for 300 steps. All training runs use a 4K response-length cap.

Evaluation. We evaluate on AIME2024, MATH (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and Olympiad-Bench (He et al., 2024), reporting pass@1 and average response length. All checkpoints are evaluated under the same 32K inference cap.

3.2 Main Results

We first present the main math results in table 1, then the Qwen3-32B larger-scale result in table 2, and finally the out-of-domain results in table 3.

LACONIC outperforms existing length-control methods and matches vanilla RL-tuning while significantly reducing response lengths. On DeepScaleR-1.5B, LACONIC achieves 50.28 macro-average pass@1 with 2462 tokens, outperforming all baselines. Existing length-control methods either lose substantially more accuracy or use noticeably more tokens at similar accuracy. Relative to the matched-cap GRPO baseline, LACONIC improves macro-average pass@1 from 48.40 to 50.28 while reducing average length from 3484 to 2462 tokens. Relative to the released base checkpoint, LACONIC preserves nearly the same pass@1 while using 52% fewer tokens.

On DeepSeek-1.5B, LACONIC again attains the best macro-average pass@1 among the compared methods and the lowest token count. It slightly improves over matched-cap GRPO in macro-average pass@1 (44.63 vs. 44.53) while reducing average length from 4892 to 2914 tokens, and it cuts response length by 71% relative to the released base checkpoint with a 2.08-point pass@1 decrease.

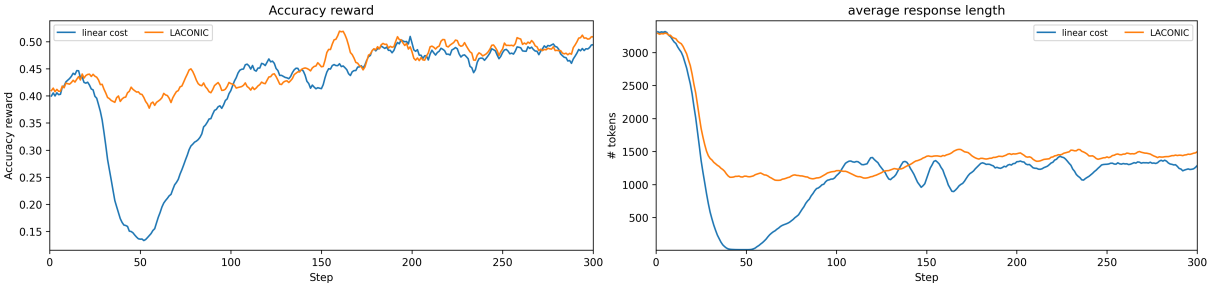
LACONIC scales to larger models. On Qwen3-32B, trained with $B = 3000$ for 300 steps under the same 4K training cap, LACONIC reduces macro-average output length from 6431 to 2902 tokens (55%) while macro-average pass@1 changes from 73.12 to 71.55. The same direction holds on all four benchmarks, reflecting a robust accuracy-length trade-off at 32B scale. This provides direct evidence that the method continues to deliver large token savings with limited accuracy loss at

Table 3: Evaluation results across out-of-domain (OOD) benchmarks.

Model	GPQA		LSAT		MMLU		Macro Average	
	Accuracy	# Tokens	Accuracy	# Tokens	Accuracy	# Tokens	Accuracy	# Tokens
DeepScaleR-1.5B								
+ GRPO	**28.88**	2229	24.6	3426	43.30	719	**32.26**	2125
+ L1-Exact	22.88	1475	25.19	**1374**	36.67	766	28.25 ↓4.01	1205 ↓43%
+ L1-Max	28.72	1773	**26.58**	2321	38.48	863	31.26 ↓1.00	1652 ↓22%
+ ThinkPrune-Iter2k	27.02	1791	24.82	3244	42.42	611	31.42 ↓0.84	1882 ↓11%
+ **LACONIC**	27.20	**1167**	24.53	1923	**43.36**	**497**	31.69 ↓0.57	**1196** ↓44%

Table 4: Evaluation results of the cost function ablation on DeepScaleR-1.5B across four math benchmarks.

Methods	AIME2024		MATH		Minerva		Olympiad		Macro Average	
	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens
DeepScaleR-1.5B	38.75	8140	85.92	3019	27.62	4134	51.94	5410	51.06	5176
+ GRPO	29.58	6122	**85.28**	1767	29.64	2630	**49.11**	3418	**48.40** ↓2.66	3484 ↓33%
+ Linear cost (1500)	29.17	5636	83.00	1593	29.14	2638	48.18	3256	47.37 ↓3.69	3281 ↓37%
+ LACONIC (1500)	**29.58**	**3458**	84.35	**1353**	**31.48**	**1582**	47.93	**2084**	48.33 ↓2.73	**2119** ↓59%



(a) Accuracy reward over training steps

(b) Average response length over training steps

Figure 3: Ablation of the cost functions on DeepScaleR-1.5B with token budget $B = 1500$. We plot (a) accuracy reward and (b) average response length over training steps. For the linear cost, the Langrangian reward used in primal updates is computed by $r(q, o) - \lambda \widetilde{c}(q, o)$, where $\widetilde{c}(q, o) = (L(o) - B)/B$. All other experiment setups and hyperparameters are identical.

a substantially larger scale. Appendix B.2 reports Qwen3-8B results with broader baseline comparisons, providing intermediate-scale corroboration between the 1.5B and 32B studies.

LACONIC preserves out-of-domain (OOD) capabilities. LACONIC remains within 0.57 macro-average accuracy points of GRPO while using 44% fewer tokens, showing that the learned brevity transfers beyond the math domain. It also surpasses both L1 variants and ThinkPrune-Iter2k in macro accuracy while using fewer tokens.

4 Further Analysis

We next examine cost design, hyperparameter sensitivity, and computational resources.

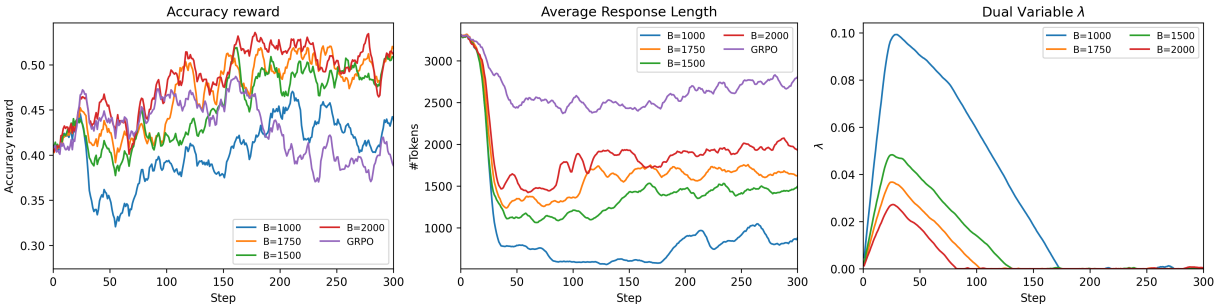
4.1 Ablation Analysis on Cost Function

We isolate the cost design by comparing the linear cost $\widetilde{c}(q, o) = \frac{L(o)-B}{B}$ with the clipped cost c used by LACONIC.

We change only the cost used in the GRPO-style primal updates while keeping all other settings fixed; fig. 3 and table 4 reports the results. The linear-cost baseline collapses early to extremely short, degenerate outputs, matching the failure mode suggested by section 2.2: once $\lambda > 0$, it keeps pressuring even within-budget responses to shorten. By pricing only excess tokens, the clipped cost in LACONIC yields stable dynamics and a better final trade-off: 48.33 macro pass@1 with 2119 tokens versus 47.37 with 3281 tokens for the linear-cost variant. Taken together, these ablations show

Table 5: Evaluation results of the token-budget B ablation on DeepScaleR-1.5B across four math benchmarks.

Token Budget	AIME2024		MATH		Minerva		Olympiad		Macro Average	
	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens
+ LACONIC (2000)	34.79	3976	85.61	1556	31.76	1869	48.96	2448	50.28	2462
+ LACONIC (1750)	31.21	3803	84.82	1516	31.52	2059	48.88	2261	49.11	2410
+ LACONIC (1500)	29.58	3458	84.35	1353	31.48	1582	47.93	2084	48.33	2119
+ LACONIC (1000)	28.12	2665	83.75	1049	29.24	1189	47.07	1669	47.05	1643



(a) Accuracy reward (b) Average response length (c) Dual variable λ

Figure 4: Ablation of the token budget B on DeepScaleR-1.5B. We plot (a) accuracy reward; (b) average response length; and (c) dual variable λ over training steps with budgets $B \in \{1000, 1500, 1750, 2000\}$. All other setups and hyperparameters are identical.

that clipped cost is experimentally motivated, theoretically justified by section 2.3 and appendix C, and practically effective.

4.2 Ablation Analysis on Budget B

We vary the token budget $B \in \{2000, 1750, 1500, 1000\}$ on DeepScaleR-1.5B with all other settings fixed. Figure 4 shows training dynamics, and table 5 reports the step-300 checkpoints.

LACONIC provides reliable, tuning-free length control. Across token budgets, training rapidly drives the average response length near the target budget and keeps it there after stabilization. In practice, B acts as a single control knob and no re-tuning of other hyperparameters is required.

LACONIC preserves reward across a wide range of token budgets. Compared with the released base checkpoints and baselines in table 1, LACONIC with budgets from 1K to 2K consistently matches or exceeds baseline reward while using substantially fewer tokens. Some mismatch between the training budget and evaluation-time average lengths is expected because the prompt distribution differs between the training corpus and the held-out benchmarks. Even so, table 5 shows a clear monotonic relation between the chosen budget and the realized response length at evaluation. This indicates that the model is not merely reacting to a particular training batch distribution; it learns a stable budget-conditioned brevity preference that transfers across benchmarks.

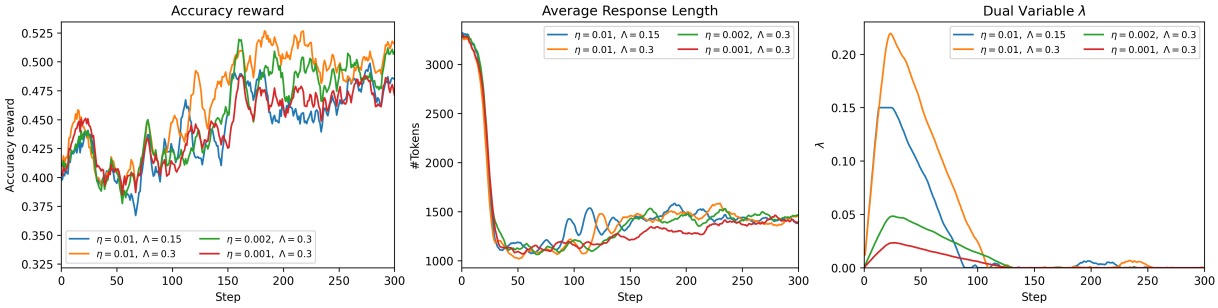
4.3 Ablation Analysis on Dual Step Size η

We vary the step size for dual updates $\eta \in \{0.001, 0.002, 0.01\}$ with all other settings fixed and train DeepScaleR-1.5B for with $B = 1500$. For $\eta = 0.01$, we also set a low λ-ceiling $\Lambda = 0.1$. Figure 5 plots the training dynamics of (a) accuracy reward; (b) average response length; and (c) dual variable λ. We evaluate the step-300 checkpoints on the 4 mathematics benchmarks, and report the pass@1 and average response lengths in appendix B.1.

LACONIC is insensitive to the dual step size η. Across different dual step sizes, LACONIC delivers consistent length control. With $\eta = 0.002$ versus 0.01, the training dynamics are similar after stabilization. All trained checkpoints reach comparable final reward. This shows that LACONIC is robust to an order-of-magnitude change in the step size η.

4.4 Computational Resource Analysis

We report in fig. 6 wall-clock step time, step time per token, and NVML GPU memory, averaged over training steps. LACONIC is end-to-end cheaper than vanilla RL-tuning. Our method is 19% faster and uses 22% less GPU memory. Per-token cost is



(a) Accuracy reward (b) Average response length (c) Dual variable λ

Figure 5: Ablation of the dual step size η on DeepScaleR-1.5B with $B = 1500$. We plot (a) accuracy reward; (b) average response length; and (c) dual variable λ over training steps with step size $\eta \in \{0.001, 0.002, 0.01\}$. We set $\Lambda = 0.3$ for all experiments except for the run with $\eta = 0.01$ and $\Lambda = 0.15$. All other setups and hyperparameters are identical.

nearly unchanged, with a small bookkeeping overhead for the length cost and dual update. Overall, LACONIC adds negligible kernel-level overhead and reduces runtime and memory by generating fewer tokens.

5 Related Work

RL fine-tuning. Reinforcement learning is now a standard ingredient in LLM post-training, especially for reasoning and alignment. Starting from policy gradients (Sutton and Barto, 2018; Williams, 1992; Schulman et al., 2017), recent work has made these methods more practical for LLMs, including GRPO and variants such as SRPO, DAPO, VAPO, GSPO, GFPO, GMPO, and Dr. GRPO (Shao et al., 2024; Zhang et al., 2025b; Yu et al., 2025; Yue et al., 2025; Zheng et al., 2025; Shrivastava et al., 2025; Zhao et al., 2025; Liu et al., 2025). Our method builds on this line by adding explicit budget control to GRPO-style tuning.

Length-Aware LLMs. Prior work controls response length through prompting, truncation, architectural changes, or heuristic reward shaping (Li et al., 2025a; Wu et al., 2025; Li et al., 2025b; Aggarwal and Welleck, 2025; Cheng et al., 2025; Huang et al., 2025; Yuan et al., 2025; Arora and Zanette, 2025; Hou et al., 2025). The closest RL approaches use fixed length penalties, which can misalign the training objective with task reward and

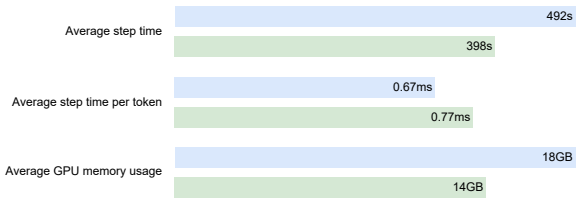


Figure 6: Average computational resource usage of LACONIC (green) and GRPO (blue).

require retuning across settings. LACONIC instead learns the penalty adaptively through a constrained optimization view.

Constrained RL. Constrained RL provides standard tools for enforcing auxiliary requirements during learning (Achiam et al., 2017; Tessler et al., 2018; Stooke et al., 2020). Recent LLM work has begun to adopt this perspective for post-training, for example to compress inference trajectories or control policy-update divergence (Tzannetos et al., 2025; Zhang et al., 2025a). We apply the same lens to response-length control in RL fine-tuning.

6 Conclusion

We introduced LACONIC, a constrained RL fine-tuning method that turns response-length control into explicit token-budget optimization rather than fixed heuristic reward shaping. LACONIC combines a clipped length cost with adaptive dual updates: the clipped cost avoids penalizing responses that already satisfy the budget, while the dual variable automatically adjusts the strength of the length penalty based on the model's average response length.

Across two 1.5B reasoning backbones, out-of-domain evaluations, and a 32B study, LACONIC substantially reduces token usage while staying close in pass@1 to matched-cap GRPO or the released base checkpoints. Our theory and ablations support the key design choices: clipping prevents destabilizing pressure on within-budget responses, and adaptive dual updates provide a practical control knob for targeting different budgets without extensive retuning. Together, these results establish LACONIC as a practical and principled approach to token-budget control in RL-based LLM post-training.

7 Limitations

While LACONIC is effective and lightweight, it has several limitations. It currently enforces a global average token budget, which may not capture prompt-specific or context-dependent needs. Our experiments are limited to math reasoning, and future work could validate generality on dialogue, summarization, or code. Finally, LACONIC handles a single constraint, but the framework naturally extends to multi-constraint settings such as latency or safety.

Ethical Considerations

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

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained policy optimization. *Preprint*, arXiv:1705.10528.

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *Preprint*, arXiv:2503.04697.

Anthropic. 2025. Claude opus 4.1. https://www.anthropic.com/news/claude-opus-4-1. Accessed: 2025-08-06.

Daman Arora and Andrea Zanette. 2025. Training language models to reason efficiently. *Preprint*, arXiv:2502.04463.

Amir Beck. 2017. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Do not think that much for 2+3=? on the overthinking of o1-like llms. *Preprint*, arXiv:2412.21187.

Xiaoxue Cheng, Junyi Li, Zhenduo Zhang, Xinyu Tang, Wayne Xin Zhao, Xinyu Kong, and Zhiqiang Zhang. 2025. Incentivizing dual process thinking for efficient large language model reasoning. *Preprint*, arXiv:2505.16315.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

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

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *Preprint*, arXiv:2410.07985.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

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

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*.

Vincent Hsiao, Morgan Fine-Morris, Mark Roberts, Leslie N Smith, and Laura M. Hiatt. 2025. A critical assessment of LLMs for solving multi-step problems: Preliminary results. In *AAAI 2025 Workshop LM4Plan*.

Chengyu Huang, Zhengxin Zhang, and Claire Cardie. 2025. Hapo: Training language models to reason concisely via history-aware policy optimization. *Preprint*, arXiv:2505.11225.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and

leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025a. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025b. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *Preprint*, arXiv:2503.20783.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog.

Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *Preprint*, arXiv:2412.09413.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

OpenAI. 2025. Gpt-5 system card. https://cdn.openai.com/gpt-5-system-card.pdf. Accessed: 2025-08-13.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-rong Wen. 2025. Tool learning with large language models: a survey. *Frontiers of Computer Science*, 19(8).

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Zhengliang Shi, Shen Gao, Lingyong Yan, Yue Feng, Xiuyi Chen, Zhumin Chen, Dawei Yin, Suzan Verberne, and Zhaochun Ren. 2025. Tool learning in the wild: Empowering language models as automatic tool agents. *Preprint*, arXiv:2405.16533.

Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. 2025. Sample more to think less: Group filtered policy optimization for concise reasoning. *Preprint*, arXiv:2508.09726.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. 2020. Responsive safety in reinforcement learning by pid lagrangian methods. *Preprint*, arXiv:2007.03964.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *Preprint*, arXiv:2503.16419.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*, second edition. The MIT Press.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. 2018. Reward constrained policy optimization. *Preprint*, arXiv:1805.11074.

Georgios Tzannetos, Parameswaran Kamalaruban, and Adish Singla. 2025. Curriculum design for trajectory-constrained agent: Compressing chain-of-thought tokens in llms. *Preprint*, arXiv:2511.02690.

Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. 2024. Q*: Improving multi-step reasoning for llms with deliberative planning. *Preprint*, arXiv:2406.14283.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. 2025. Agentic reasoning: A streamlined framework for enhancing llm reasoning with agentic tools. *arXiv preprint arXiv:2502.04644*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others.

2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Danlong Yuan, Tian Xie, Shaohan Huang, Zhuocheng Gong, Huishuai Zhang, Chong Luo, Furu Wei, and Dongyan Zhao. 2025. Efficient rl training for reasoning models via length-aware optimization. *Preprint*, arXiv:2505.12284.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, and 8 others. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *Preprint*, arXiv:2504.05118.

Botong Zhang, Shuo Li, Ignacio Hounie, Osbert Bastani, Dongsheng Ding, and Alejandro Ribeiro. 2025a. Alignment of large language models with constrained learning. *Preprint*, arXiv:2505.19387.

Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, Shimiao Jiang, Shiqi Kuang, Shouyu Yin, Chaohang Wen, Haotian Zhang, Bin Chen, and Bing Yu. 2025b. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *Preprint*, arXiv:2504.14286.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, and Furu Wei. 2025. Geometric-mean policy optimization. *Preprint*, arXiv:2507.20673.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. Group sequence policy optimization. *Preprint*, arXiv:2507.18071.

A Standard Primal-Dual Method and Linear Cost Functions

A.1 Standard Primal-Dual Updates

In this section, we derive the standard primal-dual updates. For compactness in this subsection, write

$$\mathbb{E}_\theta[\cdot] := \mathbb{E}_{q\sim P(Q),\, o\sim\pi_\theta(\cdot|q)}[\cdot].$$

Recall the constrained optimization problem of length-aware LLMs formulated in eq. (3):

$$\begin{aligned} &\max_\theta \mathbb{E}_\theta[r(q,o)], \\ &\text{s.t. } \mathbb{E}_\theta[L(o)] \le B. \end{aligned} \tag{10}$$

The corresponding Lagrangian is

$$\mathcal{L}(\theta,\lambda) = \mathbb{E}_\theta[r(q,o)] - \lambda\left(\frac{\mathbb{E}_\theta[L(o)]}{B} - 1\right). \tag{11}$$

where $\lambda \ge 0$. and the problem is commonly approached by solving the saddle point problem:

$$\max_\theta \min_{\lambda\ge 0} \mathcal{L}(\theta,\lambda), \tag{12}$$

where θ is the primal variable (in our case, the policy model) and λ is the dual variable. We assume that eq. (12) is feasible, and denote the optimal feasible solution by (π^*, λ^*). The standard primal-dual approach solves π^* and λ^* iteratively with partial derivatives. In the primal step, the dual variable λ_t is fixed, and we optimize $\mathcal{L}(\theta,\lambda_t)$ over θ:

$$\theta_{t+1} \in \arg\max_\theta \mathbb{E}_\theta[r(q,o)] - \lambda_t\left(\frac{\mathbb{E}_\theta[L(o)]}{B} - 1\right).$$

By linearity of expectations, we have

$$\theta_{t+1} \in \arg\max_\theta \mathbb{E}_\theta\left[r(q,o) - \lambda_t\frac{L(o)-B}{B}\right].$$

Define the linear cost function

$$\widetilde{c}(q,o) = \frac{L(o)-B}{B}, \tag{13}$$

and thus the primal update takes the form

$$\theta_{t+1} \in \arg\max_\theta \mathbb{E}_\theta[r(q,o) - \lambda_t\widetilde{c}(q,o)]. \tag{14}$$

In the dual update, we decrease $\mathcal{L}(\theta_{t+1},\lambda)$ by taking a gradient-descent step for λ:

$$\begin{aligned} \lambda_{t+1} &\leftarrow \max\left\{\lambda_t - \eta\frac{\partial\mathcal{L}(\theta_{t+1},\lambda)}{\partial\lambda}, 0\right\} \\ &= \max\left\{\lambda_t + \eta\left(\frac{\mathbb{E}_{\theta_{t+1}}[L(o)]}{B} - 1\right), 0\right\}. \end{aligned} \tag{15}$$

Thus far, we have derived the idealized standard primal dual updates in eqs. (4) and (5). Empirically we estimate $\mathbb{E}_{\theta_t}[L(o)]$ with the minibatch mean $\bar{L}$ generated by the policy model π_θ, and with an extra λ-ceiling Λ, we derive exactly eq. (9):

$$\lambda \leftarrow \text{clip}\left(\lambda + \eta\left(\frac{\bar{L}}{B} - 1\right), 0, \Lambda\right).$$

For the primal update in eq. (14), we treat it as a RL-tuning step where the task reward is substituted with a Langrangian reward $\widehat{\ell}_{\lambda_t}(q,o) := r(q,o) - \lambda_t\widetilde{c}(q,o)$. We refer to the aforementioned method as the standard primal-dual updates (with linear costs). Note that `LACONIC` shares the same dual updates as the standard primal-dual method, while changing the cost function used in primal updates.

A.2 Additional Cost Function Ablations

As discussed in sections 2.2 and 4.1, using the linear cost function $\widetilde{c}(q,o) = \frac{L(o)-B}{B}$ inside the primal (model) update is problematic in practice. The main text reports the corresponding DeepScaleR-1.5B training dynamics and final evaluation results in fig. 3 and table 4. Here we provide additional supplementary evidence on another backbone.

We then include additional ablations of cost functions on another base model, Qwen2.5-Math-1.5B-Instruct (Yang et al., 2024) (**Qwen-Math-1.5B** for short). Qwen-Math-1.5B is a 1.5B-parameter instruction-tuned math model. In fig. 7, we plot the training dynamics, and in table 6, we present the evaluation results of the step-350 checkpoints. The experiment shows that with linear cost function $\frac{L(o)-B}{B}$, the average accuracy plummets from 40% to 8%, and the mean response lengths fall below 10 tokens, indicating that the model is no longer producing meaningful responses during training. Although as shown in table 6, the primal-dual framework with linear cost function $\widetilde{c}$ recovers the model's performance after the model is stabilized, the unstable update steps can introduce great risk and instability to LLM fine-tuning.

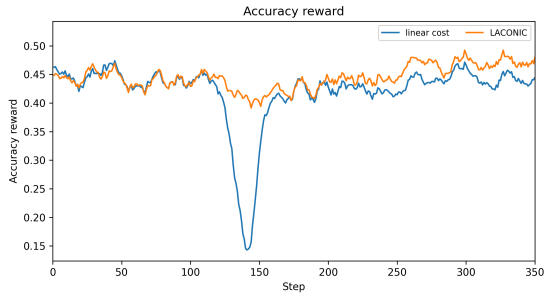
B Additional Experiments

B.1 Ablation Results on Step Sizes

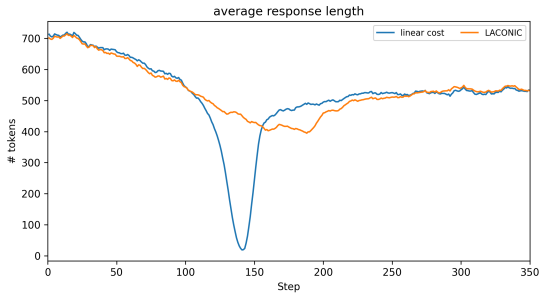
In section 4.3, we introduced the ablation experiments on DeepScaleR-1.5B by varying step sizes and λ-ceiling, and presented training dynamics. In this section, we show the evaluation results of the step-300 checkpoints of all runs.

Table 6: Evaluation results of the cost functions on Qwen-Math-1.5B across four math benchmarks.

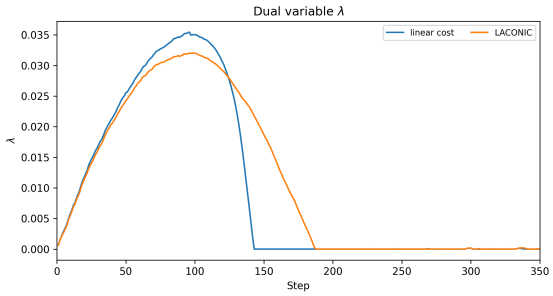
Model	AIME2024		MATH		Minerva		Olympiad		Macro Average	
	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens
Qwen-Math-1.5B	11.46	952	**74.86**	570	25.14	656	**39.84**	808	**37.89**	747
+ Linear cost (550)	11.35	**665**	73.42	479	**25.53**	531	38.42	627	37.18 ↓0.71	576 ↓23%
+ LACONIC (550)	**11.46**	674	73.85	**464**	25.25	**524**	38.39	**603**	37.24 ↓0.65	**566** ↓24%



(a) Accuracy reward over training steps



(b) Average response length over training steps



(c) Dual variable λ over training steps

Figure 7: Ablation of the cost functions on Qwen-Math-1.5B with a token budget $B = 550$: (a) reward accuracy, (b) average response length, (c) dual variable λ,

B.2 Experiments on Qwen3-8B

To test the effectiveness of our method on large models, we train Qwen3-8B on LACONIC with a token budget $B = 1000$ and 4K response length cap for 150 steps, and we present the evaluation results in table 8.

The experimental results show that LACONIC reduces response length by 46% while incurring a modest 2.77% change in pass@1 compared with vanilla GRPO fine-tuning. Moreover, LACONIC outperforms both L1 variants while using fewer tokens.

B.3 Ablation Analysis on Fixed Dual Variables

To isolate the effect of online dual adaptation, we replace LACONIC's dual update with a fixed λ throughout training and sweep λ from $\{0.001, 0.01, 0.05\}$. Table 9 reports pass@1 and average output length under identical training settings and the same token budget $B = 1000$. The sweep reveals strong sensitivity to λ: small λ behaves similarly to the base model, yielding limited length reduction, whereas large λ over-emphasizes the length penalty, producing short outputs and clear accuracy degradation. In contrast, LACONIC's adaptive λ achieves better pass@1 at substantially lower length, without per-task λ tuning, underscoring that online multiplier adaptation is essential for stable and controllable length reduction.

C Formal Analysis of Clipped-Cost Primal-Dual Updates

This appendix gives the formal statements and proofs behind the theory summary in the main paper. We analyze an idealized setting in which the primal step exactly solves the clipped-cost objective and the dual update uses exact expectations.

We first restate some definitions and notations.

Table 7: Evaluation results of the token-budget B ablation across four math benchmarks.

Step size η	**AIME2024**		**MATH**		**Minerva**		**Olympiad**		**Macro Average**	
	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens
$\eta = 0.001, \Lambda = 0.3$	27.50	3231	84.25	1356	30.92	1543	47.87	2090	47.64	2055
$\eta = 0.002, \Lambda = 0.3$	29.58	3458	84.35	1353	31.48	1582	47.93	2084	48.33	2119
$\eta = 0.01, \Lambda = 0.3$	29.79	3429	84.30	1365	31.42	1602	47.19	2097	48.18	2123
$\eta = 0.01, \Lambda = 0.15$	29.79	3477	84.20	1319	31.15	1562	47.83	2067	48.24	2106

Table 8: Evaluation results of length-control on Qwen3-8B across four math benchmarks.

Methods	**AIME2024**		**MATH**		**Minerva**		**Olympiad**		**Macro Average**	
	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens
Qwen3-8B										
+ GRPO	37.50	2918	87.95	1532	40.14	1549	54.40	2167	55.00	2042
+ L1-Exact	27.92	1637	86.37	987	37.50	800	48.54	1254	50.08 ↓4.92	1170 ↓43%
+ L1-Max	30.33	1936	87.30	865	37.59	759	51.11	1209	51.58 ↓3.42	1192 ↓42%
+ LACONIC (1000)	30.42	1643	87.01	792	38.46	783	53.05	1166	52.23 ↓2.77	1096 ↓46%

Let $q \sim P(Q)$ denote a prompt and $o \sim \pi(\cdot \mid q)$ be a response. Let $L(o)$ be the length of response o and $B > 0$ be the token budget. For compactness in this section, let

$$\mathbb{E}_\pi[\cdot] := \mathbb{E}_{q\sim P(Q),\, o\sim\pi(\cdot|q)}[\cdot].$$

Define

$$R(\pi) := \mathbb{E}_\pi[r(q,o)],$$
$$\widetilde{C}(\pi) := \mathbb{E}_\pi\left[\frac{L(o)-B}{B}\right],$$
$$C(\pi) := \mathbb{E}_\pi\left[\frac{(L(o)-B)_+}{B}\right],$$

where $(x)_+ = \max\{x, 0\}$. The target constrained problem is

$$\pi^\star \in \arg\max_\pi R(\pi) \quad \text{s.t.} \quad \widetilde{C}(\pi) \le 0.$$

C.1 Idealized clipped-cost dynamics

For any $\lambda \ge 0$, define the optimal policy under the clipped-cost objective as

$$\pi(\lambda) \in \arg\max_\pi R(\pi) - \lambda\, C(\pi), \tag{16}$$

and let

$$\mu(\lambda) := \widetilde{C}(\pi(\lambda)). \tag{17}$$

We study the deterministic projected updates

$$\pi_t = \pi(\lambda_t), \tag{18}$$

$$\lambda_{t+1} = \mathrm{clip}(\lambda_t + \eta\, \mu(\lambda_t), 0, \Lambda), \tag{19}$$

and let $(\pi^\sharp, \lambda^\sharp)$ denote a limit pair when these dynamics converge, with $\pi^\sharp = \pi(\lambda^\sharp)$.

C.2 Bounding Reward Suboptimality of Clipped-Cost

Lemma C.1. *Denote the response length cap by $L_{\max}$, and $0 \le L(o) \le L_{\max}$ for any response o. We also require $L_{\max} > B$, otherwise the problem is trivial. Let π be any feasible policy for the linear constraint $\widetilde{C}(\pi) \le 0$, i.e., $\mathbb{E}_\pi[L(o)] \le B$. Then*

$$C(\pi) = \mathbb{E}_\pi\left[\frac{(L(o)-B)_+}{B}\right] \le \frac{L_{\max}-B}{L_{\max}}. \tag{20}$$

Proof. We first notice that for any response o,

$$(L(o)-B)_+ \le \frac{L_{\max}-B}{L_{\max}} L(o). \tag{21}$$

When $L(o) < B$, then LHS $< 0 <$ RHS. When $B \le L(o) \le L_{\max}$, then

$$(L(o)-B)_+ = L(o) - B \le \frac{L_{\max}-B}{L_{\max}} L(o),$$

where the inequality can be verified by arithmetic rearranging.

Then by taking expecations on both sides of eq. (21), we have

$$\mathbb{E}_\pi[(L(o)-B)_+] \le \frac{L_{\max}-B}{L_{\max}} \cdot \mathbb{E}_\pi[L(o)].$$

Since $\mathbb{E}_\pi[L(o)] \le B$ for the feasible policy π, we conclude that

$$C(\pi) = \mathbb{E}_\pi\left[\frac{(L(o)-B)_+}{B}\right] \le \frac{L_{\max}-B}{L_{\max}}.$$

□

Table 9: Evaluation results of fixed λ ablation across four math benchmarks.

Model	AIME2024		MATH		Minerva		Olympiad		Macro Average	
	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens	Pass@1	# Tokens
DeepScaleR-1.5B	38.75	8140	85.92	3019	27.62	4134	51.94	5410	51.06	5176
+ GRPO ($\lambda = 0$)	29.58	6122	85.28	1767	29.64	2630	49.11	3418	48.40 ↓2.66	3484 ↓33%
+ $\lambda = 0.001$	27.92	5834	82.73	2089	29.64	2200	46.76	3663	46.76 ↓4.3	3396 ↓34%
+ $\lambda = 0.01$	26.25	4709	81.65	1927	28.90	1990	45.52	2963	45.58 ↓5.48	2897 ↓44%
+ $\lambda = 0.05$	21.66	2417	78.28	1018	24.92	1073	41.25	1602	41.53 ↓9.53	1528 ↓70%
+ **LACONIC**	28.12	2665	83.75	1049	29.24	1189	47.07	1669	47.05 ↓4.01	1643 ↓68%

Theorem C.2 (Price of clipped cost)**.** *Let $\pi^\star \in \max_{\pi:\widetilde{C}(\pi)\le 0} R(\pi)$ be an optimal feasible policy of the length-constrained problem in eq. (3). Let $(\pi^\sharp, \lambda^\sharp)$ be the feasible limit of the idealized clipped-cost primal-dual updates in eqs. (18) and (19). Then*

$$0 \le R(\pi^\star) - R(\pi^\sharp) \le \lambda^\sharp C(\pi^\star). \tag{22}$$

Moreover, for indicator rewards with the λ-ceiling $\Lambda = \frac{B}{L_{\max}-B}$ and a maximum length cap $L(o) \le L_{\max}$, we have

$$0 \le R(\pi^\star) - R(\pi^\sharp) \le \frac{B}{L_{\max}}. \tag{23}$$

Proof. Since $\widetilde{C}(\pi^\sharp) \le 0$, $\pi^\sharp$ is feasible for the clipped-cost primal-dual optimization, and therefore feasible for the original constrained optimization. By optimality of $\pi^\star$,

$$R(\pi^\star) \ge R(\pi^\sharp),$$

which gives the left inequality.

For the upper bound, optimality of $\pi^\sharp$ for the clipped-cost objective implies that for any policy π,

$$R(\pi^\sharp) - \lambda^\sharp C(\pi^\sharp) \ge R(\pi) - \lambda^\sharp C(\pi).$$

Let π be $\pi^\star$ and rearranging,

$$\begin{aligned} R(\pi^\star) - R(\pi^\sharp) &\le \lambda^\sharp C(\pi^\star) \\ &\quad - \lambda^\sharp C(\pi^\sharp). \end{aligned}$$

Since $C(\pi^\sharp) \ge 0$, dropping the negative term, we conclude that

$$R(\pi^\star) - R(\pi^\sharp) \le \lambda^\sharp C(\pi^\star).$$

Thus, we have proved eq. (22). Finally, since

$$\lambda^\sharp \le \Lambda \le \frac{B}{L_{\max} - B},$$

and since $\pi^\star$ is feasible, we apply Theorem C.1, and obtain eq. (23):

$$\begin{aligned} R(\pi^\star) - R(\pi^\sharp) &\le \lambda^\sharp C(\pi^\star) \\ &\le \frac{B}{L_{\max} - B} \cdot \frac{L_{\max} - B}{L_{\max}} \\ &= \frac{B}{L_{\max}}. \end{aligned}$$

□

C.3 Convergence Guarantee

We impose the following assumptions on the scalar response function $\mu(\lambda) = \widetilde{C}(\pi(\lambda))$.

A1. For any $\lambda \in [0, \Lambda]$, $\pi(\lambda)$ exists.

A2 (Monotonicity of μ). The response function μ is nonincreasing and $\mu(\Lambda) \le 0$.

The monotonicity assumption states that as the dual variable λ increases, the corresponding optimal policy that maximizes $R(\pi) - \lambda C(\pi)$ will not generate longer responses in expectation. Also, the budget can be enforced without needing $\lambda > \Lambda$.

A3 (Continuity of μ). The response function μ is continuous and γ-lipschitz on $[0, \Lambda]$, i.e., for any $\lambda, \lambda' \in [0, \Lambda]$:

$$|\mu(\lambda) - \mu(\lambda')| \le \gamma|\lambda - \lambda'|.$$

Lemma C.3 (Fixed-point feasibility)**.** *Let $\lambda^\sharp \in [0, \Lambda]$ be the fixed point of the dual update*

$$\lambda^\sharp = \mathit{clip}(\lambda^\sharp + \eta\, \mu(\lambda^\sharp), 0, \Lambda).$$

Then $\mu(\lambda^\sharp) \le 0$, i.e., $\widetilde{C}(\pi(\lambda^\sharp)) \le 0$. Hence, $\pi^\sharp = \pi(\lambda^\sharp)$ is feasible.

Proof. We discuss the value of $\lambda^\sharp$: (i) if $0 < \lambda^\sharp < \Lambda$, then it implies $\eta\, \mu(\lambda^\sharp) = 0$, and thus $\mu(\lambda^\sharp) = 0$; (ii) if $\lambda^\sharp = 0$, then $\eta\, \mu(\lambda^\sharp) \le 0$, and thus $\mu(\lambda^\sharp) \le 0$; (iii) if $\lambda^\sharp = \Lambda$, then by assumption A2, we have $\mu(\Lambda) \le 0$. □

Theorem C.4 (Convergence)**.** *Let the step size* $0 < \eta \leq 1/\gamma$*. Then for any initialization* $\lambda_0 \in [0, \Lambda]$*, the sequence* $\lambda_{t+1} = \mathit{clip}(\lambda_t + \eta\, \mu(\lambda_t), 0, \Lambda)$ *converges to a fixed point* $\lambda^\sharp \in [0, \Lambda]$*. Moreover, the corresponding policy* $\pi^\sharp = \pi(\lambda^\sharp)$ *is feasible, i.e.,* $\widetilde{C}(\pi^\sharp) \leq 0$*.*

Proof. We turn the problem into convex optimization. First, define the potential

$$\Phi(\lambda) = \int_0^\lambda \mu(s)\, ds.$$

By Assumption A2, μ is nonincreasing, so Φ is concave; by Assumption A3, Φ is γ-smooth. Then we define a convex objective

$$f(\lambda) = -\Phi(\lambda).$$

We then have

$$\nabla f(\lambda) = -\mu(\lambda),$$

and the dual updates can be written as

$$\lambda_{t+1} = \mathrm{clip}(\lambda_t - \eta \nabla f(\lambda_t), 0, \Lambda).$$

By Fejer monotonicity (Theorem 10.23 and 10.24 in (Beck, 2017)), we claim that the distance between the iterates λ_t and the optimal solution (i.e., the fixed point) $\lambda^\sharp$ is nonincreasing:

$$\|\lambda_{t+1} - \lambda^\sharp\|^2 \leq \|\lambda_t - \lambda^\sharp\|^2,$$

and λ_t converges to the fixed point $\lambda^\sharp$. Finally, by Theorem C.3, the corresponding policy $\pi^\sharp = \pi(\lambda^\sharp)$ is feasible. □

C.4 Convergence Rate

For a clean linear convergence rate, we impose an additional assumption on μ.

A4 (Strong monotonicity). There exists $\xi > 0$ such that for all $\lambda \geq \lambda' \in [0, \Lambda]$, $\mu(\lambda) - \mu(\lambda') \leq -\xi(\lambda - \lambda')$.

Under Assumptions A1–A4, there exists a unique fixed point $\lambda^\sharp \in [0, \Lambda]$ such that $\mu(\lambda^\sharp) = 0$.

Theorem C.5 (Convergence rate)**.** *Let the step size* $0 < \eta \leq \min\{1/\gamma, 1/\xi\}$*. Then the dual iterates* λ_t *satisfy*

$$|\lambda_t - \lambda^\sharp| \leq (1 - \eta\xi)^t |\lambda_0 - \lambda^\sharp|. \tag{24}$$

Moreover, the constraint violation decays

$$|\mu(\lambda_t)| \leq \gamma|\lambda_t - \lambda^\sharp| \leq \gamma(1 - \eta\xi)^t |\lambda_0 - \lambda^\sharp|, \tag{25}$$

which implies that $\widetilde{C}(\pi_t) = \mu(\lambda_t)$ *converges to 0 at a geometric rate.*

Proof. We denote the dual update by a mapping operator $T(\lambda)$:

$$T(\lambda) = \mathrm{clip}(\lambda + \eta\, \mu(\lambda), 0, \Lambda).$$

Since $\lambda^\sharp \in [0, \Lambda]$ and $\mu(\lambda^\sharp) = 0$,

$$T(\lambda^\sharp) = \lambda^\sharp.$$

Since the clipping function is non-expansive, we have

$$\begin{aligned} |T(\lambda) - T(\lambda^\sharp)| &\leq |\lambda - \lambda^\sharp + \eta(\mu(\lambda) - \mu(\lambda^\sharp))| \\ &= |\lambda - \lambda^\sharp + \eta\, \mu(\lambda)|. \end{aligned} \tag{26}$$

Then by Assumption A4, if $\lambda \geq \lambda^\sharp$,

$$\mu(\lambda) \leq \mu(\lambda^\sharp) - \xi(\lambda - \lambda^\sharp) = -\xi(\lambda - \lambda^\sharp).$$

Therefore,

$$\begin{aligned} \lambda - \lambda^\sharp + \eta\, \mu(\lambda) &\leq (\lambda - \lambda^\sharp) - \eta\xi(\lambda - \lambda^\sharp) \\ &= (1 - \eta\xi)(\lambda - \lambda^\sharp), \end{aligned}$$

and

$$|\lambda - \lambda^\sharp + \eta\, \mu(\lambda)| \leq (1 - \eta\xi)(\lambda - \lambda^\sharp).$$

The same bound follows for $\lambda \leq \lambda^\sharp$. Plugging back in eq. (26), we have

$$|T(\lambda) - T(\lambda^\sharp)| \leq (1 - \eta\xi)|\lambda - \lambda^\sharp|.$$

Let $\lambda = \lambda_t$, and note that $T(\lambda_t) = \lambda_{t+1}$, we obtain the contraction of the dual update operator T:

$$|\lambda_{t+1} - \lambda^\sharp| \leq (1 - \eta\xi)|\lambda_t - \lambda^\sharp|.$$

Telescoping gives eq. (24). Finally, notice that $\mu(\lambda^\sharp) = 0$, we have

$$|\mu(\lambda_t)| = |\mu(\lambda_t) - \mu(\lambda^\sharp)| \leq \gamma|\lambda_t - \lambda^\sharp|,$$

and plugging in eq. (24) gives us eq. (25). □