DISTRIBUTION-PRESERVING RESPONSE LENGTH SEPARATION FOR LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Response length is a major source of uncertainty in serving large language models. Under stochastic decoding, the same request can produce responses that differ several-fold in length: for prompts from a single dataset, within-prompt variance in response length can exceed between-prompt variance. This uncertainty undermines efficient resource management. Predicting length from the prompt cannot address within-prompt variation, while steering generation towards a target length changes the model's output distribution.

We introduce **Length-Separated Sampling**, which divides generation into short and long regimes chosen at random before generation. We construct both regimes from the frozen target model so that their mixture is exactly the original distribution. The regime is therefore known before generation and informative about response length, while the responses users receive remain distributed exactly as under ordinary sampling. The long regime reliably generates longer responses than the short regime, with mean response length up to 2.8 times that of the short regime.

1 INTRODUCTION

Large language models (LLMs) can produce responses with vastly different lengths. Some requests may require only a few tokens, while others may generate thousands. Worse, the length is not settled once the prompt is known: under stochastic decoding the same prompt, sampled twice, can yield responses that differ several-fold in length. Response length is therefore uncertain in two distinct ways, and only one of them is a property of the prompt. As Figure 1 shows, the within-prompt variation is comparable in magnitude to the variation between prompts.

This variation directly affects the time, amount, and type of resources required to serve the request to completion, undermining efficient resource management. Indeed, decisions about admission, routing, scheduling, and batching must be made ahead of time Cheng et al. (2024; 2026).

A natural way to manage response-length variability is to predict response length from the prompt Jin et al. (2023); Zheng et al. (2023); Qiu et al. (2024). A system can then use this prediction for routing Fu et al. (2024); Shahout et al. (2024). This approach exploits variation *between* prompts, while leaving $p(x \mid z)$ unchanged for each prompt. However, it cannot anticipate which response length a particular prompt will produce. Another approach is to steer generation towards shorter or longer responses by modifying token probabilities Yang and Klein (2021); Gu et al. (2025); Samuel et al. (2025). This can exploit variation *within* a prompt, but changes the distribution from which the model samples. For a serving system, this is not an acceptable solution: the goal is to manage the outputs of the model being served, not to change their distribution.

We therefore ask: **Can length information of the response that is about to be generated be exposed before generation begins, without changing the distribution the model samples from?**

In this paper, we answer this question with **Length-Separated Sampling**. It splits generation into two regimes, one favouring shorter responses and one favouring longer responses. Before generation, a hidden variable s is drawn to determine which regime is used. The two regime distributions require no LLM fine-tuning: both are derived directly from the frozen target model using only a lightweight remaining-length predictor. Their mixture is exactly the original model distribution, so the responses users see, and any metric computed from them, are distributed exactly as under ordi-

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.

nary sampling. Because the regime is known before generation, it can support downstream decisions such as routing, scheduling, and load balancing.

In summary, our contributions are:

1. We introduce a method for separating LLM generation into short and long-response regimes while preserving the original model distribution exactly.
2. We show how to construct these two regimes during generation using a learned predictor of remaining response length.
3. We empirically verify the approach across eleven LLMs and three datasets, obtaining separation of up to 2.8× on mathematical reasoning and typically 1.1–1.5× on other tasks.

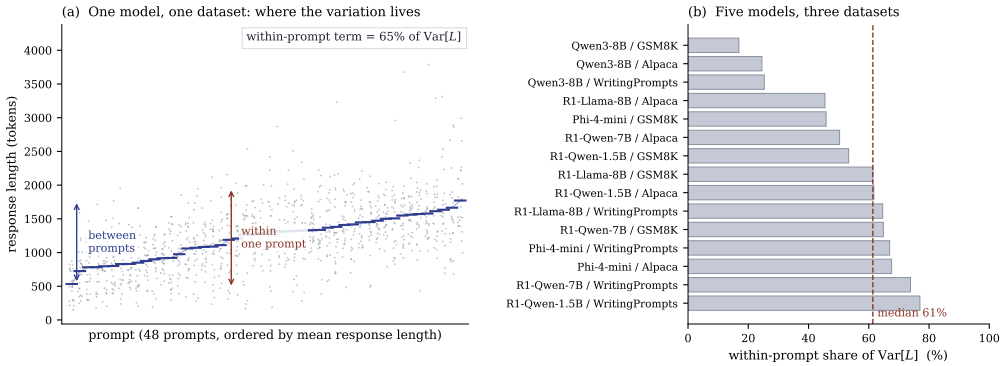

Figure 1: **Within-prompt variation in response length is comparable to between-prompt variation. (a)** Lengths of 24 responses to 48 WritingPrompts prompts (R1-Llama-8B, temperature 1), ordered by mean length. Within-prompt variation accounts for 65% of total variance, exceeding between-prompt variation. **(b)** Within-prompt share of total response-length variance across five models and three datasets. The median share is 61%, indicating within-prompt variation can exceed between-prompt variation. Qwen3-8B is an exception, with little variation across samples.

2 Related Work

Response-length prediction. A substantial body of work predicts response length from the prompt before generation. S^3 uses a lightweight predictor to estimate output length for memory allocation and scheduling Jin et al. (2023), while Response Length Perception predicts target model response length to group requests by expected length Zheng et al. (2023). SSJF uses predicted lengths to approximate shortest-job-first scheduling Qiu et al. (2024), and Fu et al. (2024) predicts relative length rankings rather than absolute lengths. Others leverage target model information, including hidden representations Shahout et al. (2024), graph-based representations Piotrowski et al. (2025), and entropy-based features Xie et al. (2026). Recent work also models length probabilistically to explicitly account for within-prompt variation Zheng et al. (2026). These methods inform the expected response length but do not alter the generation process itself.

Length-controlled generation. Another line of work directly guides generation towards a desired response length. FUDGE modifies token probabilities using a predictor over partial sequences Yang and Klein (2021), while Gu et al. (2025) use sampling-based methods to satisfy length constraints. CIE learns continuous controls for response length Samuel et al. (2025), while SimpleStrat samples from different length groups Wong et al. (2025). These approaches can reduce response-length variability or target particular lengths, but the resulting distribution differs from that of the original model. Related serving systems such as CascadeInfer use length information to create specialised groups of resources and adapt routing as generation progresses Yuan et al. (2025). In contrast, our method creates different length regimes before generation while preserving the original sampling distribution when the regimes are mixed.

Distribution-preserving sampling. Our approach is also related to methods that modify the sampling process while preserving a target distribution. Speculative decoding uses a proposal distri-

bution together with rejection and residual sampling to generate exactly from a target model distribution in order to accelerate inference Leviathan et al. (2023). Unbiased watermarking similarly constructs modified token distributions while preserving the underlying model distribution Hu et al. (2024). Provably secure steganography takes a closely related approach: Meteor Kaptchuk et al. (2021), DisCoP Ding et al. (2023), and de Witt et al. (2023) encode information in the sampling process while preserving the generated content distribution. Those methods hide the encoded information from an observer; our auxiliary variable is instead known to the serving system and deliberately made informative about response length. We use this degree of freedom to create distinct short and long-response regimes without changing the model's marginal output distribution.

3 Problem Formulation

3.1 Response Length under Stochastic Generation

We consider a fixed, pretrained language model, which we refer to as the *target model*, and let $p(x \mid z)$ denote its distribution over complete responses x for a prompt z. Let $L(x)$ denote the length of response x. Under stochastic decoding, different response samples from the same prompt can have substantially different lengths, giving a non-degenerate distribution $p(L \mid z)$.

Response length varies both across prompts and within a single prompt. Over a workload of prompts, we can decompose the total variance in response length as:

$$\mathrm{Var}[L] = \underbrace{\mathrm{Var}_z\big(\mathbb{E}[L \mid z]\big)}_{\text{between prompts}} + \underbrace{\mathbb{E}_z\big(\mathrm{Var}[L \mid z]\big)}_{\text{within prompts}}.$$

A prompt-level predictor can account for the first component by estimating the expected length for a given prompt. However, it cannot resolve the second component: it cannot determine which specific response length will actually be sampled from $p(L \mid z)$. Figure 1 illustrates this distinction.

3.2 Length Separation with Marginal Preservation

We seek an *inference-time sampling procedure* that provides information about response length before generation begins, without changing the response distribution. We introduce a binary generation variable $s \in \{\text{SHORT}, \text{LONG}\}$ drawn before generation:

$$s \sim \mathrm{Bernoulli}(\alpha).$$

The value of s selects one of two sampling procedures, which we refer to as the *short regime* and the *long regime*, inducing complete-response distributions $p_{\mathrm{short}}(x \mid z)$ and $p_{\mathrm{long}}(x \mid z)$ respectively. α is a tunable parameter determining the probability of assigning a request to the short regime.

Figure 2 illustrates this construction: ordinary sampling draws directly from $p(x \mid z)$, whereas we first draw s and then sample from the corresponding conditional distribution.

We want these $p_{\mathrm{short}}(x \mid z)$ and $p_{\mathrm{long}}(x \mid z)$ to satisfy two properties.

Goal: Length Separation. The regime variable s should be informative about response length, so the two regimes' expected lengths should differ as much as possible:

$$\mathbb{E}_{p_{\mathrm{short}}}[L \mid z] \;<\; \mathbb{E}_{p_{\mathrm{long}}}[L \mid z].$$

The degree of separation we can achieve depends on how predictable the remaining length given a partially generated response is.

Constraint: Marginal Preservation. Marginalised over s, the probability of generating any response x is the same as if we sampled directly from the target model. Explicitly, for any prompt–response pair (z, x):

$$\alpha p_{\mathrm{short}}(x \mid z) + \big(1-\alpha\big) p_{\mathrm{long}}(x \mid z) = p(x \mid z). \tag{1}$$

This is a hard requirement. Marginal preservation ensures responses received by a model user are distributed exactly as they would be under ordinary sampling. Thus, any downstream evaluation metric, such as accuracy or human preference, is guaranteed to remain identical to the target model.

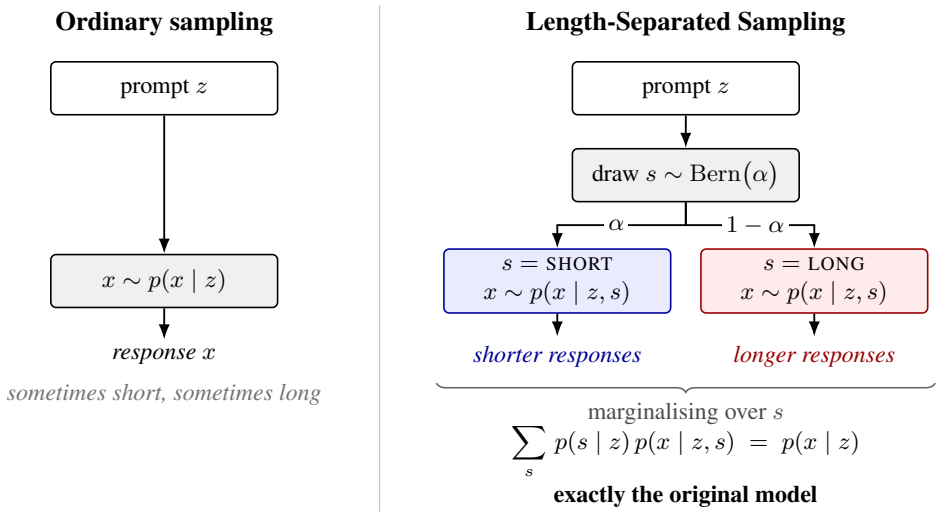

Figure 2: **Ordinary versus Length-Separated Sampling.** Left: Ordinary sampling draws a response directly from the target model $p(x \mid z)$. Right: Length-Separated Sampling first samples a binary generation variable s and then samples from a conditional distribution $p(x \mid z, s)$. The short and long conditionals favour shorter and longer responses, respectively, while their mixture remains exactly equal to the original model distribution.

4 Methodology

4.1 Overview

This section describes how to construct, given a target model p, a short regime p_{short} and a long regime p_{long} that satisfy the properties defined in Section 3.2. We first show that, in order to respect the marginal preservation constraint and exactly recover p, choosing p_{short} uniquely determines p_{long}. We therefore characterise the set $\mathcal{Q}_n$ made of short regimes p_{short} that yield a valid probability distribution for p_{long}.

These observations dictate our overall approach to construct p_{short}: we build an unconstrained candidate distribution q that encourages short responses, and then map q into the valid set $\mathcal{Q}_n$ to obtain an admissible p_{short}. We provide two methods to perform this mapping: a direct projection onto $\mathcal{Q}_n$, and a more general accept/reject sampling scheme.

Notation Let $\mathcal{V}$ be the target vocabulary and $\Delta(\mathcal{V})$ the set of probability distributions over it. We write $x_{:n} = (x_1, \ldots, x_n)$ for the prefix after n decoding steps and $y \in \mathcal{V}$ for a candidate next token. When conditioning on $x_{:n}$ and prompt z is clear, we abbreviate the next-token distributions as $p(y)$ and $q(y)$. Finally, let $\alpha_n := p(\text{SHORT} \mid x_{:n}; z)$ denote the posterior probability that prefix $x_{:n}$ was generated by the short regime.

4.2 Constructing the Short and Long Regimes

The complete-response marginal constraint (Eqn. 1) is equivalent to requiring that, at every decoding step and for every possible prefix $x_{:n}$, the mixture of the two next-token distributions equals the target distribution:

$$\alpha_n p_{\text{short}}(y \mid x_{:n}; z) + (1 - \alpha_n) p_{\text{long}}(y \mid x_{:n}; z) = p(y \mid x_{:n}; z), \qquad \forall y \in \mathcal{V}. \tag{2}$$

Consequently, choosing p_{short} uniquely determines p_{long}:

$$p_{\text{long}}(y) = \frac{p(y) - \alpha_n p_{\text{short}}(y)}{1 - \alpha_n}. \tag{3}$$

For p_{long} to remain a valid probability distribution, it must be non-negative. This imposes a strict envelope condition on p_{short}, which exactly characterises the set of all admissible short regimes at

step n:

$$\mathcal{Q}_n = \{r \in \Delta(\mathcal{V}) : \alpha_n r(y) \le p(y) \ \ \forall y\}. \tag{4}$$

With the valid set $\mathcal{Q}_n$ defined, we construct the regimes by first forming an unconstrained candidate distribution q that favours shorter continuations. We then project q onto $\mathcal{Q}_n$ by finding the distribution within $\mathcal{Q}_n$ that minimises the KL divergence from q; we call this the direct-projection approach (detailed in Appendix B.3). The resulting distribution serves as p_{short}. The long regime is then obtained from Eqn. 3.

More generally, we can enforce the same constraint at sampling time. We sample a proposal $y \sim q$ and accept it with probability

$$\text{acc}(y) = \min\left(1, \frac{p(y)}{\alpha_n q(y)}\right).$$

In the case of rejection, sample a replacement token from the residual distribution

$$p_{\text{res}}(y) = \text{norm}\left(\max\big(p(y) - \alpha_n q(y), 0\big)\right).$$

Hence, the short regime consists of accepted proposals and replacement draws, while the long regime samples directly from p_{res}.

Remark. Our experiments use the direct-projection method. We include the more general accept/reject construction to show that the same constraint can also be enforced by sampling, closely related to speculative decoding.

The construction requires the current posterior α_n at each decoding step. We initialise $\alpha_0 = \alpha$, where α is the probability of selecting the short regime before generation. After observing the next token x_{n+1}, we update the posterior using Bayes' rule:

$$\alpha_{n+1} = \alpha_n \frac{p_{\text{short}}(x_{n+1} \mid x_{:n}; z)}{p(x_{n+1} \mid x_{:n}; z)}. \tag{5}$$

The resulting construction preserves the target marginal for any choice of candidate distribution q.

Theorem 4.1 (Marginal preservation)**.** *At each decoding step, suppose p_{short} and p_{long} satisfy Eqn. 2, with α_n updated according to Eqn. 5. Then, for all z, x,*

$$\alpha p_{\text{short}}(x \mid z) + (1-\alpha) p_{\text{long}}(x \mid z) = p(x \mid z).$$

The proof is given in Appendix A.

4.3 Learning the Length-Biased Distribution

It remains to specify how the unconstrained distribution q is formed at each decoding step to favour shorter responses. To construct q, we must estimate how each potential next token impacts the final response length. We train a remaining-length predictor $V(y; x_{:n}, z)$ to predict a centred, log-transformed measure of the mean remaining response length after candidate token y is generated. We define q via a KL-regularised objective:

$$\min_q \quad \mathbb{E}_{y\sim q}[V(y)] + \frac{1}{\beta_n}\text{KL}(q||p).$$

The first term favours tokens predicted to lead to shorter continuations, while the KL term penalises deviation from the target distribution. The coefficient β_n controls the trade-off between these two terms. The solution has an exponential-tilt form:

$$q(y) \propto p(y)\exp[-\beta_n V(y)]. \tag{6}$$

Thus, tokens predicted to lead to longer continuations receive less probability, while tokens predicted to lead to shorter continuations receive more.

In practice, we apply this tilt to the top-k candidate tokens under p. Let $\mathcal{C}_n$ denote these candidates and

$$\bar{C}_n = \frac{\sum_{y\in\mathcal{C}_n} p(y)\exp[-\beta_n V(y)]}{\sum_{y\in\mathcal{C}_n} p(y)}.$$

We set

$$q(y) = \begin{cases} p(y)\exp[-\beta_n V(y)]/\bar{C}_n, & y \in \mathcal{C}_n, \\ p(y), & y \notin \mathcal{C}_n. \end{cases} \tag{7}$$

Thus, we obtain q by reweighting the candidate token probabilities in p according to the predicted remaining length. We then project q onto $\mathcal{Q}_n$ via the direct-projection method outlined in Section 4.2 to obtain the short regime distribution p_{short}. The practical implementation also clips the multiplicative tilt before projection; details are given in Appendix B.3.

Algorithm 1 summarises the resulting inference procedure.

Algorithm 1 Length-Separated Sampling

Require: Prompt z, target model p, length predictor V, initial probability α

Draw $s \sim \text{Bernoulli}(\alpha)$; set $\alpha_0 = \alpha$ and $x_{:0} = \emptyset$.
for $n = 0, 1, \ldots$ until EOS **do**
  Evaluate $p(\cdot \mid x_{:n}; z)$.
  Use V to exponentially tilt p, obtaining $q(\cdot \mid x_{:n}; z)$ (Eqn. 7).
  Project q onto $\mathcal{Q}_n$ to obtain p_{short}.
  Set $p_{\text{long}} = \frac{p - \alpha_n p_{\text{short}}}{1 - \alpha_n}$.
  if $s =$ SHORT **then**
    Sample $x_{n+1} \sim p_{\text{short}}$.
  else
    Sample $x_{n+1} \sim p_{\text{long}}$.
  end if
  Update $\alpha_{n+1} = \alpha_n \frac{p_{\text{short}}(x_{n+1})}{p(x_{n+1})}$.
  Append x_{n+1} to $x_{:n}$.
end for

4.3.1 Choosing the Tilt Strength

The marginal-preservation constraint imposes a hard upper bound on the cumulative deviation of the short regime from the target model. From Eqn. 5,

$$\log \alpha_{n+1} - \log \alpha_n = \log \frac{p_{\text{short}}(x_{n+1} \mid x_{:n}; z)}{p(x_{n+1} \mid x_{:n}; z)}.$$

Taking the expectation over $x_{n+1} \sim p_{\text{short}}(\cdot \mid x_{:n}; z)$ gives

$$\mathbb{E}\left[\log \alpha_{n+1} - \log \alpha_n\right] = \text{KL}\left(p_{\text{short}}(\cdot \mid x_{:n}; z) \,\|\, p(\cdot \mid x_{:n}; z)\right).$$

Since $\alpha_n \leq 1$, the total expected divergence accumulated over a response is bounded by

$$\sum_n \mathbb{E}\left[\text{KL}\left(p_{\text{short}} \| p\right)\right] \leq \log \frac{1}{\alpha_0}. \tag{8}$$

We refer to $\log(1/\alpha_0)$ as the ***KL-Budget***: it is the maximum cumulative KL divergence of p_{short} from the target model allowed by marginal preservation. Thus, smaller α_0 provides a larger budget, allowing the short regime to deviate further from p and produce shorter responses.

The tilt strengths β_n determine how strongly the candidate distribution is biased towards shorter continuations. We set $\beta_n = \beta_0 r_n$, with the overall tilt strength selected separately for each model–dataset pair using held-out validation data; the selection procedure is described in Appendix B. The weights r_n are fitted separately and are generally larger early in generation, where the predictor is more informative. The envelope projection guarantees marginal preservation for any choice of β_n.

4.3.2 Training the Remaining-Length Predictor

We train V from responses sampled from the fixed target model. Along each response, we select prefixes spread across the decoding steps. At each prefix, we draw six candidate next tokens from the target model's top-32 tokens, with probabilities proportional to their target probabilities. For each candidate y, we generate 16 independent continuations from the target model and use $\log(1 + \bar{R}_y)$ as the training target, where $\bar{R}_y$ is the mean remaining response length across those continuations.

Table 1: **Length separation across models and datasets.** For each prompt, we generate 8 responses from the short and long regimes, compute the mean response length for each, and take the long-to-short ratio. Entries in the α rows are the average of these ratios over 480 prompts; 1.0 means no separation. The "–" row gives the mean response length under ordinary sampling, in tokens. The $\pm$ values give 95% bootstrap confidence intervals, obtained by resampling the 480 prompts.

Dataset	α	AceReason Nemotron-7B	DeepMath 1.5B	R1-Llama 8B	Open-RS3 1.5B	Mistral-7B Instruct	R1-Qwen 14B	R1-Qwen 7B	Phi-4 mini
GSM8K	–	1284	1205	755	880	338	592	631	130
	0.25	2.64 $\pm$0.17	2.21 $\pm$0.14	2.00 $\pm$0.12	1.91 $\pm$0.13	1.86 $\pm$0.13	1.51 $\pm$0.08	1.55 $\pm$0.08	1.22 $\pm$0.02
	0.5	2.78 $\pm$0.17	2.29 $\pm$0.15	2.03 $\pm$0.12	1.86 $\pm$0.11	1.84 $\pm$0.11	1.62 $\pm$0.09	1.68 $\pm$0.09	1.26 $\pm$0.03
	0.75	2.81 $\pm$0.19	2.46 $\pm$0.15	2.16 $\pm$0.11	2.00 $\pm$0.12	2.08 $\pm$0.12	1.85 $\pm$0.10	1.81 $\pm$0.11	1.32 $\pm$0.03
Writing Prompts	–	1456	1257	1188	1150	991	1188	1202	358
	0.25	1.07 $\pm$0.03	1.26 $\pm$0.04	1.36 $\pm$0.03	1.13 $\pm$0.03	1.88 $\pm$0.09	1.04 $\pm$0.02	1.09 $\pm$0.03	1.62 $\pm$0.10
	0.5	1.04 $\pm$0.02	1.26 $\pm$0.04	1.24 $\pm$0.03	1.12 $\pm$0.04	1.84 $\pm$0.09	1.02 $\pm$0.02	1.09 $\pm$0.04	1.52 $\pm$0.09
	0.75	1.05 $\pm$0.02	1.29 $\pm$0.05	1.19 $\pm$0.02	1.11 $\pm$0.03	1.91 $\pm$0.09	1.04 $\pm$0.02	1.08 $\pm$0.03	1.50 $\pm$0.09
Alpaca	–	1158	1129	846	970	704	799	867	224
	0.25	1.17 $\pm$0.05	1.26 $\pm$0.05	1.06 $\pm$0.02	1.20 $\pm$0.05	2.90 $\pm$0.37	1.13 $\pm$0.02	1.14 $\pm$0.04	1.51 $\pm$0.36
	0.5	1.12 $\pm$0.04	1.22 $\pm$0.05	1.07 $\pm$0.03	1.18 $\pm$0.04	2.45 $\pm$0.24	1.08 $\pm$0.03	1.13 $\pm$0.05	1.36 $\pm$0.10
	0.75	1.09 $\pm$0.03	1.26 $\pm$0.04	1.09 $\pm$0.03	1.15 $\pm$0.04	2.51 $\pm$0.23	1.09 $\pm$0.04	1.10 $\pm$0.03	1.54 $\pm$0.18

At each prefix, we centre the targets by subtracting their mean across the candidates. Thus, V learns which candidate tokens lead to shorter or longer continuations relative to the other continuation tokens at the same prefix. This centring removes only a prefix-dependent offset and does not change the candidate-local tilt used to construct q.

5 Experiments

5.1 Setup

We evaluate Length-Separated Sampling on 11 open models and three datasets: WritingPrompts (story writing), GSM8K (maths word problems), and Alpaca (general instructions). Eight models are included in the main paper and three additional models are reported in Appendix C.2. The models span RL-trained reasoning models (AceReason, DeepMath, and Open-RS2/3), distillation-trained R1 models, Qwen3-8B run with thinking enabled, and non-thinking models (Mistral-7B-Instruct and Phi-4-Mini). Mean response length ranges from 130 tokens (Phi-4-Mini on GSM8K) to 1,456 tokens (AceReason on WritingPrompts) in the main table.

For each of 480 held-out prompts, we draw 8 responses (at temperature 1) from each of the three regimes: ordinary sampling, *short*, and *long*. We repeat the experiment for $\alpha \in \{0.25, 0.5, 0.75\}$, for a total of 99 (model, dataset, α) combinations. For each model–dataset pair, a length predictor is trained and β is selected using a separate validation set; full details are given in Appendix B.

5.2 Results

Table 1 reports the long-to-short ratio in mean response length: for each prompt, the mean length of its long-regime responses divided by that of its short-regime responses, averaged over prompts. A ratio of 1.0 means no separation. Separation is strongest on GSM8K, where the average long-to-short ratio reaches 2.81: for a typical prompt, the long regime produces nearly three times the response length of the short regime. On WritingPrompts and Alpaca the effect is smaller but present for most models, and it is consistent across the three values of α. The method also separates responses from non-thinking models (Mistral-7B-Instruct and Phi-4-Mini), so it does not depend on an explicit reasoning phase. Appendix C.1 compares these ratios against the maximum separation achievable for each model–dataset pair; Appendix C.6 examines how much of the separation is due to the learned predictor.

Figure 3 provides details about a representative model–dataset combination, i.e., R1-Llama-8B on GSM8K at $\alpha = 0.5$. In this combination, the mean response length of the long regime exceeds that of the short regime for 417 of the 480 prompts, with a mean long-to-short ratio of 2.03. The pooled

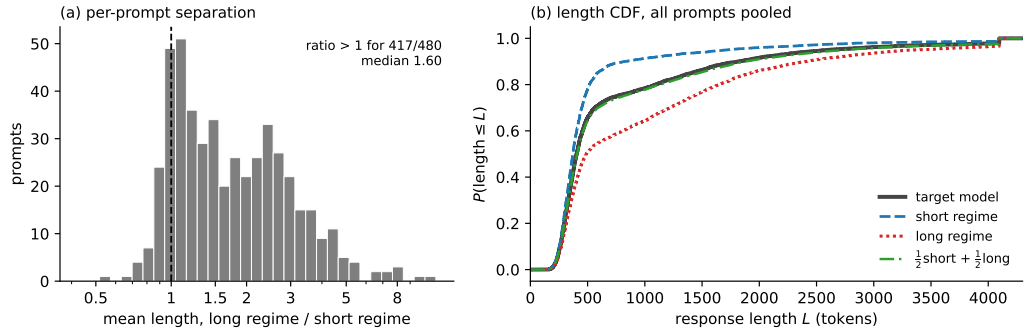

Figure 3: **Consistent length separation across prompts.** R1-Llama-8B on GSM8K (α=0.5, T=1, 480 prompts, 8 samples/prompt). **(a)** Histogram of the ratio of mean long to mean short response length per prompt (mean 2.03; > 1 for 417/480 prompts). **(b)** Empirical length CDFs pooled across all prompts. The short (blue) and long (red) regimes successfully isolate shorter and longer responses, while their mixture (green dashed) closely matches ordinary sampling (black).

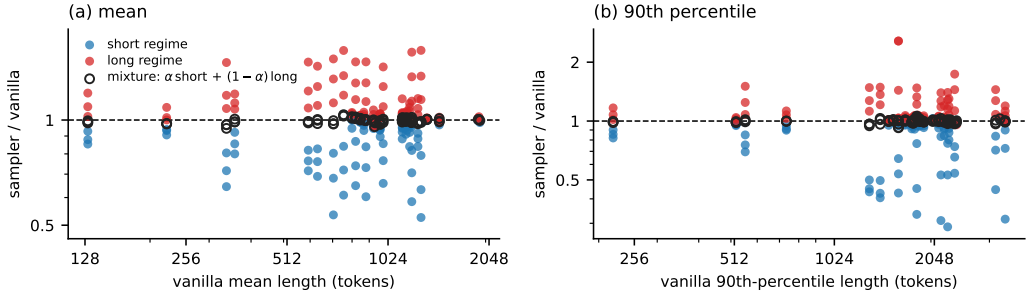

Figure 4: **Marginal preservation.** The short and long regimes shift response lengths in opposite directions relative to ordinary (vanilla) sampling, while their mixture does not. Each point is one (model, dataset, α) combination, giving 99 combinations in total. The vertical axis shows response length relative to ordinary sampling: values below 1 are shorter and values above 1 are longer. Blue shows the short regime, red the long regime, and black circles their α-mixture. **(a)** Mean response length. **(b)** 90th-percentile response length. When the regimes separate, the short regime is below vanilla and the long regime above it, reaching $0.53\times$ and $1.58\times$ in the mean and $0.29\times$ and $2.57\times$ in the tail. The mixture stays within 5% of vanilla in 98 of 99 settings.

response-length CDFs in the right panel show that the two regimes produce clearly different length distributions, while their mixture closely matches ordinary sampling, as required by the construction.

Marginal preservation: Theorem 4.1 guarantees that mixing p_{short} and p_{long} in proportion $\alpha : 1-\alpha$ exactly reproduces the target distribution by construction. Figure 4 provides an empirical check based on response lengths. Across all 99 (model, dataset, α) combinations, we compare the mean and 90th-percentile response lengths of ordinary sampling against the short and long regimes and their α-mixture. For both statistics, the short regime produces shorter responses and the long regime longer ones. Crucially, the α-mixture matches the ordinary sampling mean in all 99 settings, fulfilling the theoretical guarantee. Its 90th-percentile length is within 5% in 98 of 99 settings. A distribution-level test is in Appendix C.3.

The Role of α: As discussed in Section 4.3.1, α determines how much the short regime may deviate from the target model: smaller α provides a larger KL budget, allowing p_{short} to be pushed further towards shorter responses. Figure 5 confirms this behaviour. When α is small, the short regime produces substantially shorter responses while the long regime remains closer to ordinary sampling. As α increases, the short-regime budget shrinks towards zero, so it approaches the target distribution, while the long regime absorbs the remaining deviation and produces longer responses. Thus, α provides a low-cost deployment-time lever for controlling which response-length regime is isolated, without retraining the predictor.

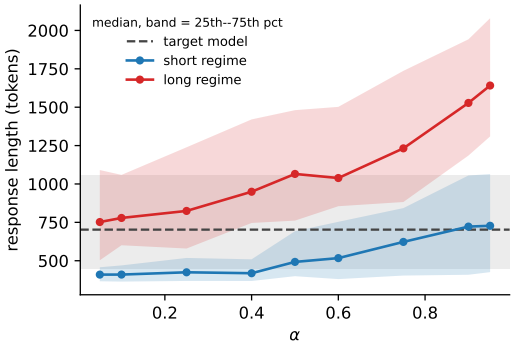

Figure 5: **Median response length against α.** Shading shows the 25th–75th percentile across prompts. Lowering α shortens the short regime's responses, while raising α lengthens the long regime's responses. The dashed line shows the median (independent of α) for ordinary sampling. As $\alpha \to 1$, the short regime approaches ordinary; as $\alpha \to 0$, the long regime approaches ordinary. Nine values of α; 24 prompts from GSM8K $\times$ 16 response samples from R1-Llama-8B.

6 Discussion

The effectiveness of Length-Separated Sampling depends on i) inherent characteristics of the model and dataset, and ii) controllable decoding parameters. High-variance settings, such as GSM8K, yield substantial length separation. Conversely, models with little within-prompt variation, like Qwen3-8B, yield little, but there is correspondingly less to gain: length is largely determined by the prompt. Such variance, and thus the method's effectiveness, is further modulated by configurations like the adopted sampling temperature. Appendix C.4 examines the effect of this factor.

Limitations Length-Separated Sampling has several limitations. It adds a small computational overhead to inference; roughly a 1% increase in latency for a 32B model at batch size 256 (Appendix B.5). Its effectiveness depends on within-prompt response-length variation; separation is weak when this is small, as for Qwen3-8B. Finally, the method separates the response-length distributions rather than imposing a strict length threshold: short-regime responses can occasionally be longer than long-regime responses. For example, for R1-Qwen-7B on GSM8K at $\alpha = 0.25$, 1.4% of short-regime responses exceed the 90th-percentile length of the long regime.

Significance and Future Work The main significance of the method is that it exposes information about the response that is about to be generated before generation begins, without changing the marginal distribution of the model's outputs. Unlike prompt-level length prediction, which predicts expected response length from the prompt, our method provides information about which of the possible response lengths will be realised. This could support serving decisions such as routing requests to different hardware or grouping requests for batching, although we do not evaluate these downstream benefits here. More broadly, the same construction could expose other properties of a future response, such as whether it is likely to invoke a particular tool or contain a particular type of content. Appendix D shows that the construction also generalises to more than two regimes; further generalisations, such as a continuous regime space, are left to future work.

7 Conclusions

This work introduced Length-Separated Sampling to expose information about response length before generation, while preserving the target model's marginal response distribution exactly. The method separates shorter and longer responses across a range of models and datasets, while leaving the distribution seen by users unchanged. This provides a way to anticipate the likely length of a response before it is generated, with potential applications in model serving and other settings where decisions must be made in advance.

AI Assistance Disclosure

The core ideas, methodology, experimental design, mathematical claims, and interpretation of results in this work were developed by the authors. Large Language Models (LLMs) and other AI tools were used during the research process to assist with implementing and running experiments, writing up the proofs from the authors' derivations, drafting and editing parts of the manuscript, and searching for and summarising related literature. AI tools were also used as programming assis-

tants for code development and debugging. The authors take full responsibility for the originality, accuracy, and integrity of the final submission.

Reproducibility Statement

To ensure the full reproducibility of our results, we have detailed the experimental setup, model configurations, and hyperparameter choices in the Appendices B, C. This includes exact specifications for the remaining-length predictor architecture, training data generation (fork data), and inference-time projection parameters. All datasets used in our evaluation (GSM8K, Alpaca, and WritingPrompts) and models are publicly available.

References

Jiaming Cheng, Duong The Do, and Duong Tung Nguyen. Robust KV cache management for LLM serving under output token length uncertainty, 2026. arXiv:2607.16892.

Ke Cheng, Wen Hu, Zhi Wang, Peng Du, Jianguo Li, and Sheng Zhang. Enabling efficient batch serving for lmaas via generation length prediction. In *2024 IEEE International Conference on Web Services (ICWS)*, 2024. doi: 10.1109/ICWS62655.2024.00104.

Christian Schroeder de Witt, Samuel Sokota, J. Zico Kolter, Jakob Nicolaus Foerster, and Martin Strohmeier. Perfectly secure steganography using minimum entropy coupling. In *International Conference on Learning Representations*, 2023.

Jinyang Ding, Kejiang Chen, Yaofei Wang, Na Zhao, Weiming Zhang, and Nenghai Yu. Discop: Provably secure steganography in practice based on "distribution copies". In *2023 IEEE Symposium on Security and Privacy*, pages 2238–2255, 2023. doi: 10.1109/SP46215.2023.10179287.

Yichao Fu, Siqi Zhu, Runlong Su, Aurick Qiao, Ion Stoica, and Hao Zhang. Efficient llm scheduling by learning to rank. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yuxuan Gu, Wenjie Wang, Xiaocheng Feng, Weihong Zhong, Kun Zhu, Lei Huang, Ting Liu, Bing Qin, and Tat-Seng Chua. Length controlled generation for black-box LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16878–16895, 2025.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *International Conference on Learning Representations*, 2024.

Fangyin Jin et al. S3: Increasing gpu utilization during generative inference for higher throughput. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Gabriel Kaptchuk, Tushar M. Jois, Matthew Green, and Aviel D. Rubin. Meteor: Cryptographically secure steganography for realistic distributions. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1529–1548, 2021. doi: 10.1145/3460120.3484550.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 19274–19286. PMLR, 2023.

Grzegorz Piotrowski, Mateusz Bystroński, Mikołaj Hołysz, Jakub Binkowski, Grzegorz Chodak, and Tomasz Jan Kajdanowicz. When will the tokens end? graph-based forecasting for LLMs output length. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 843–848, 2025.

Haoran Qiu, Weichao Mao, Archit Patke, Shengkun Cui, Saurabh Jha, Chen Wang, Hubertus Franke, Zbigniew T. Kalbarczyk, Tamer Basar, and Ravishankar K. Iyer. Efficient interactive llm serving with proxy model-based sequence length prediction. *arXiv preprint arXiv:2404.08509*, 2024.

Vinay Samuel, Harshita Diddee, Yiming Zhang, and Daphne Ippolito. Cie: Controlling language model text generations using continuous signals. *arXiv*, 2025. doi: 10.48550/arxiv.2505.13448.

Rana Shahout, Eran Malach, Chunwei Liu, Weifan Jiang, Minlan Yu, and Michael Mitzenmacher. Don't stop me now: Embedding based scheduling for llms. *arXiv preprint arXiv:2410.01035*, 2024.

Justin Wong, Yury Orlovskiy, Alexander Shypula, Michael Luo, Sanjit A. Seshia, and Joseph E. Gonzalez. Simplestrat: Diversifying language model generation with stratification. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Huanyi Xie, Yubin Chen, Liangyu Wang, Lijie Hu, and Di Wang. Predicting LLM output length via entropy-guided representations. *arXiv preprint arXiv:2602.11812*, 2026.

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, 2021.

Yitao Yuan, Chenqi Zhao, Bohan Zhao, Zane Cao, Yongchao He, and Wenfei Wu. Cascadeinfer: Length-aware scheduling of LLM serving with low latency and load balancing. *arXiv preprint arXiv:2512.19179*, 2025.

Haoyu Zheng, Yongqiang Zhang, Fangcheng Fu, Xiaokai Zhou, Hao Luo, Hongchao Zhu, Yuanyuan Zhu, Hao Wang, Xiao Yan, and Jiawei Jiang. Scheduling LLM inference with uncertainty-aware output length predictions. *arXiv preprint arXiv:2604.00499*, 2026.

Zangwei Zheng, Xiaozhe Ren, Fuzhao Xue, Yang Luo, Xin Jiang, and Yang You. Response length perception and sequence scheduling: An llm-empowered llm inference pipeline. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A Proofs

A.1 Marginal preservation

The proof of Thm. 4.1 has two parts: a per-token identity, and an induction that lifts it to complete responses.

Lemma A.1 (Per-token identity)**.** *For any prefix $x_{:n}$ with $\alpha_n \in (0, 1)$, the short and long next-token distributions constructed above satisfy*

$$\alpha_n \, p_{\mathrm{short}}(y) + (1 - \alpha_n) \, p_{\mathrm{long}}(y) = p(y) \qquad \forall y \in \mathcal{V}. \tag{9}$$

Our implementation constructs p_{short} by projecting the candidate q onto $\mathcal{Q}_n$, so the resulting q satisfies the envelope condition $\alpha_n q(y) \le p(y)$ for every token. The more general accept/reject construction handles candidates that do not satisfy this condition. When the envelope condition does hold, every proposal is accepted and the rejection probability is zero, so the accept/reject construction reduces exactly to our direct projection, with $p_{\mathrm{short}} = q$. Thus, it is sufficient to prove the per-token identity for the general accept/reject construction; the result then also applies to our implementation.

Proof. Write α for α_n and suppress the conditioning on $x_{:n}$ and z. For the general accept/reject construction, a candidate $y \sim q$ is accepted with probability $\mathrm{acc}(y) = \min\big(1, p(y)/\alpha q(y)\big)$ and otherwise draws from $p_{\mathrm{res}}(y) = \mathrm{norm}\big(\max(p(y) - \alpha q(y), 0)\big)$, so with ρ the rejection probability

$$p_{\mathrm{short}}(y) = q(y) \, \mathrm{acc}(y) + \rho \, p_{\mathrm{res}}(y), \qquad p_{\mathrm{long}}(y) = p_{\mathrm{res}}(y). \tag{10}$$

Let $\mu(y) \coloneqq \min\big(\alpha q(y), \, p(y)\big)$ and $M \coloneqq \sum_{y \in \mathcal{V}} \mu(y)$. Directly from the definitions,

$$\alpha \, q(y) \, \mathrm{acc}(y) = \mu(y) \qquad \text{and} \qquad \max\big(p(y) - \alpha q(y), \, 0\big) = p(y) - \mu(y).$$

The second equality gives $p_{\text{res}}(y) = \frac{p(y)-\mu(y)}{1-M}$, well defined because $\mu(y) \le \alpha q(y)$ forces $M \le \alpha < 1$. The first equality gives $\sum_y q(y)\,\text{acc}(y) = \frac{M}{\alpha}$, so $\rho = 1 - \frac{M}{\alpha}$ and $\alpha\rho = \alpha - M$. Hence

$$\begin{aligned}\alpha\, p_{\text{short}}(y) + (1-\alpha)\, p_{\text{long}}(y) &= \alpha\, q(y)\,\text{acc}(y) + (\alpha\rho + 1 - \alpha)\, p_{\text{res}}(y) \\ &= \mu(y) + (1-M)\, p_{\text{res}}(y) \\ &= \mu(y) + \big(p(y) - \mu(y)\big) = p(y).\end{aligned}$$

□

Proof of Theorem 4.1. Each of the two regimes and the target model generates responses token by token, so their response probabilities factorise into per-token conditionals:

$$p_{\text{short}}(x_{:n}) := \prod_{k=1}^{n} p_{\text{short}}(x_k \mid x_{:k-1}; z), \qquad p_{\text{long}}(x_{:n}) := \prod_{k=1}^{n} p_{\text{long}}(x_k \mid x_{:k-1}; z),$$

and likewise $p(x_{:n} \mid z) = \prod_k p(x_k \mid x_{:k-1}; z)$. We show by induction on n that for every prefix $x_{:n}$,

$$\alpha_0\, p_{\text{short}}(x_{:n}) + (1-\alpha_0)\, p_{\text{long}}(x_{:n}) = p(x_{:n} \mid z). \tag{11}$$

which for a complete response x is exactly Eqn. 1.

For $n = 0$ both sides equal 1. For the inductive step, first observe that the update rule Eqn. 5 telescopes: $\alpha_n = \alpha_0\, p_{\text{short}}(x_{:n})/p(x_{:n} \mid z)$, i.e. α_n is precisely the Bayes posterior of the event $s = \text{SHORT}$ given the observed prefix. Consequently

$$\alpha_0 p_{\text{short}}(x_{:n}) = \alpha_n\, p(x_{:n} \mid z),$$

and, by the inductive hypothesis Eqn. 11,

$$(1-\alpha_0) p_{\text{long}}(x_{:n}) = (1-\alpha_n)\, p(x_{:n} \mid z).$$

Extending the prefix by one token,

$$\begin{aligned}&\alpha_0 p_{\text{short}}(x_{:n+1}) + (1-\alpha_0) p_{\text{long}}(x_{:n+1}) \\ &\quad= \alpha_0 p_{\text{short}}(x_{:n})\, p_{\text{short}}(x_{n+1} \mid x_{:n}) + (1-\alpha_0) p_{\text{long}}(x_{:n})\, p_{\text{long}}(x_{n+1} \mid x_{:n}) \\ &\quad= p(x_{:n} \mid z)\, \Big[\alpha_n\, p_{\text{short}}(x_{n+1} \mid x_{:n}) + (1-\alpha_n)\, p_{\text{long}}(x_{n+1} \mid x_{:n})\Big] \\ &\quad= p(x_{:n} \mid z)\; p(x_{n+1} \mid x_{:n}; z) && \text{by Lemma A.1} \\ &\quad= p(x_{:n+1} \mid z). && \square\end{aligned}$$

B Further Implementation Details

B.1 Tilt Strength and Reliability Weights

At decoding step n, let p_n denote the target next-token distribution, q_n the length-biased candidate before projection, and $p_{\text{short},n}$ the resulting short-regime distribution. The marginal constraint places a hard limit on how far the short-regime distribution can deviate from the target model: across an entire response, the expected cumulative KL divergence

$$\mathbb{E}\left[\sum_n \text{KL}\left(p_{\text{short},n} \| p_n\right)\right]$$

cannot exceed

$$\log \frac{1}{\alpha_0}.$$

We refer to $\log(1/\alpha_0)$ as the ***KL-Budget***.

We use this bound to define a reference scale for the tilt.

We let the tilt strength vary with decoding position, $\beta_n = \beta_0 r_n$, where $r_n \in [0, 1]$ down-weights positions at which the predictor is unreliable. To set r_n, we group decoding positions into a few ranges and, for each range, measure on held-out forks how well V predicts its target. Where V

carries no signal, $r_n = 0$ and the candidate reverts to p, so no budget is spent there. For weak tilts the cumulative KL divergence is then approximately quadratic in β_0 (App. B.2).

Writing $S(\beta_0)$ for the expected cumulative divergence gives $S(\beta_0) \approx \beta_0^2 S(1)$, leading to the reference scale

$$\beta_0 = \sqrt{\frac{\log(1/\alpha_0)}{S(1)}}.$$

The reported runs then select the tilt strength for each model–dataset pair from $\{0, \frac{1}{4}, \frac{1}{2}, 1, 2, 4\} \times \beta_0$ using a separate validation set. The marginal-preservation bound holds for any choice of β because it depends only on the final short-regime distribution.

Neither V nor the weights r_n depend on α_0. Thus, changing the initial short-regime probability changes the tilt scale without requiring the predictor to be retrained.

B.2 Quadratic approximation of the KL divergence

We use the following second-order approximation to define a reference scale for the tilt strength. For this derivation, consider the idealised full-vocabulary exponential tilt

$$q_\beta(y) = \frac{p(y)e^{-\beta V(y)}}{Z(\beta)}, \qquad Z(\beta) = \sum_y p(y)e^{-\beta V(y)}. \tag{12}$$

The KL divergence can be written as

$$\mathrm{KL}(q_\beta \| p) = -\beta\, \mathbb{E}_{q_\beta}[V] - \log Z(\beta). \tag{13}$$

We first expand the normalising constant around $\beta = 0$. Using $e^{-\beta V} \approx 1 - \beta V + \frac{1}{2}\beta^2 V^2$ and $\log(1 + x) \approx x - \frac{1}{2}x^2$ gives

$$\log Z(\beta) \approx -\beta\, \mathbb{E}_p[V] + \frac{1}{2}\beta^2 \mathrm{Var}_p(V). \tag{14}$$

Moreover,

$$\mathbb{E}_{q_\beta}[V] = -\frac{d}{d\beta} \log Z(\beta) \approx \mathbb{E}_p[V] - \beta\, \mathrm{Var}_p(V). \tag{15}$$

Substituting these expressions into Eqn. 13, the linear terms cancel, yielding

$$\mathrm{KL}(q_\beta \| p) \approx \frac{1}{2}\beta^2 \mathrm{Var}_p(V). \tag{16}$$

This quadratic scaling motivates the parameterisation used in our experiments. For $\beta_n = \beta_0 r_n$, we approximate the expected cumulative KL divergence as

$$S(\beta_0) \approx \beta_0^2 S(1),$$

where $S(\beta_0)$ denotes the cumulative KL divergence between the length-biased candidate and the target distribution. We use this relation only to define the reference scale β_0; the final β is selected on the validation set.

B.3 Inference-Time Implementation Details

At inference time, we apply the length bias to the top-64 tokens under the target distribution; the remaining vocabulary mass is grouped into a single tail bucket and distributed proportionally to p. Before projection, we normalise the tilt over the candidate tokens so that its p-weighted average is one, then clip it to $[1/16, 16]$.

Projection onto $\mathcal{Q}_n$. The resulting tilted distribution $\tilde{q}$ need not satisfy the envelope condition, so we project it onto $\mathcal{Q}_n$ by solving

$$q^\star = \arg\min_{q \in \mathcal{Q}_n} \mathrm{KL}(q \| \tilde{q}). \tag{17}$$

The solution has the simple form

$$q^\star(y) = \min\left(\gamma\,\tilde{q}(y), \frac{p(y)}{\alpha_n}\right), \tag{18}$$

where $\gamma \geq 1$ is chosen so that $\sum_y q^\star(y) = 1$. Thus, tokens below the envelope are rescaled by the same factor, while tokens that would violate the envelope are clipped to its boundary. We solve for γ in closed form.

Termination tokens are excluded from predictor training and assigned the largest probability permitted by the envelope before projection. The projection is applied to the top-64 tokens and the pooled tail; because the tail is distributed proportionally to p, this enforces the envelope over the full vocabulary.

B.4 Fork Data and Predictor Training

Fork data. For each model and dataset, we sample one response per training prompt from the frozen model at temperature 1, using 200–600 prompts per cell depending on the model and dataset. All training prompts are disjoint from the reported evaluation prompts.

We select fork sites at the first token and at up to one site from each of the log-spaced ranges $[1, 5)$, $[5, 17)$, $[17, 49)$, $[49, 129)$, $[129, 385)$, $[385, 1025)$, and $[1025, \infty)$ reached by the response, then add random sites from the remaining positions if needed, up to 8–10 sites per response. At each site, we consider six candidate next tokens: the termination token and five tokens sampled without replacement from the model's 32 most probable next tokens, with probability proportional to their model probabilities.

For each candidate, we generate 16 independent continuations from the frozen model at temperature 1, using the same length cap as in evaluation. The training target is $\log(1 + \bar{L})$, where $\bar{L}$ is the mean remaining length over the 16 continuations. We centre this target across the non-termination candidates at each fork and standardise using a single standard deviation computed over the training data.

This produces roughly 600–5,300 fork sites per model–dataset pair. The resulting candidate sets contain roughly 3,400–31,500 candidate rows, each with 16 continuations. For each model–dataset pair, we select the tilt strength using a separate set of 120 validation prompts, disjoint from the 480 prompts used for the reported results.

Predictor. V is a small feed-forward head on the frozen target model. It uses the target model's hidden state at a fixed intermediate layer (about 0.84–0.86 of network depth), together with the input embedding of the candidate token. The hidden state and candidate embedding are each projected to 512 dimensions; these projections, their elementwise product, and the projected scalar feature are concatenated and passed through a $2048 \to 512 \to 512 \to 1$ GELU network with dropout. The head has 5–6M parameters. We train with AdamW (learning rate 10^{-3}, batch size 1024) for 12 epochs with a Huber loss, selecting the epoch with the highest held-out within-fork correlation. Training takes roughly 15–50 seconds on a CPU across our smallest and largest fork sets.

B.5 Computational Cost

Our method adds computation at each decoding step to evaluate the length predictor for 64 candidate next tokens and project the resulting distribution onto the valid set. As a vLLM logits processor at batch size 128, this adds 2.1–2.6% to decode latency for Qwen2.5-32B-Instruct at tensor parallelism 2 and 3.7–3.8% for R1-Distill-Qwen-14B.

The relative overhead decreases with batch size: for Qwen2.5-32B-Instruct it is 3.4% at batch 16 and 1.0% at batch 256, because the added time is roughly fixed, about 0.5 ms per step up to batch 128 and 0.25 ms at batch 256, while the model's own step time grows from 15.5 to 26.6 ms. Most of the added cost comes from the full-vocabulary softmax and top-k selection used to form the candidate set. Because this overhead depends primarily on vocabulary size, rather than model depth or width, its relative cost should decrease for larger dense models. For a dense model an order of magnitude larger than those we measured, the overhead should be well below 1%.

C Further Experiment details

C.1 Performance Relative to Oracle

Table 2 compares our separation with an oracle that has access to the vanilla response lengths for each prompt. For a given α, for each prompt, the oracle assigns the shortest α fraction of responses to the short regime and the longest $1 - \alpha$ fraction to the long regime. The average long-regime response length is then divided by the average short-regime response length. This ratio is averaged over the prompts. This gives the largest long-to-short mean-length ratio achievable by any split that preserves the vanilla response distribution for each prompt. Equivalently, for each prompt the oracle value is the ratio of the mean response length in the upper $1 - \alpha$ quantile to that in the lower α quantile; these ratios are then averaged over prompts.

We report the fraction of this available separation closed by our method as

$$100 \times \frac{\text{achieved} - 1}{\text{oracle} - 1},$$

where 'achieved' is the separation reported in Table 1. This is zero when there is no separation and 100% when the oracle is matched; it can be slightly negative when the measured separation falls just below 1, which is within noise of no separation at all.

On GSM8K, our method typically captures around 60-90% of the oracle split, reaching 91% in the best case. On WritingPrompts and Alpaca, it captures a smaller fraction, typically 10–40% of the oracle.

C.2 Other Models

Table 1 presents the average length separation achieved for 8 models. Table 3 reports 3 additional models that did not fit in the main paper: R1-Qwen-1.5B, Qwen3-8B, and Open-RS2-1.5B. The method is generally effective for R1-Qwen-1.5B and Open-RS2-1.5B, with ratios ranging from 1.06–1.89. Qwen3-8B is a notable exception. Across all three datasets, the method provides no clear gain over ordinary sampling, with split ratios remaining close to 1.

C.3 Empirical Check of Marginal Preservation

Theorem 4.1 guarantees marginal preservation exactly, so this check only confirms that the implementation follows the construction. We test the response-length distribution on R1-Llama-8B $\times$ GSM8K at $\alpha = 0.5$, with 240 prompts and 64 responses per prompt from each of ordinary sampling, the short regime and the long regime.

For each prompt we split the ordinary responses into two halves and use one half as a reference. Each response in the other half, and each response in the α-weighted mixture of the short and long regimes, is assigned its percentile among the reference lengths. If a set of responses has the same length distribution as ordinary sampling, these percentiles are uniform. We pool them across prompts, measure the departure from uniformity with the Anderson–Darling statistic, and repeat the random split 200 times. A set is flagged when its statistic exceeds the 95th percentile of the values obtained for the held-out ordinary half, so ordinary sampling itself is flagged on 5% of splits by construction.

compared against the reference	statistic	% of splits flagged
held-out ordinary half	1.89	5.0
α-weighted mixture	1.59	1.0
short regime alone	317.25	100.0

The mixture is flagged on 1% of splits, no more often than ordinary sampling, while the short regime alone is flagged on every split. The test therefore detects the shift introduced by an individual regime and finds none in their mixture.

Table 2: **Fraction of the available length separation closed.** For each model–dataset pair and value of α, the oracle gives the maximum long-to-short ratio obtainable by perfectly splitting the ordinary responses by length. 'Closed' is the fraction of the available separation achieved by our method, $(\text{achieved} - 1)/(\text{oracle} - 1)$: 0% means no length-separation and 100% means matching the oracle. Both quantities are averaged over prompts as in Table 1.

		$\alpha = 0.25$		$\alpha = 0.5$		$\alpha = 0.75$	
model	dataset	oracle	closed	oracle	closed	oracle	closed
AceReason-Nemotron-7B	GSM8K	3.17	76%	3.04	87%	2.99	91%
	WritingPrompts	2.14	6%	1.90	4%	1.87	5%
	Alpaca	2.08	15%	1.95	12%	1.95	10%
DeepMath-1.5B	GSM8K	2.92	63%	2.89	68%	3.01	73%
	WritingPrompts	2.76	15%	2.46	18%	2.46	20%
	Alpaca	2.53	17%	2.40	15%	2.43	18%
R1-Llama-8B	GSM8K	2.30	77%	2.33	78%	2.37	85%
	WritingPrompts	1.87	41%	1.68	34%	1.64	30%
	Alpaca	1.76	7%	1.69	10%	1.71	12%
Mistral-7B-Instruct†	GSM8K	3.58	34%	3.21	38%	3.47	44%
	WritingPrompts	4.32	26%	3.66	32%	3.59	35%
	Alpaca	8.67	25%	6.63	26%	5.50	33%
Open-RS3 (1.5B)	GSM8K	2.36	67%	2.33	64%	2.56	64%
	WritingPrompts	2.55	9%	2.24	10%	2.15	10%
	Alpaca	2.14	18%	2.04	18%	2.10	14%
R1-Qwen-1.5B	GSM8K	2.32	63%	2.32	63%	2.44	62%
	WritingPrompts	2.36	10%	2.06	9%	2.02	9%
	Alpaca	2.05	12%	1.93	14%	2.02	15%
R1-Qwen-14B	GSM8K	1.83	62%	1.89	70%	2.02	84%
	WritingPrompts	1.99	4%	1.76	2%	1.71	5%
	Alpaca	1.73	18%	1.67	12%	1.68	14%
R1-Qwen-7B	GSM8K	1.90	61%	1.94	72%	2.14	71%
	WritingPrompts	2.18	8%	1.94	9%	1.89	9%
	Alpaca	1.94	15%	1.83	15%	1.85	12%
Phi-4-Mini†	GSM8K	1.58	38%	1.56	47%	1.61	54%
	WritingPrompts	3.71	23%	2.79	29%	2.64	31%
	Alpaca	2.65	31%	2.41	26%	2.52	36%
Open-RS2 (1.5B)	GSM8K	2.56	5%	2.49	6%	2.75	3%
	WritingPrompts	2.50	10%	2.23	12%	2.16	10%
	Alpaca	2.14	14%	2.02	16%	2.09	16%
Qwen3-8B	GSM8K	1.46	7%	1.40	7%	1.43	8%
	WritingPrompts	1.48	1%	1.42	7%	1.42	6%
	Alpaca	1.49	2%	1.44	-1%	1.46	4%

C.4 Length-Separation and Temperature

We use $T = 1$ for the main experiments, but the amount of within-prompt length variation depends strongly on sampling temperature. At $T = 0$, decoding is deterministic and there is no within-prompt variation to exploit. Figure 6 shows this relationship for AceReason-Nemotron-7B on GSM8K with $\alpha = 0.5$. As temperature increases from zero, both the oracle split and the separation achieved by our method increase, peaking at around $T = 1.3$. Beyond this point, the oracle split falls as responses become more uniformly long, and the separation achieved by our method falls with it. Thus, the method benefits from the greater within-prompt variation produced by higher temperatures, up to the point where that variation begins to collapse.

Table 3: Length separation for the remaining models. For each prompt, we generate 8 responses from the short and long regimes, compute the mean response length for each, and take the long-to-short ratio. Entries in the α rows are the average of these ratios over 480 prompts; 1.0 means no separation. The "–" row gives the mean response length under ordinary sampling, in tokens. Qwen3-8B shows essentially no separation across all three datasets. The $\pm$ values give 95% bootstrap confidence intervals, obtained by resampling the 480 prompts.

Dataset	α	R1-Qwen 1.5B	Qwen3 8B	Open-RS2 1.5B
GSM8K	–	820	1912	935
	0.25	1.83 $\pm$0.10	1.03 $\pm$0.01	1.08 $\pm$0.05
	0.5	1.84 $\pm$0.10	1.03 $\pm$0.01	1.09 $\pm$0.04
	0.75	1.89 $\pm$0.10	1.04 $\pm$0.01	1.06 $\pm$0.04
Writing Prompts	–	1130	1337	1166
	0.25	1.13 $\pm$0.03	1.01 $\pm$0.01	1.15 $\pm$0.04
	0.5	1.09 $\pm$0.03	1.03 $\pm$0.01	1.14 $\pm$0.04
	0.75	1.10 $\pm$0.03	1.03 $\pm$0.01	1.12 $\pm$0.04
Alpaca	–	927	1161	967
	0.25	1.13 $\pm$0.07	1.01 $\pm$0.01	1.16 $\pm$0.10
	0.5	1.13 $\pm$0.04	1.00 $\pm$0.01	1.16 $\pm$0.04
	0.75	1.15 $\pm$0.04	1.02 $\pm$0.01	1.17 $\pm$0.05

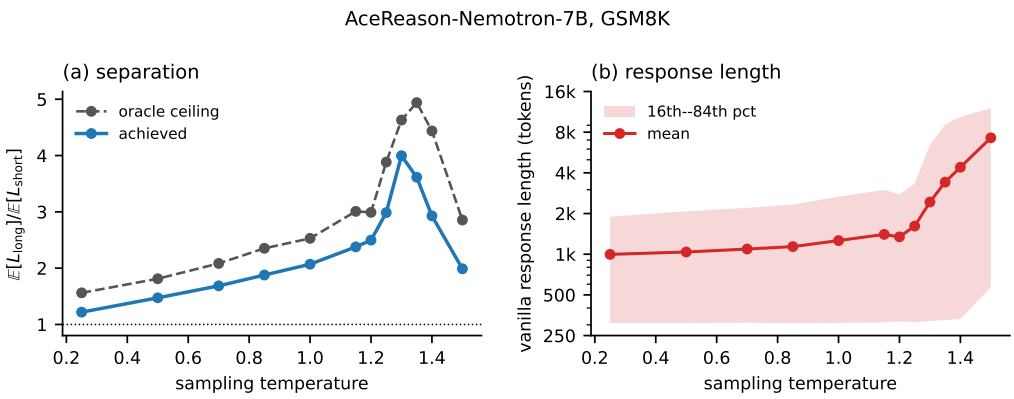

Figure 6: **Separation rises with sampling temperature, then falls as length variance collapses.** AceReason-Nemotron-7B on GSM8K ($\alpha = 0.5$). **(a)** Achieved separation compared to the "oracle" (maximum achievable split). **(b)** Mean response length with a 16th–84th percentile band. We observe two primary takeaways: (1) Increasing temperature initially increases response length variation (visible in the widening bounds in **b** and the rising oracle in **a**), which drives up achieved separation. (2) Around $T = 1.3$, separation peaks and begins to fall. Beyond this point, the model loses the ability to generate short responses; outputs become uniformly long, eliminating the variability required for effective splitting.

C.5 Does the predictor transfer across datasets?

Training one predictor for each model–dataset pair raises the question of how much of the learned signal is specific to a dataset. To assess this, for the models R1-Llama-8B, R1-Qwen-7B, R1-Qwen-1.5B, Qwen3-8B and Phi-4-mini and all three datasets we compared two predictors: one trained on forks from that dataset and one trained on WritingPrompts for the same model. We selected between them using predictive performance on held-out forks, without using the final response lengths or the resulting length separation.

The WritingPrompts predictor transferred successfully for two of the five models studied. For R1-Llama-8B and R1-Qwen-7B, it was selected for every dataset and retained 75–93% of the predictive correlation achieved on WritingPrompts.

For the remaining three models, we use dataset-specific predictors. Transfer can fail substantially: for Phi-4-Mini on GSM8K, the WritingPrompts predictor showed no usable predictive signal, whereas a GSM8K-trained predictor achieved about two-thirds of the correlation of the best possible predictor given the noise in the targets. Overall, the learned signal transfers across datasets for some models but no others. We did not evaluate transfer across models.

C.6 Ablation: how much of the split comes from the predictor?

Two things push the short regime towards shorter responses. The learned tilt of Eqn. 7 favours tokens that V expects to lead to short continuations, and, separately, termination tokens are always given the largest probability the envelope allows. The second is a fixed rule, not a learned signal, so we want to know how much of the split in Table 1 it produces on its own. To find out we rerun the sampler with the tilt removed ($\beta = 0$), leaving only the termination rule. This is still a valid sampler, since Theorem 4.1 holds for any candidate distribution; it simply has no length predictor.

Table 4 compares the two at $\alpha = 0.5$. On GSM8K the termination rule alone separates almost nothing (ratios 1.03–1.14) and the full method is far ahead for every model but Mistral: the large splits on GSM8K come from the predictor. On WritingPrompts and Alpaca the picture is mixed. For the R1 models, AceReason and R1-Qwen-14B, neither sampler separates much. For Mistral, DeepMath, Open-RS3 and Phi-4-Mini, the termination rule alone already gives most or all of the split, and adding the predictor does not improve on it; for three of these cells the validation-based choice of β selected $\beta = 0$, so the reported result is the termination rule itself.

Our reading is that the two mechanisms suit different tasks. Where a response can plausibly end at many points, as in open-ended writing or short chat replies, the termination rule already carries most of the usable length signal and the predictor has little to add. Another possibility, which we have not tested, is that the two mechanisms compete for the same divergence budget (Eqn. 8): a tilt carrying little length information still consumes headroom that the termination rule could have used later in the response, which would explain why adding the predictor occasionally shrinks the split, as for Phi-4-Mini on Alpaca.

D More than two regimes

The main text draws the regime from a Bernoulli variable: with probability α the short regime, otherwise the long one. Nothing in the construction requires this variable to be binary. Instead of dividing responses into two regimes, we can construct several regimes ordered from shortest to longest responses. We call this sequence a K-regime *chain*: each successive regime is constructed from the probability mass left after the preceding regimes. This appendix describes the construction and evaluates it empirically.

Construction. Fix probabilities $w_1, \dots, w_K$ summing to one, where w_k is the probability of selecting regime k, and write $m_k = \sum_{j>k} w_j$ for the probability left over after the first k regimes. I.e. m_{k-1} is the probability mass not yet assigned to regimes $1, \dots, k-1$, and a_k is the fraction of that remaining mass assigned to regime k. At each decoding step let p be the target model's next-token distribution, and set $\bar{p}_0 = p$. The chain is constructed sequentially: at each stage, one regime is extracted from the probability mass remaining after the preceding regimes. For $k = 1, \dots, K-1$,

$$a_k = \frac{w_k}{m_{k-1}}, \qquad p^{(k)} = \Pi_{a_k}\big(\mathrm{tilt}_k(\bar{p}_{k-1})\big), \qquad \bar{p}_k = \frac{\bar{p}_{k-1} - a_k\, p^{(k)}}{1 - a_k}, \tag{19}$$

Here $\bar{p}_k$ is the residual distribution left after constructing regimes $1, \dots, k$. We let the last regime take the final residual, $p^{(K)} = \bar{p}_{K-1}$. In other words, regime 1 is the short regime of the original target distribution; regime 2 is the short regime of the probability mass left by regime 1; and so on. The final regime receives whatever remains.

tilt_k is the length tilt of Eqn. 7 applied to $\bar{p}_{k-1}$ in place of p, and Π_a is the envelope projection (the runs below used a water-filling variant), enforcing the envelope condition $a\, q(y) \le \bar{p}_{k-1}(y)$ for every $y \in \mathcal{V}$.

Table 4: **Full method against a termination-only** p_{short}, at $\alpha = 0.5$. Same prompts, samples and statistic as Table 1. *term.* is the same construction with the learned tilt removed ($\beta = 0$), leaving only the termination rule. Bold marks the cells where the termination rule alone gives most of the split.

model	dataset	full	term.
AceReason-Nemotron-7B	GSM8K	2.78	1.10
	WritingPrompts	1.04	1.04
	Alpaca	1.12	1.06
DeepMath-1.5B	GSM8K	2.29	1.13
	WritingPrompts	1.26	**1.26**
	Alpaca	1.22	**1.23**
R1-Llama-8B	GSM8K	2.03	1.08
	WritingPrompts	1.24	1.03
	Alpaca	1.07	1.04
Open-RS3-1.5B	GSM8K	1.86	1.14
	WritingPrompts	1.12	**1.12**
	Alpaca	1.18	1.10
Mistral-7B-Instruct†	GSM8K	1.84	**1.88**
	WritingPrompts	1.84	**1.84**
	Alpaca	2.45	**2.56**
R1-Qwen-14B	GSM8K	1.62	1.05
	WritingPrompts	1.02	1.02
	Alpaca	1.08	1.02
R1-Qwen-7B	GSM8K	1.68	1.03
	WritingPrompts	1.09	1.03
	Alpaca	1.13	1.03
Phi-4-Mini†	GSM8K	1.26	1.05
	WritingPrompts	1.52	1.24
	Alpaca	1.36	**1.53**

After observing a generated token, the posterior probability of each regime is updated as

$$w_k \leftarrow w_k \, \frac{p^{(k)}(x)}{p(x)}, \tag{20}$$

For $K = 2$, this reduces exactly to the two-regime construction in Section 4, with $w_1 = \alpha$, $p^{(1)} = p_{\text{short}}$, and $p^{(2)} = p_{\text{long}}$.

Marginal preservation. At each stage, the current residual distribution is decomposed into the probability mass assigned to regime k and the residual passed to later regimes: $m_{k-1}\bar{p}_{k-1} = w_k p^{(k)} + m_k \bar{p}_k$ for each k. Summing over k telescopes to

$$\sum_{k=1}^{K} w_k \, p^{(k)}(y) = p(y) \qquad \forall y \in \mathcal{V} \tag{21}$$

at every decoding step. This is the K-term form of Lemma A.1, and the induction of Theorem 4.1 applies unchanged: routing a prompt to regime k with probability w_k reproduces ordinary sampling exactly.

Budgets. Each regime has a KL-Budget analogous to the short-regime budget in the two-regime construction. Let $a_k^{(n)} = w_k^{(n)} / m_{k-1}^{(n)}$ be regime k's share of the mass not claimed by regimes $1, \ldots, k-1$ after n tokens. Since $m_{k-1}\bar{p}_{k-1} = \sum_{j \geq k} w_j \, p^{(j)}$, the update in Eqn. 20 gives

$$\log a_k^{(n+1)} - \log a_k^{(n)} = \log \frac{p^{(k)}(x)}{\bar{p}_{k-1}(x)},$$

the same relation as in the two-regime case with $\bar{p}_{k-1}$ in place of p. Taking the expectation over $x \sim p^{(k)}$ and using $a_k^{(n)} \leq 1$, exactly as in Section 4.3, bounds the total divergence of regime k from the residual distribution $\bar{p}_{k-1}$ from which it is constructed:

$$\sum_n \mathbb{E}\left[\mathrm{KL}\left(p^{(k)} \,\middle\|\, \bar{p}_{k-1}\right)\right] \leq \log \frac{1}{a_k^{(0)}}. \tag{22}$$

With equal weights $a_k^{(0)} = 1/(K-k+1)$, so the budget is $\log(K-k+1)$: the first regime has $\log K$ and the last has none. Each regime is tilted short relative to a base that is itself the longer remainder of those before it, and this is what orders the chain. The ordering is not itself guaranteed: what is guaranteed is Eqn. 21 and the per-regime budget, and how cleanly the regimes separate depends, as in the main text, on the quality of the predictor.

Experiment. We evaluate K-regime chains with $K \in \{2, 3, 4, 6\}$ and equal routing weights $w_k := 1/K$, on R1-Llama-8B $\times$ GSM8K and Phi-4-Mini $\times$ WritingPrompts. Each run uses the same remaining-length predictor as the main experiments; only the tilt strength of regime k is rescaled, in closed form, to its budget $\log(1/a_k^{(0)})$. Each run uses 24 prompts and 16 samples per regime. For every K, the identity Eqn. 21 and the envelope conditions were asserted at every decoding step and never violated.

Results. Figure 7 plots each regime's mean and 90th-percentile response length against its index, and Table 5 gives the means. Two things stand out. First, the regimes come out in order, shortest to longest, in every run but one ($K = 6$ for Phi-4-Mini on WritingPrompts, where the middle regimes are not resolved at this sample size). Second, the spread between the shortest and longest regimes grows with K up to $K = 4$ (from 603–1,158 tokens at $K = 2$ to 463–1,406 at $K = 4$ on R1-Llama-8B) and then plateaus: $K = 6$ spans 478–1,427, adding intermediate regimes rather than a wider range.

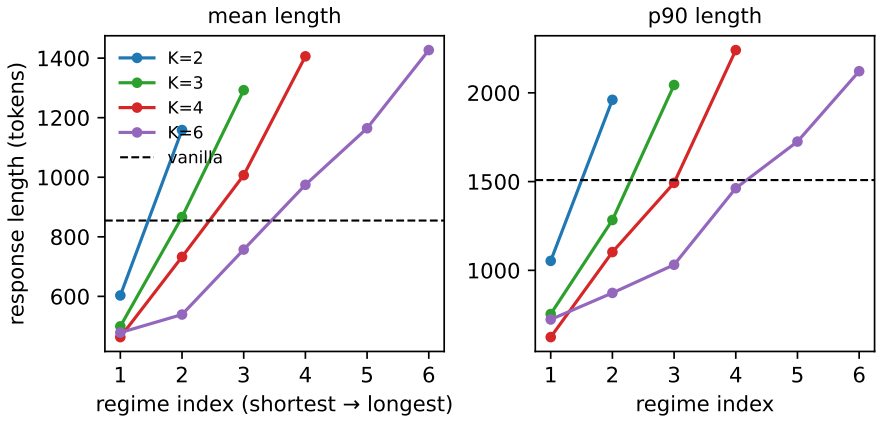

Figure 7: **A chain of K regimes ordered shortest to longest.** Mean (left) and 90th-percentile (right) response length against regime index, for $K \in \{2, 3, 4, 6\}$ with equal routing weights, on R1-Llama-8B $\times$ GSM8K. The dashed line is ordinary sampling. The same predictor is used for every K; only the tilt strength of each regime is adjusted. 24 prompts, 16 samples per regime.

Table 5: K-regime chain with equal weights: mean response length of each regime (tokens). 24 prompts, 16 samples per regime, with vanilla drawn afresh in each run.

cell	K	–	regimes $1 \ldots K$
R1-Llama-8B / GSM8K	2	835	603, 1158
	3	851	499, 867, 1292
	4	855	463, 733, 1007, 1406
	6	878	478, 539, 757, 975, 1164, 1427
Phi-4-Mini / WritingPrompts	2	454	438, 505
	3	477	394, 474, 499
	4	425	390, 454, 483, 531
	6	440	360, 429, 484, 450, 478, 534