

KAD: A Framework for Proxy-based Test-time Alignment with Knapsack Approximation Deferral

Ayoub Hammal¹ Pierre Zweigenbaum¹ Caio Corro²

¹Université Paris-Saclay, CNRS, LISN

²INSA Rennes, IRISA, CNRS, Université de Rennes

{ayoub.hammal,pz}@lisen.fr caio.corro@irisa.fr

Abstract

Several previous works concluded that the largest part of generation capabilities of large language models (LLM) are learned (early) during pre-training. However, LLMs still require further alignment to adhere to downstream task requirements and stylistic preferences, among other desired properties. As LLMs continue to scale in terms of size, the computational cost of alignment procedures increase prohibitively. In this work, we propose a novel approach to circumvent these costs via proxy-based test-time alignment, *i.e.* using guidance from a small aligned model. Our approach can be described as token-specific cascading method, where the token-specific deferral rule is reduced to 0-1 knapsack problem. In this setting, we derive primal and dual approximations of the optimal deferral decision. We experimentally show the benefits of our method both in task performance and speculative decoding speed.

1 Introduction

Large language models’ (LLM) alignment¹ is employed for reshaping the pre-trained model’s output distribution so that it adheres to expected (human) preferences, formatting or instructions, and more generally downstream task requirements (Bai et al., 2022; Kumar et al., 2025; Ouyang et al., 2022). A variety of fine-tuning approaches are employed with different computational requirements and overheads, such as reinforcement learning from human feedback (RLHF, Ziegler et al., 2020; Rafailov et al., 2023; Ethayarajh et al., 2024). The alignment phase follows an already costly pre-training phase, and alignment costs scale prohibitively with model sizes, meaning that tailoring LLMs for diverse real-world scenarios is challenging. For example, the RLHF stage of Tulu 3 scales from 520 H100 GPU

hours for the 8B model to 2 880 hours for the 70B model and to 11 776 hours for the 405B model, following an exponential trend (Lambert et al., 2025)

To avoid paying the high cost of fine-tuning a large model, previous works have proposed to adapt the pre-trained model’s output distribution at test-time (Welleck et al., 2024), using external guidance either from a reward signal (Deng and Raffel, 2023; Troshin et al., 2025), implicitly extracting the reward from a small aligned model (Liu et al., 2024; Mitchell et al., 2024) or using a mixture of a large unaligned model and a small aligned model (Lu et al., 2023; Fei et al., 2025). As such, when novel alignment specifications are needed, one only need to either train a (simple) reward or align a small model, drastically reducing alignment time, costs and even lowering the hardware requirement.

In this work, we follow the *proxy-based alignment* methodology where an unaligned large model distribution is “distorted” at test-time using a small aligned model. One of the motivations of test-time alignment is that previous analysis has shown that most of a LLM’s generation capabilities is acquired during pre-training (Zhou et al., 2023), making the following steps mostly intervene on stylistic and transitional tokens positions (Chang et al., 2024; Hu et al., 2024). Following this observation, Fei et al. (2025) proposed to use the large model’s confidence as an alignment indicator: if the unaligned large model distribution has a low-entropy, the next token is sampled from it, otherwise the generation is deferred to a smaller aligned model that assists in “nudging” the response.

Our main contribution is a novel framework for proxy-based alignment. Instead of fully deferring the decision to the small model when the large one has a high entropy, we build a mixture distribution of the two models. Our intuition is as follows: we should defer the decision to the small model for low-probability tokens. We model this deferral rule as a *binary knapsack* problem (Martello and Toth,

¹The term “alignment” has several different meanings in the literature. In this work, we call alignment the supervised training process of a LLM following the (~unsupervised) pre-training phase.

1990), where one may fix a maximum mass (budget) of probability whose decision can be deferred to the small model. As the binary knapsack problem is known to be intractable (Karp, 1972), we propose dual and primal approximations of the optimal deferral condition. Overall, this framework allows us to derive several deferral rules (depending, for example, on the loss function used to measure the quality of generations), including the nudging model of Fei et al. (2025). Our approach, while adding more degrees of freedom to the deferral rule, allows to focus deferral on specific uncertain tokens, leveraging richer token-level information rather than a single global confidence score. We furthermore show experimentally that token-specific deferral decision is faster than distribution-level decision in comparable speculative decoding setups.

Our contributions can be summarized as follows: (1) we introduce KAD (knapsack approximation deferral), a framework for proxy-based alignment; (2) we derive several dual and primal approximate solutions and provide theoretical guarantees on the regret and error-rate achieved by them while linking our propositions with previous work; (3) we experiment with OLMo 2 and Qwen 3, showing the performance improvement brought by our approach compared to several baselines; and (4) we show our method leads to increased generation speed when combined with speculative decoding.

Our implementation is publicly available.²

Notations. We write Iverson brackets as $\llbracket \cdot \rrbracket$, taking the value 1 if the inner condition is true and 0 otherwise. For any $k \in \mathbb{N}_{\geq 1}$, $\Delta(k)$ is the $(k - 1)$ -dimensional simplex. To simplify notations, we denote the token vocabulary as $V = \{1, \dots, |V|\}$.

An autoregressive language model is a conditional distribution of the next token given the previously generated ones. Let p be such a language model, we can then write the next token distribution as $p(x_t | \mathbf{x}_{1:t-1} = \mathbf{x}_{1:t-1})$. To simplify notations, we will drop the dependence on history $\mathbf{x}_{1:t-1}$ and write the model’s conditional probability distribution as a vector $\mathbf{p} \in \Delta(|V|)$ defined as follows:

$$p_v = p(x_t = v | \mathbf{x}_{1:t-1} = \mathbf{x}_{1:t-1}),$$

i.e. p_v is the probability of token v given history $\mathbf{x}_{1:t-1}$. More generally, all variables and functions presented are assumed to be context-dependent unless stated otherwise.

²<https://github.com/ayoubhammal/knapsack-approximation-deferral>

2 Motivations

The term *base*³ language model (LM) refers to an LM that has not undergone specific instruction or preference tuning via supervised fine-tuning, reinforcement learning from human feedback (RLHF), or any other related methods. A model resulting from the latter procedures is called an *aligned* LM.

For the purpose of this paper, we will consider models of different sizes and at different training stages. We denote $\mathbf{p}$ and $\mathbf{p}^*$ a large LM in its base and aligned variants, respectively. Similarly, $\mathbf{q}$ and $\mathbf{q}^*$ refer to small (base and aligned) LMs, where “small” means that generation with these models is significantly faster than with the large ones. Note that $\mathbf{p}$, $\mathbf{p}^*$, $\mathbf{q}$ and $\mathbf{q}^*$ are used to denote both language models and associated conditional distribution vectors. The only extra assumption we introduce is that the large and small LMs share the same tokenization vocabulary.

2.1 Background

In order to avoid high alignment costs, there is an interest in test-time alignment, where a large base LM is guided to achieve desired outputs directly during generation, without extra tuning steps. *Proxy tuning* is one such approach, where a small language model is aligned and is used to guide the large one (Liu et al., 2024; Mitchell et al., 2024).

Interestingly, several authors hypothesized that the alignment training phase mainly impacts style, discourse structure, and other superficial features, but that most of the actual “knowledge” is acquired during the base model training (Zhou et al., 2023; Chang et al., 2024; Hu et al., 2024). Lin et al. (2024) showed that the alignment procedure changes the base distribution only for a few output positions. In particular, transitional and structural positions are shifted toward more stylistic tokens such as those showing agreement, e.g. “of course”, or positivity, e.g. “great question”, conveying less information but helping set the tone of the response.

Fei et al. (2025) further show that the base model’s certainty is tightly related to the agreement between the base and aligned models’ distributions: the top predictions of both models tend to agree more when the base model’s top probability is higher. From this observation, they propose *nudging* the base model’s generation when its certainty falls below a specific threshold $\lambda \in \mathbb{R}_{\geq 0}$, that

³Also called *pre-trained* language model.

is, sampling from q^* only for positions where p is uncertain. In other words, the sampling procedure for the next token is:⁴

- if $\max_v p_v \geq \lambda$, sample from p ;
- otherwise, sample from q^* .

2.2 Broader Perspective

Nudging can be interpreted as a specific case of *model cascades*, an ensembling technique that relies on a sequence of classifiers: each classifier in the sequence can choose to either return the prediction or defer the prediction to the next model in the sequence (Narasimhan et al., 2025; Varshney and Baral, 2022; Wang et al., 2018; Li et al., 2021).

First, notice that we can rewrite the nudging procedure as sampling from a mixture distribution $\phi \in \Delta(|V|)$ defined as follows:

$$\phi_v = \begin{cases} q_v^* & \text{if } \max_w p_w < \lambda, \\ p_v & \text{otherwise.} \end{cases} \quad (1)$$

The condition $\max_w p_w < \lambda$ is the position-dependent *deferral rule*, whose value indicates the *deferral decision*, that is when the first model in the cascade p must defer the decision to the (single) next model q^* .

From this perspective, we see that nudging as proposed by Fei et al. (2025) is a special case of model cascades with a sequence of 2 classifiers, where the second model q^* is invoked when the first p is uncertain, a condition which is implemented using the so-called Chow’s rule (Chow, 1970; Jitkrittum et al., 2023; Gupta et al., 2024). As such, an obvious question is whether we can use different deferral rules.

3 Knapsack Approximation Deferral

As described in the previous section, the nudging technique for proxy-based alignment can be interpreted as a specific case of cascading. Based on this observation, we first introduce a framework that allows to derive token-specific mixtures for test-time alignment based on a loss function that measures the base model’s uncertainty, and a maximum deferral budget. Then, we propose two approximation methods of the optimal deferral decision.

⁴In Fei et al. (2025), the nudging step generates the next word until the following space instead of a single token.

3.1 Formalization

We seek to build a mixture distribution π using a token-specific deferral decision $d \in \{0, 1\}^{|V|}$ (Narasimhan et al., 2025):

$$\pi_v = p_v(1 - d_v) + q_v^* \alpha \quad (2)$$

where α is a normalization term that ensures the distribution π is well-defined:

$$\alpha = \sum_{w \in V} p_w d_w.$$

The deferral decision d allows to defer decision to q independently for each token, and as such generalizes the nudging procedure described in Sec. 2.2.

We are now left with defining rules used to compute the deferral decision vector d . As explained in Sec. 2.1, alignment seems to only shift the probability of a few tokens, especially when the model is *unsure* about an outcome. To measure the uncertainty about a specific output, we rely on a loss function $\ell : \Delta(k) \times V \rightarrow \mathbb{R}_{\geq 0}$. We define the *risk* (Shalev-Shwartz and Ben-David, 2014) associated with d as:

$$r(d) = \sum_{v \in V} \mathbb{P}_v \ell(p, v)(1 - d_v),$$

where $\mathbb{P}_v = \mathbb{P}(x_t = v | x_{1:t-1} = x_{1:t-1})$ is the (unknown) ground-truth target distribution.

Notice that naive risk minimization leads to a trivial deferral decision as $r(\mathbf{1}) = 0$, *i.e.* ignoring the base model p and deferring the outcome to q^* only. Instead, we assume a maximum deferral budget $b > 0$, that is the maximum probability mass that can be ignored in p . We obtain the following constrained risk minimization problem:

$$(P1) \quad \min_{d \in \{0,1\}^{|V|}} r(d) \quad \text{s.t.} \quad \sum_{v \in V} \mathbb{P}_v d_v \leq b.$$

We denote $\hat{d}$ an optimal solution of (P1).

Lemma 1. Computing the optimal deferral rule $\hat{d}$ is NP-Hard.

The lemma can be proved by observing that (P1) is equivalent to the 0-1 knapsack optimization problem, which is known to be NP-Hard (Cacchiani et al., 2022). Although exact dynamic programming algorithms are known for solving this problem, their worst-case complexity is exponential in the vocabulary size (Martello and Toth, 1990; Pisinger, 1997; Martello et al., 1999). Therefore, we will instead rely on fast dual and primal approximations, which are guaranteed to define bounds on the optimal risk.

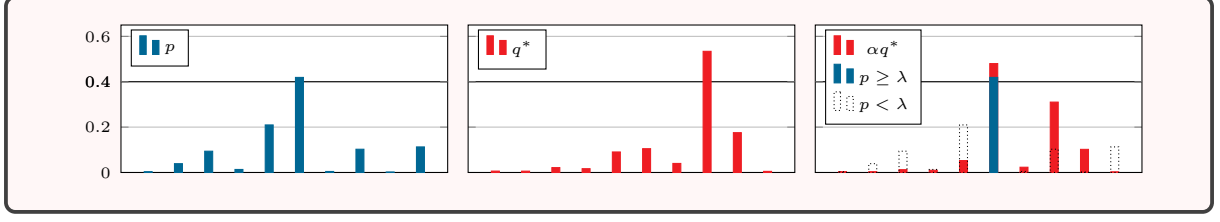


Figure 1: Illustration of token-specific cascading distribution $\pi^{<\lambda}$ with $\lambda = 0.4$. The left and center plots show the probability mass function (PMF) of p and q^* , respectively. The right plot shows the PMF of $\pi^{<\lambda}$, where the blue parts shows the mass coming from p and the red parts the one coming from q^* (including rescaling by α). Dotted bars show that mass from p that was rejected by the deferral decision. We can observe the most probable token of p and the one of q^* both have high probabilities in the resulting mixture.

3.2 Dual Approximation

In this section, we introduce a first approximation based on duality. This approximation does not rely on the ground-truth distribution $\mathbb{P}$.

By relaxing the budget constraint, we build the following Lagrangian (Lemar  chal, 2001):

$$L(\mathbf{d}, \mu) = r(\mathbf{d}) + \mu \left(\sum_{v \in V} \mathbb{P}_v d_v - b \right),$$

where $\mu \in \mathbb{R}_{\geq 0}$ is called a Lagrangian multiplier or dual variable. In other words, the hard constraint has been replaced by a penalty in the objective. Our dual approximation is simply defined as a dual solution for a given multiplier μ :

$$\hat{\mathbf{d}}^-(\mu) \in \arg \min_{\mathbf{d} \in \{0,1\}^{|V|}} L(\mathbf{d}, \mu).$$

Note that $\hat{\mathbf{d}}^-(\mu) \in \mathbb{R}^{|V|}$ may not be primal feasible, *i.e.* it may not satisfy the budget constraint.

Lemma 2. For any dual variable $\mu \geq 0$, the deferral decision vector $\hat{\mathbf{d}}^-(\mu)$ is defined as:

$$\forall v \in V : \left[\hat{\mathbf{d}}^-(\mu) \right]_v = \llbracket \ell(\mathbf{p}, v) > \mu \rrbracket.$$

where the right-hand side condition is called the deferral condition. Moreover, the following bound on the true risk holds:

$$L(\hat{\mathbf{d}}^-(\mu), \mu) \leq r(\hat{\mathbf{d}}).$$

Proof of Lemma 2 can be found in Appendix A. For the negative log-likelihood (or logistic) loss, we can rewrite the dual approximation as follows:

$$\left[\hat{\mathbf{d}}^-(\mu) \right]_v = \llbracket -\log p_v > \mu \rrbracket = \llbracket p_v < \exp(-\mu) \rrbracket.$$

Loss	$\ell(\mathbf{p}, v)$	$\left[\hat{\mathbf{d}}^-(\mu) \right]_v$
Logistic	$-\log p_v$	$\llbracket p_v < \exp(-\mu) \rrbracket$
Squared	$(1 - p_v)^2$	$\llbracket p_v < 1 - \sqrt{\mu} \rrbracket$
Perceptron	$-p_v + \max_w p_w$	$\llbracket p_v < \max_w p_w - \mu \rrbracket$
0-1	$\llbracket v \notin \arg \max_w p_w \rrbracket$	$\llbracket p_v < \max_w p_w \rrbracket$
Min-ent.	$-\log \max_w p_w$	$\llbracket \max_w p_w < \exp(-\mu) \rrbracket$

Table 1: Different losses and their associated deferral rules in the dual approximation scenario. For the 0-1 loss, we assume that $\mu \in (0, 1)$.

For any loss of the form $\ell(\mathbf{p}, v) = f(p_v)$ where f is continuous and strictly decreasing, we have:

$$\left[\hat{\mathbf{d}}^-(\mu) \right]_v = \llbracket p_v < f^{-1}(\mu) \rrbracket. \quad (3)$$

To simplify notation, we will denote $\pi^{<\lambda}$ the distribution given by such decision rule:

$$\pi_v^{<\lambda} = p_v \times \llbracket p_v \geq \lambda \rrbracket + q_v^* \alpha,$$

where $\lambda = f^{-1}(\mu)$. Figure 1 illustrates such a mixture.

Further examples with different losses are presented in Table 1 and their derivations in App. B.

Chow’s rule. We now show that our framework generalizes the nudging method of Fei et al. (2025). Consider the min-entropy loss function (Renner and Wolf, 2004) defined as follows:

$$\ell(\mathbf{p}, v) = -\log \max_{w \in V} p_w.$$

Note that this loss function is unsupervised, in the sense that the loss value does not depend on the gold token v . With this loss, the deferral decision is:

$$\left[\hat{\mathbf{d}}^-(\mu) \right]_v = \llbracket \max_{w \in V} p_w < \exp(-\mu) \rrbracket,$$

which is Chow’s rule used for nudging, which we recover here as a dual approximation.

3.3 Primal Approximation

The proposed dual approximation does not depend on $\mathbb{P}$, but requires setting a hyperparameter. In this section, we derive a method to build a primal feasible solution using the Lagrangian dual problem. As computing the solution requires access to the ground-truth distribution, we rely on a plug-in approximation of $\mathbb{P}$ in practice.

From Lemma 2, we know that for any $\mu \in \mathbb{R}_{\geq 0}$, the Lagrangian gives a lower bound to the optimal risk. As such, we seek the dual variable value $\hat{\mu}$ that maximizes this bound:

$$\hat{\mu} \in \arg \max_{\mu \geq 0} L(\mu),$$

where $L(\mu) = \min_{\mathbf{d} \in \{0,1\}^{|V|}} L(\mathbf{d}, \mu)$.

Lemma 3. Wlog, we assume the vocabulary is sorted in increasing loss order, that is:

$$\forall v, w \in V : v < w \implies \ell(\mathbf{p}, v) < \ell(\mathbf{p}, w).$$

Let $\hat{k}$ be defined such that the following condition holds:

$$\sum_{v=\hat{k}+1}^{|V|} \mathbb{P}_v \leq b < \sum_{v=\hat{k}}^{|V|} \mathbb{P}_v. \quad (4)$$

Then, we have $\hat{\mu} = \ell(\mathbf{p}, \hat{k})$. We call $\hat{k}$ the critical element index.

The proof can be found in Appendix C.

The primal approximation is then defined as:

$$\hat{\mathbf{d}}^+ = \hat{\mathbf{d}}^-(\hat{\mu}).$$

As $\hat{\mathbf{d}}_v^+ = \mathbb{I}[\ell(\mathbf{p}, v) > \ell(\mathbf{p}, \hat{k})]$, and assuming that the vocabulary is sorted in increasing loss order, we can see that the following inequality holds:

$$\sum_{v \in V} \hat{\mathbf{d}}_v^+ \mathbb{P}_v = \sum_{v=\hat{k}+1}^{|V|} \mathbb{P}_v \leq b.$$

Thus $\hat{\mathbf{d}}^+$ is primal feasible, and the following bound holds:

$$r(\hat{\mathbf{d}}) \leq r(\hat{\mathbf{d}}^+).$$

Plugin approximation. Unfortunately, computing the critical element index $\hat{k}$ requires access to the ground-truth data distribution $\mathbb{P}$, which is unknown. Therefore, we follow the methodology

of Narasimhan et al. (2025) and use a plugin approximation of $\mathbb{P}$, that is we rely on $\mathbf{p}$ as an approximation of $\mathbb{P}$. More precisely, we compute an approximation $\tilde{\mu}$ of the optimal Lagrangian multiplier using an approximation of the critical index element $\tilde{k}$:

$$\tilde{\mu} = \ell(\mathbf{p}, \tilde{k}) \quad \text{s.t.} \quad \sum_{v=\tilde{k}+1}^{|V|} p_v \leq b < \sum_{v=\tilde{k}}^{|V|} p_v.$$

The primal approximation $\tilde{\mathbf{d}}^+ = \hat{\mathbf{d}}^-(\tilde{\mu})$ of the deferral rule is defined as follows:⁵

$$\forall v \in V : \tilde{\mathbf{d}}_v^+ = \mathbb{I}[\ell(\mathbf{p}, v) > \tilde{\mu}].$$

3.4 Theoretical Analysis

In this section, we show some theoretical properties of primal approximations.

First, it is interesting to note that the previously considered decreasing loss functions all produce the same primal approximation as per Lemma 4. Therefore, although there is an interest of evaluating different losses for the dual approximation, the primal one always lead to the same deferral rule for a wide class of losses (Lemma 4). Second, we show that the quality of the approximation using $\mathbf{p}$ as a plugin estimator depends on the total variation distance between $\mathbf{p}$ and $\mathbb{P}$ (Lemma 5).

Lemma 4. A loss function ℓ is *order inverting* if, $\forall v, w \in V$, we have:

$$p_v < p_w \iff \ell(\mathbf{p}, w) < \ell(\mathbf{p}, v).$$

Let ℓ_1 and ℓ_2 be two order inverting losses. Then, ℓ_1 and ℓ_2 produce the same critical element index $\hat{k}$ and the same approximation $\tilde{k}$. Moreover, they produce the same deferral decision $\hat{\mathbf{d}}^+$, and the same approximation $\tilde{\mathbf{d}}^+$.

Lemma 5. The following bound on the risk holds:

$$|r(\hat{\mathbf{d}}^+) - r(\tilde{\mathbf{d}}^+)| \leq \ell(\mathbf{p}, u) \left(\frac{D_{\text{TV}}(\mathbb{P}, \mathbf{p})}{\sum_{v=u}^{|V|} \mathbb{P}_v} \right),$$

where $l = \min(\hat{k}, \tilde{k}) + 1$ and $u = \max(\hat{k}, \tilde{k})$. Moreover, we have:

$$r(\hat{\mathbf{d}}^+) - r(\hat{\mathbf{d}}) \leq \ell(\mathbf{p}, \hat{k}) \mathbb{P}_{\hat{k}}.$$

Proofs are given in Appendices E and F.

⁵Pseudo-code is given in Appendix G.

4 Speculative Decoding

In the nudging approach of Fei et al. (2025), they only need to forward in the large model p when its certainty is high, and calls to q^* are limited to a few tokens for which the decision is deferred to it, see Equation (1). On the contrary, our mixture distribution defined in Equation (2) relies on both p and q^* as soon as there is a single element of the deferral decision equal to 1. Hence, a naive use of our approach will lead to a slower generation.

In this section, we propose to accelerate generation using *speculative decoding* (Leviathan et al., 2023; Chen et al., 2023), a two step sampling process. First, in the drafting step, a *draft model* is used to sample a draft of $\gamma \in \mathbb{Z}_{>1}$ tokens. Then, in the verification step, tokens are sequentially accepted using a *target model*, until one token is rejected. The resulting sampling process is guaranteed to match the target model distribution. In practice, the draft model is assumed to be small and fast, whereas the target model is large and slow. The forward pass in the target model can be efficiently parallelized over all tokens in the draft instead of generating tokens autoregressively, which leads to experimental speed improvement. We refer readers to (Leviathan et al., 2023) for further details.

In our case, the target distribution is the mixture π , and the proposal distribution is the aligned model q^* . The probability $a(\pi, q^*) \in [0, 1]$ of accepting the next token in the draft is equal to:

$$a(\pi, q^*) = 1 - D_{TV}(\pi, q^*).$$

As such, a lower divergence between the target and the draft models means a higher acceptance probability, and therefore a faster speculative procedure.

Lemma 6. The following bound on $D_{TV}(\pi, q^*)$ holds:

$$D_{TV}(\pi^{<\lambda}, q^*) \leq D_{TV}(\phi, q^*) + \alpha(1 - \beta),$$

where $\beta = \sum_v d_v q_v^*$.

Moreover, if the following condition are satisfied:

1. $\max_{v \in V} p_v \geq \lambda$,
2. $\forall v : p_v < \lambda \implies p_v \leq \alpha q_v^*$,

then, the following bound holds:

$$TV(\pi^{<\lambda}, q^*) \leq D_{TV}(\phi, q^*).$$

The proof can be found in Appendix H. Lemma 6 shows that even if our mixture distribution $\pi^{<\lambda}$ is more expressive than the nudging distribution ϕ , the speculative decoding process will only have a slightly lower acceptance rate, as $\alpha(1 - \beta) \ll 1$. The second part of theorem even show that in many cases, the acceptance will be at least as good, and even better. We will confirm these results experimentally in Section 6.

5 Related work

Test-time alignment. To avoid the high cost of LLM alignment via parameter fine-tuning, different previous work considered to leverage a reward signal to steer the generation of the base model. Unlike our approach, these methods require a specifically trained reward model to evaluate partial sequences to guide the decoding step from the base model (Khanov et al., 2024; Deng and Raffel, 2023; Troshin et al., 2025). Mitchell et al. (2024) and Liu et al. (2024) proposed extracting the partial reward from a smaller aligned model, but their approach requires both the base and aligned checkpoints of the smaller model. Alternatively, Faria and Smith (2025) proposed a Metropolis-Hastings sampling approach that can use a more standard reward model defined on complete outputs.

Our test-time alignment method is more straightforward as it only requires the aligned small model and isolates its contribution to the final mixture distribution to the weak components of p . Additionally, it does not require sampling several responses for each prediction.

Model cascading. Model cascading is a popular technique used in computer vision (Wang et al., 2018) and natural language processing (Varshney and Baral, 2022; Li et al., 2021), with an emphasis on reducing generation cost by evaluating larger models only when necessary. By design, cascading can only defer to larger models using information about the the smaller model outputs. For text generation, Narasimhan et al. (2025) proposed to mix cascading and speculative decoding to build deferral rules based on larger model outputs while ensuring fast generation. Our work differ in its goal, that is proxy-based test-time alignment of a large base model, which leads to a different definition of the deferral rule and resulting output distributions.

Knapsack formulation. Nishikawa et al. (2014) and Tonglet et al. (2023) proposed reductions to

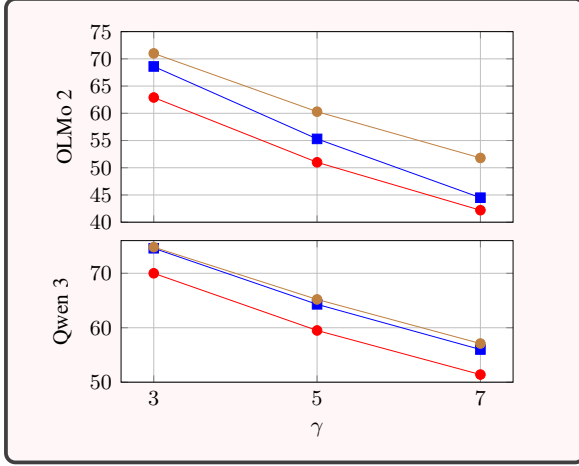


Figure 2: Empirical acceptance rate per model in speculative generation schema. In brown the dual approximation $\pi^{<\lambda}$ with $\lambda = 0.4$, in blue the primal approximation with $b = 0.9$ and in red Nudging with $\lambda = 0.4$.

the 0-1 knapsack problem for summarization and in-context examples selection, respectively.

6 Experiment

Datasets. We evaluate our method on three *math reasoning* datasets, GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2023) and SVAMP (Patel et al., 2021), and three *commonsense reasoning* datasets, ARC-Challenge (Clark et al., 2018), CommonsenseQA (Talmor et al., 2019) and TruthfulQA (Lin et al., 2022). We use standard prompting and evaluation strategies for all datasets. Examples are given in Appendix I.

Models. We use two LLM families for experiments. First, OLMo 2 (OLMo et al., 2025), which is a fully open source model family. Second, Qwen 3 (Yang et al., 2025), which is relatively closed source. We select 2 contrasting sizes of each family on which we carry out our experiments. The choice of the model families is based on the available model sizes and checkpoints.

Unlike Fei et al. (2025), we do not employ greedy decoding. Instead, we sample from the unmodified distributions with temperature 0.7 as recommended for each model to isolate the benefits of our method.

All the experiments were run on single Nvidia A100-80GB GPUs. The evaluation of each dataset was parallelized on 8-GPU nodes.

Test-time methods. For our approach, we evaluate both with dual and primal approximations.

For the dual approximation case, we test 3 loss functions. For the primal approximation case, we shown that all (non-degenerated) losses lead to the same deferral rule, therefore will only test with the negative log-likelihood.

We compare our approach against two baselines: (1) nudging (Fei et al., 2025), and (2) the implicit reward approach introduced by Mitchell et al. (2024) and Liu et al. (2024). The implicit reward builds a mixture in which the probability of token $v \in V$ is proportional to $p_v \frac{q_v^*}{q_v}$. Note this approach requires access to both q^* and q , whereas nudging and ours only require access to q^* .

Hyperparameters. For our mixture $\pi^{<\lambda}$, we follow Fei et al. (2025) and set $\lambda \in \{0.3, 0.4\}$. We do not explore further specific tuning for our token-specific deferral rule, showing that it can serve as a plug-in replacement for the distribution-level deferral rule with no further hyperparameter search. As for the other deferral rules, we performed a quick hyperparameter scan on a small data subset and selected promising hyperparameter values, that is $b \in \{0.6, 0.9\}$ and $\mu \in \{0.05, 0.1\}$.

6.1 Performance evaluation

The main experimental results are summarized in Table 2. As expected, an aligned model always performs better than its base model, and a larger base (resp. aligned) model always performs better than its smaller base (resp. aligned) model. With OLMo 2, q^* is nearly always better than the larger base model p , whereas this only happens in two out of six datasets for Qwen 3. For a given model, scores vary widely across datasets, e.g. from 9.4 (MATH500) to 57.6 (SVAMP) for the base OLMo 2. Therefore, it is convenient to focus on average scores over the six datasets.

All test-time alignment methods in the table consistently reach an average performance above that of the base p and q and of the small aligned q^* model: in other words, they all play the expected role. While our dual approximation allows to recovers Chow’s rule used in the nudging method of Fei et al. (2025), our other proposed generalizations approximations obtain average scores that are always above that of the nudging method. They are also always above that of the implicit reward method except once (primal approximation with OLMo 2 and $b = 0.6$ results in lower accuracy than implicit reward).

Overall, our approach obtains the highest av-

	GMS8K MATH SVAMP ARC CSQA TQA Avg.							GMS8K MATH SVAMP ARC CSQA TQA Avg.						
	OLMo 2 (1B and 13B)							Qwen 3 (1.7B and 14B)						
Building blocks														
p	54.5	09.4	57.6	29.6	19.4	29.3	33.3	75.5	51.8	80.0	86.6	76.9	57.8	71.4
q^*	62.5	16.4	70.3	43.8	48.4	22.3	43.9	75.3	53.0	86.6	82.9	68.7	43.2	68.2
Nudging (Fei et al., 2025)														
$\lambda=0.3$	60.3	13.8	67.3	53.7	33.4	38.5	44.5	78.1	52.8	85.0	88.8	78.5	57.0	73.3
$\lambda=0.4$	61.9	18.8	66.6	57.8	46.3	38.5	48.3	79.5	57.8	87.3	89.6	79.3	60.7	75.7
Our work: dual approximation														
$\hookrightarrow$ 0-1 loss														
$\mu \in (0, 1)$	68.8	24.6	74.3	58.9	48.7	38.3	52.2	81.7	59.2	89.6	92.1	82.0	58.1	77.1
$\hookrightarrow$ perceptron loss														
$\mu = 0.05$	68.2	22.0	74.3	55.8	46.1	37.6	50.6	82.4	59.6	89.6	91.4	81.4	60.0	77.4
$\mu = 0.1$	67.7	21.4	73.6	57.3	44.6	39.2	50.6	82.0	60.0	89.0	92.1	81.9	58.6	77.2
$\hookrightarrow \pi^{<\lambda}$ with $\lambda = f^{-1}(\mu)$														
$\lambda = 0.3$	69.5	22.8	76.6	61.6	52.3	42.8	54.2	81.3	59.6	89.0	91.4	81.3	59.6	77.0
$\lambda = 0.4$	72.3	23.4	75.3	61.9	55.6	40.2	54.7	81.7	60.6	87.3	91.5	80.7	58.3	76.6
Our work: primal approximation														
$b=0.6$	65.5	21.0	74.3	54.9	42.8	38.1	49.4	81.6	59.8	87.6	91.3	81.4	58.3	76.6
$b=0.9$	68.3	23.8	70.6	59.0	49.2	38.6	51.5	81.6	60.6	88.6	90.7	81.5	59.7	77.1
For reference														
p^*	84.3	39.6	87.6	82.5	76.9	53.6	70.7	82.4	64.0	88.3	93.8	83.1	70.1	80.2
q	14.4	01.0	17.6	19.6	15.8	18.8	14.5	21.9	12.2	25.3	47.1	29.5	15.0	25.1
Imp. reward	58.4	18.2	73.0	63.3	55.8	30.1	49.8	80.7	60.6	89.0	88.9	78.1	55.2	75.4

Table 2: Model’s accuracy. Best results on each column are in bold, second best are underlined.

erage accuracy for OLMo 2 and also the highest average over OLMo 2 and Qwen 3.

For Qwen 3, each primal approximation obtains at least one average score within the small range [77.0, 77.4]. This plateau effect might be related to the relatively small distance between p (71.4) and p^* (80.2), where the base model p already performs fairly high, and to the fact that the small aligned model q^* does not outperform the larger base model p , hence possibly making it more difficult to provide relevant guidance to it. In contrast, for OLMo-2, the best average score of each primal approximation varies within the larger range [50.6, 54.7]. This might be linked to the much larger gap between p (33.3) and p^* (70.7) and to the fact that the small aligned model q^* performs better than the larger base model p .

Further experimental analysis and motivation are presented in Appendix J.

6.2 Speed evaluation

Figure 2 shows the empirically calculated acceptance rates for ϕ and $\pi^{<\lambda}$ across different speculative window sizes $\gamma \in \{3, 5, 7\}$. We can see in

this figure that $\pi^{<\lambda}$ always achieve higher acceptance rates than ϕ across the 3 window sizes, with our dual approximation achieving slightly higher acceptance rates than the primal approximation.

This higher acceptance rate for $\pi^{<\lambda}$ translates into higher throughput in speculative generation mode as shown in Table 4 in Appendix K.

7 Conclusion

In this paper, we introduce a novel framework for test-time alignment based on a 0-1 knapsack problem. This framework allows to derive several dual and primal approximations, that all take the form of a token-specific deferral rules similar to cascading (Narasimhan et al., 2025). We proposed different instances of these deferral rules, generalizing existing work, and evaluated experimentally their performance against strong baselines.

Limitations

While our approach cuts training costs by a considerable margin, it draws its guidance from a smaller less capable model and it slightly underperforms the larger aligned model p^* . Although hyperpa-

rameters choices such as that of the budget b and the threshold λ does not heavily influence the mixture performance, they still need to be tuned individually for each model and dataset on a separate validation set. We however show in this paper that a fixed hyperparameter choice can safely be carried between models and datasets with marginal performance degradation.

Acknowledgments

We thank Miguel Couceiro for his comments and suggestions.

This work is supported by the SEMIAMOR (CE23-2023-0005) and InExtenso (ANR-23-IAS1-0004) project grants given by the French National Research Agency (ANR). This work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011015801 made by GENCI.

References

- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *CoRR*, abs/2204.05862.
- Amir Beck. 2017. *First-order methods in optimization*. SIAM.
- Stephen P Boyd and Lieven Vandenbergh. 2004. *Convex optimization*. Cambridge university press.
- Valentina Cacchiani, Manuel Iori, Alberto Locatelli, and Silvano Martello. 2022. [Knapsack problems — an overview of recent advances. part i: Single knapsack problems](#). *Computers & Operations Research*, 143:105692.
- Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. 2024. [How do large language models acquire factual knowledge during pretraining?](#) In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. [Accelerating large language model decoding with speculative sampling](#). *Preprint*, arXiv:2302.01318.
- C. Chow. 1970. [On optimum recognition error and reject tradeoff](#). *IEEE Transactions on Information Theory*, 16(1):41–46.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Haikang Deng and Colin Raffel. 2023. [Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11781–11791, Singapore. Association for Computational Linguistics.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. [Model alignment as prospect theoretic optimization](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12634–12651. PMLR.
- Gonçalo Faria and Noah A. Smith. 2025. [Sample, don’t search: Rethinking test-time alignment for language models](#). *Preprint*, arXiv:2504.03790.
- Yu Fei, Yasaman Razeghi, and Sameer Singh. 2025. [Nudging: Inference-time alignment of LLMs via guided decoding](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12702–12739, Vienna, Austria. Association for Computational Linguistics.
- Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. [Language model cascades: Token-level uncertainty and beyond](#). In *The Twelfth International Conference on Learning Representations*.
- Xuming Hu, Junzhe Chen, Xiaochuan Li, Yufei Guo, Lijie Wen, Philip S. Yu, and Zhijiang Guo. 2024. [Towards understanding factual knowledge of large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Wittawat Jitkrittum, Neha Gupta, Aditya K Menon, Harikrishna Narasimhan, Ankit Rawat, and Sanjiv Kumar. 2023. [When does confidence-based cascade deferral suffice?](#) In *Advances in Neural Information Processing Systems*, volume 36, pages 9891–9906. Curran Associates, Inc.
- Richard M. Karp. 1972. [Reducibility among Combinatorial Problems](#), pages 85–103. Springer US, Boston, MA.
- Maxim Khanov, Jirayu Burapachee, and Yixuan Li. 2024. [ARGS: Alignment as reward-guided search](#).

- In *The Twelfth International Conference on Learning Representations*.
- Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip H. S. Torr, Fahad Shahbaz Khan, and Salman Khan. 2025. [Llm post-training: A deep dive into reasoning large language models](#). *Preprint*, arXiv:2502.21321.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxu Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, and 4 others. 2025. [Tulu 3: Pushing frontiers in open language model post-training](#). In *Second Conference on Language Modeling*.
- Claude Lemaréchal. 2001. [Lagrangian relaxation](#). In Michael Jünger and Denis Naddef, editors, *Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions*, pages 112–156. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. [Fast inference from transformers via speculative decoding](#). *Preprint*, arXiv:2211.17192.
- Lei Li, Yankai Lin, Deli Chen, Shuhuai Ren, Peng Li, Jie Zhou, and Xu Sun. 2021. [CascadeBERT: Accelerating inference of pre-trained language models via calibrated complete models cascade](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 475–486, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. [Let’s verify step by step](#). *Preprint*, arXiv:2305.20050.
- Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024. [The unlocking spell on base LLMs: Rethinking alignment via in-context learning](#). In *The Twelfth International Conference on Learning Representations*.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [TruthfulQA: Measuring how models mimic human falsehoods](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.
- Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024. [Tuning language models by proxy](#). In *First Conference on Language Modeling*.
- Ximing Lu, Faeze Brahman, Peter West, Jaehun Jung, Khyathi Chandu, Abhilasha Ravichander, Prithviraj Ammanabrolu, Liwei Jiang, Sahana Ramnath, Nouha Dziri, Jillian Fisher, Bill Lin, Skyler Hallinan, Lianhui Qin, Xiang Ren, Sean Welleck, and Yejin Choi. 2023. [Inference-time policy adapters \(IPA\): Tailoring extreme-scale LMs without fine-tuning](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6863–6883, Singapore. Association for Computational Linguistics.
- Silvano Martello, David Pisinger, and Paolo Toth. 1999. [Dynamic programming and strong bounds for the 0-1 knapsack problem](#). *Management Science*, 45(3):414–424.
- Silvano Martello and Paolo Toth. 1990. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., USA.
- Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning. 2024. [An emulator for fine-tuning large language models using small language models](#). In *The Twelfth International Conference on Learning Representations*.
- Harikrishna Narasimhan, Wittawat Jitkittum, Ankit Singh Rawat, Seungyeon Kim, Neha Gupta, Aditya Krishna Menon, and Sanjiv Kumar. 2025. [Faster cascades via speculative decoding](#). In *The Thirteenth International Conference on Learning Representations*.
- Hitoshi Nishikawa, Kazuho Arita, Katsumi Tanaka, Tsutomu Hirao, Toshiro Makino, and Yoshihiro Matsuo. 2014. [Learning to generate coherent summary with discriminative hidden semi-Markov model](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1648–1659, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. [2 OLMo 2 furious](#). *Preprint*, arXiv:2501.00656.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. [Are NLP models really able to solve simple math word problems?](#) In *Proceedings of the 2021 Conference of the North American Chapter of the*

- Association for Computational Linguistics: *Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.
- David Pisinger. 1997. [A minimal algorithm for the 0-1 knapsack problem](#). *Operations Research*, 45(5):758–767.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- R. Renner and S. Wolf. 2004. [Smooth renyi entropy and applications](#). In *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.*, pages 233–.
- Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. [CommonsenseQA: A question answering challenge targeting commonsense knowledge](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jonathan Tonglet, Manon Reusens, Philipp Borchert, and Bart Baesens. 2023. [SEER : A knapsack approach to exemplar selection for in-context HybridQA](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13569–13583, Singapore. Association for Computational Linguistics.
- Sergey Troshin, Vlad Niculae, and Antske Fokkens. 2025. [On the low-rank parametrization of reward models for controlled language generation](#). *Transactions on Machine Learning Research*.
- Neeraj Varshney and Chitta Baral. 2022. [Model cascading: Towards jointly improving efficiency and accuracy of NLP systems](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11007–11021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Xin Wang, Yujia Luo, Daniel Crankshaw, Alexey Tumanov, Fisher Yu, and Joseph E. Gonzalez. 2018. [Idk cascades: Fast deep learning by learning not to overthink](#). *Preprint*, arXiv:1706.00885.
- Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. 2024. [From decoding to meta-generation: Inference-time algorithms for large language models](#). *Transactions on Machine Learning Research*. Survey Certification.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. [Lima: Less is more for alignment](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.
- Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. [Fine-tuning language models from human preferences](#). *Preprint*, arXiv:1909.08593.

A Proof of Lemma 2

Proof. The Lagrangian can be rewritten as follows:

$$\begin{aligned} L(\mathbf{d}, \mu) &= r(\mathbf{d}) + \mu \left(\sum_{i \in V} \mathbb{P}_i d_i - b \right) \\ &= \sum_{v \in V} d_v \mathbb{P}_v (\mu - \ell(p, v)) \\ &\quad + \sum_{v \in V} \mathbb{P}_v \ell(p, v) - \mu b, \end{aligned}$$

where only the first sum depends on $\mathbf{d}$. Therefore, computing the dual approximation can be rewritten as:

$$\hat{\mathbf{d}}^-(\mu) \in \arg \min_{\mathbf{d} \in \{0,1\}^{|V|}} \sum_{v \in V} d_v \mathbb{P}_v (\mu - \ell(p, v)),$$

where, for all $v \in V$, $\mathbb{P}_v \geq 0$ and $\ell(p, v) \geq 0$ by definition. Therefore, to minimize the sum in the objective, we must set $d_v = 1$ if $\mu - \ell(p, v) < 0$, and 0 otherwise. As such, we can write:

$$\begin{aligned} \left[\hat{\mathbf{d}}^-(\mu) \right]_v &= \llbracket \ell(p, v) > \mu \rrbracket \\ &\implies \\ \hat{\mathbf{d}}^-(\mu) &\in \arg \min_{\mathbf{d} \in \{0,1\}^{|V|}} L(\mathbf{d}, \mu). \end{aligned}$$

Note that if we use the condition:

$$\left[\hat{\mathbf{d}}^-(\mu) \right]_v = \llbracket \ell(p, v) \geq \mu \rrbracket,$$

although it does not change the objective value, it increases the probability of violating the budget constraint.

The bound on the true risk follows standard linear optimization properties (Lemar  chal, 2001). As $\hat{\mathbf{d}}$ satisfy the budget constraint, we have $\sum_{v \in [V]} \mathbb{P}_v \hat{d}_v - b \leq 0$, therefore:

$$r(\hat{\mathbf{d}}) \geq r(\hat{\mathbf{d}}) + \mu \left(\sum_{v \in V} \mathbb{P}_v \hat{d}_v - b \right).$$

We derive a lower bound on this term by minimizing on the deferral decision, which leads to the desired bound:

$$\begin{aligned} &\geq \min_{\mathbf{d} \in \{0,1\}^{|V|}} L(\mathbf{d}, \mu) \\ &= L(\hat{\mathbf{d}}^-(\mu), \mu). \end{aligned}$$

□

B Deferral Rules for Dual Approximation

The deferral rule for dual approximation has the following form:

$$\left[\hat{\mathbf{d}}^-(\mu) \right]_v = \llbracket \ell(\mathbf{p}, v) > \mu \rrbracket.$$

We derive the particular form of this rule for the losses that are summarized in Table 1.

Negative log-likelihood (logistic). Let ℓ be defined as:

$$\ell(\mathbf{p}, v) = -\log p_v.$$

Then we can rewrite the deferral condition as:

$$\begin{aligned} \ell(\mathbf{p}, v) &> \mu, \\ -\log p_v &> \mu, \\ p_v &< \exp(-\mu), \end{aligned}$$

which shows in this case the rule defer to $\mathbf{q}^*$ for low probability tokens.

Squared. Let ℓ be defined as:

$$\ell(\mathbf{p}, v) = (1 - p_v)^2$$

Then we can rewrite the deferral condition as:

$$\begin{aligned} \ell(\mathbf{p}, v) &> \mu, \\ (1 - p_v)^2 &> \mu, \\ 1 - p_v &> \sqrt{\mu}, \\ p_v &< 1 - \sqrt{\mu}. \end{aligned}$$

which shows in this case the rule defer to $\mathbf{q}^*$ for low probability tokens.

Note the similarity with the condition of the logistic loss. Given $\mu, \mu' \in \mathbb{R}_{\geq 0}$, and the two conditions:

$$p_v < \exp(-\mu) \tag{5}$$

$$\text{and } p_v < 1 - \sqrt{\mu'}. \tag{6}$$

If the following condition holds:

$$\mu = -\log \left(1 - \sqrt{\mu'} \right), \tag{7}$$

or equivalently:

$$\mu' = (1 - \exp(-\mu))^2, \tag{8}$$

then the two deferral rules are equivalent.

Perceptron. Let ℓ be defined as:

$$\ell(\mathbf{p}, v) = -p_v + \max_{w \in V} p_w$$

Then we can rewrite the deferral condition as:

$$\begin{aligned} \ell(\mathbf{p}, v) &> \mu, \\ -p_v + \max_{w \in V} p_w &> \mu, \\ p_v &< \max_{w \in V} p_w - \mu, \end{aligned}$$

which shows in this case the rule defer to $\mathbf{q}^*$ for tokens whose probability is lower than the max token probability minus a margin given by the dual variable μ .

Zero-one. Let ℓ be defined as:

$$\ell(\mathbf{p}, v) = \llbracket v \notin \arg \max_{w \in V} p_w \rrbracket$$

and since $\mu \in (0, 1)$ and $\ell(\mathbf{p}, v) \in \{0, 1\}$, $\left[\hat{\mathbf{d}}^-(\mu)\right]_v = 1$ if and only if $\ell(\mathbf{p}, v) = 1$ which is the case when $v \notin \arg \max_{w \in V} p_w$ or equivalently when $p_v < \max_{w \in V} p_w$.

C Proof of Lemma 3

Proof. First, note that L is a concave function since it is a pointwise minimum over a family of affine functions (Boyd and Vandenberghe, 2004, Sec. 3.2.3):

$$\begin{aligned} L(\mu) = \min_{\mathbf{d} \in \{0,1\}^{|V|}} \sum_{v \in V} \mathbb{P}_v \ell(\mathbf{p}, v) (1 - d_v) \\ + \mu \left(\sum_{v \in V} \mathbb{P}_v d_v - b \right). \end{aligned}$$

Per Lemma 2, for a given dual variable $\mu \in \mathbb{R}_{\geq 0}$, the optimal minimizer is given by $\hat{\mathbf{d}}^-(\mu)$ defined as:

$$\left[\hat{\mathbf{d}}^-(\mu)\right]_v = \llbracket \ell(\mathbf{p}, v) > \mu \rrbracket.$$

We observe that the value of L changes at breakpoints given by $\ell(\mathbf{p}, v)$, $\forall v \in V$. The subdifferential set at $\mu \in \mathbb{R}_+$ is given by (Beck, 2017, Thm. 3.50):

$$\partial L(\mu) = \left[\sum_{v: \ell(\mathbf{p}, v) > \mu} \mathbb{P}_v - b, \sum_{v: \ell(\mathbf{p}, v) \geq \mu} \mathbb{P}_v - b \right],$$

which is the convex hull of derivatives of functions attaining the minimum.

A sufficient condition for $\hat{\mu} \in \mathbb{R}_{\geq 0}$ to maximize L is that $0 \in \partial L(\hat{\mu})$. Assuming that the vocabulary is sorted in increasing loss order and knowing that $\hat{k}$ is defined as per condition (4), we have

$$\sum_{v=\hat{k}}^{|V|} \mathbb{P}_v - b > 0 \quad \text{and} \quad \sum_{v=\hat{k}+1}^{|V|} \mathbb{P}_v - b \leq 0.$$

Thus, $\hat{\mu} = \ell(\mathbf{p}, \hat{k})$. $\square$

D Primal Approximation with the 0-1 Loss

Let $\ell(\mathbf{p}, v)$ be the 0-1 loss function, that is:

$$\ell(\mathbf{p}, v) = \llbracket v \notin \arg \max_{w \in V} p_w \rrbracket.$$

As the optimal Lagrangian multiplier has the form:

$$\tilde{\mu} = \ell(\mathbf{p}, \tilde{k}), \quad (9)$$

where k is the critical index element, therefore we know that $\tilde{\mu} \in \{0, 1\}$.

If $\tilde{\mu} = 1$, then we have:

$$\left[\tilde{\mathbf{d}}^+\right]_v = \llbracket \ell(\mathbf{p}, v) > 1 \rrbracket = 0,$$

as, by definition, the 0-1 loss cannot be strictly greater than 1. In other word, we defer all the decision to p .

If $\tilde{\mu} = 0$, then we have:

$$\left[\tilde{\mathbf{d}}^+\right]_v = \llbracket \ell(\mathbf{p}, v) > 0 \rrbracket = \llbracket v \notin \arg \max_{w \in V} p_w \rrbracket,$$

that is, we keep from $\mathbf{p}$ only the mass of the most probable element.

We are left with finding when we have $\tilde{\mu} = 0$. By Eq. equation 9 and the definition of the 0-1 loss, this can only happen if $\tilde{k} = 1$. Remember that the critical index element $\tilde{k}$ is defined such that it satisfies the following conditions:

$$\sum_{v=\tilde{k}+1}^{|V|} p_v \leq b < \sum_{v=\tilde{k}}^{|V|} p_v.$$

Hence, to have $\tilde{k} = 1$, we must have $p_1 \geq 1 - b$. We end up with the rule:

$$\left[\tilde{\mathbf{d}}^+\right]_v = \llbracket p_1 \geq 1 - b \wedge v \notin \arg \max_w p_w \rrbracket.$$

E Proof of Lemma 4

Proof. We give the proof for the primal feasible solution $\hat{\mathbf{d}}^+$, and a similar reasoning holds for the approximate solution $\tilde{\mathbf{d}}^+$.

According to the hypothesis on ℓ_1 and ℓ_2 , sorting tokens in the increasing order of $\ell_1(\mathbf{p}, v)$ or $\ell_2(\mathbf{p}, v)$ result in the same ordering. Therefore, they both lead to the same critical element index $\hat{k}$, which depends only on $\mathbb{P}$ and the ordering (and not the particular loss values).

Thus, the optimal Lagrangian multipliers given by the two losses are $\hat{\mu}_1 = \ell_1(\mathbf{p}, \hat{k})$ and $\hat{\mu}_2 = \ell_2(\mathbf{p}, \hat{k})$. Since the losses have the same order, it is easy to see that for any $v \in V$, we have

$$\llbracket \ell_1(\mathbf{p}, v) > \hat{\mu}_1 \rrbracket = \llbracket \ell_2(\mathbf{p}, v) > \hat{\mu}_2 \rrbracket,$$

and therefore both losses produces the same primal feasible solution $\hat{\mathbf{d}}^+$. $\square$

F Proof of Lemma 5

Proof. Let us consider the quantity

$$|r(\hat{\mathbf{d}}^+) - r(\tilde{\mathbf{d}}^+)|.$$

Assuming that vocabulary is sorted in increasing loss order, we can see that

$$\begin{aligned} |r(\hat{\mathbf{d}}^+) - r(\tilde{\mathbf{d}}^+)| &= |r(\hat{\mathbf{d}}^-(\hat{\mu})) - r(\hat{\mathbf{d}}^-(\tilde{\mu}))| \\ &\leq \sum_v \mathbb{P}_v \ell(\mathbf{p}, v) \left| [\hat{\mathbf{d}}^-(\hat{\mu})]_v - [\hat{\mathbf{d}}^-(\tilde{\mu})]_v \right| \end{aligned}$$

Since $\hat{\mathbf{d}}^-(\hat{\mu})$ and $\hat{\mathbf{d}}^-(\tilde{\mu})$ agree on all positions except $\{\min(\hat{k}, \tilde{k}) + 1, \dots, \max(\hat{k}, \tilde{k})\}$

$$= \sum_{v=\min(\hat{k}, \tilde{k})+1}^{\max(\hat{k}, \tilde{k})} \mathbb{P}_v \ell(\mathbf{p}, v).$$

Let us define $l = \min(\hat{k}, \tilde{k}) + 1$ and $u = \max(\hat{k}, \tilde{k})$, remember that the vocabulary is sorted in increasing loss order:

$$\leq \ell(\mathbf{p}, u) \sum_{v=l}^u \mathbb{P}_v.$$

Let us consider the second factor in the left side of the inequality

$$\begin{aligned} \sum_{v=l}^u \mathbb{P}_v &= \frac{1}{2} \sum_{v=1}^{|V|} (\mathbb{P}_v - p_v) + \frac{1}{2} \sum_{v=1}^{|V|} p_v \\ &\quad + \frac{1}{2} \sum_{v=l}^u \mathbb{P}_v - \frac{1}{2} \sum_{v=1: v \notin [u, l]}^{|V|} \mathbb{P}_v \\ &\leq D_{\text{TV}}(\mathbb{P}, p) \\ &\quad + \frac{1}{2} \left(1 + \sum_{v=l}^u \mathbb{P}_v - \sum_{v=1: v \notin \{u, \dots, l\}}^{|V|} \mathbb{P}_v \right) \\ &= D_{\text{TV}}(\mathbb{P}, p) + \sum_{v=l}^u \mathbb{P}_v, \end{aligned}$$

where D_{TV} is the total-variation distance. Going back to our initial bound

$$\begin{aligned} |r(\hat{\mathbf{d}}^+) - r(\tilde{\mathbf{d}}^+)| &\leq \\ &\ell(\mathbf{p}, u) \left(D_{\text{TV}}(\mathbb{P}, p) + \sum_{v=l}^u \mathbb{P}_v \right). \end{aligned}$$

As for the second part of the Lemma, we have

$$\begin{aligned} L(\hat{\mu}) &= L(\hat{\mathbf{d}}^+, \hat{\mu}) \\ &= r(\hat{\mathbf{d}}^+) + \hat{\mu} \left(\sum_{i \in V} \mathbb{P}_i \hat{d}_i^+ - b \right) \\ &\leq r(\hat{\mathbf{d}}). \end{aligned}$$

From the last inequality, we have

$$\begin{aligned} r(\hat{\mathbf{d}}^+) - r(\hat{\mathbf{d}}) &\leq \hat{\mu} \left(b - \sum_{i \in V} \mathbb{P}_i \hat{d}_i^+ \right) \\ &\leq \ell(\mathbf{p}, \hat{k}) \mathbb{P}_{\hat{k}}. \end{aligned}$$

Since $b < \sum_{i=\hat{k}}^{|V|} \mathbb{P}_i \implies b - \sum_{i=\hat{k}+1}^{|V|} \mathbb{P}_i < \mathbb{P}_{\hat{k}}$. $\square$

G Primal Approximation Algorithm

Algorithm 1 shows a simple sorting-based procedure for calculating the primal approximation deferral rule.

H Proof of lemma 6

Proof. For the first part of the lemma, notice that if $\max_w p_w < \lambda$ then $\forall v \in V, p_v < \lambda$ and $\mathbf{d} = \mathbf{1}$ giving $\phi = \pi^{<\lambda} = \mathbf{q}^*$ (trivial case). On the other hand, if $\max_w p_w \geq \lambda$ then $\phi = \mathbf{p}$ and we have:

$$\begin{aligned}
& \frac{1}{2} \sum_i (1 - d_i) |p_i + q_i^* \alpha - q_i^*| \\
& \leq \frac{1}{2} \sum_i (1 - d_i) |p_i - q_i^*| \\
& \quad + \frac{1}{2} \alpha \sum_i (1 - d_i) q_i^* \\
& \leq \frac{1}{2} \sum_i |p_i - q_i^*| - \frac{1}{2} \sum_i d_i |p_i - q_i^*| \\
& \quad + \frac{1}{2} \alpha - \frac{1}{2} \alpha \sum_i d_i q_i^* \\
& = D_{\text{TV}}(\mathbf{p}, \mathbf{q}^*) - \frac{1}{2} \sum_i d_i |p_i - q_i^*| \\
& \quad + \frac{1}{2} \alpha - \frac{1}{2} \alpha \sum_i d_i q_i^*.
\end{aligned}$$

Using this result we derive

$$\begin{aligned}
D_{\text{TV}}(\pi^{<\lambda}, \mathbf{q}^*) &= \frac{1}{2} \sum_i d_i |q_i^* - q_i^* \alpha| \\
&+ (1 - d_i) |p_i + q_i^* \alpha - q_i^*| \\
&\leq D_{\text{TV}}(\mathbf{p}, \mathbf{q}^*) - \frac{1}{2} \sum_i d_i |p_i - q_i^*| \\
&\quad + \frac{1}{2} \alpha - \frac{1}{2} \alpha \sum_i d_i q_i^* \\
&\quad + \frac{1}{2} \sum_i d_i q_i^* - \frac{1}{2} \alpha \sum_i d_i q_i^* \\
&= D_{\text{TV}}(\mathbf{p}, \mathbf{q}^*) - \frac{1}{2} \sum_i d_i |p_i - q_i^*| \\
&\quad + \frac{1}{2} \alpha - \alpha \sum_i d_i q_i^* + \frac{1}{2} \sum_i d_i q_i^*
\end{aligned}$$

$$\begin{aligned}
&= D_{\text{TV}}(\mathbf{p}, \mathbf{q}^*) - \frac{1}{2} \sum_i d_i |p_i - q_i^*| \\
&\quad - \alpha \sum_i d_i q_i^* + \frac{1}{2} \sum_i d_i |p_i + q_i^*| \\
&\leq -\alpha \sum_i q_i^* d_i + \sum_i d_i p_i + D_{\text{TV}}(\mathbf{p}, \mathbf{q}^*) \\
&= \alpha(1 - \sum_i q_i^* d_i) + D_{\text{TV}}(\mathbf{p}, \mathbf{q}^*),
\end{aligned}$$

which shows that $D_{\text{TV}}(\pi^{<\lambda}, \mathbf{q}^*) \leq \alpha(1 - \sum_i q_i^* d_i) + D_{\text{TV}}(\mathbf{p}, \mathbf{q}^*)$.

For the second part of the lemma, remember $D_{\text{TV}}(\mathbf{p}, \mathbf{q}^*) = 1 - \sum_v \min(p_v, q_v^*)$.

We have $D_{\text{TV}}(\pi^{<\lambda}, \mathbf{q}^*) \leq D_{\text{TV}}(\mathbf{p}, \mathbf{q}^*)$ is equivalent to

$$\sum_v \min(\pi_v^{<\lambda}, q_v^*) \geq \sum_v \min(p_v, q_v^*)$$

which is equivalent to

$$\begin{aligned}
& \sum_{v \in \mathcal{A}} (\min(p_v + \alpha q_v^*, q_v^*) - \min(p_v, q_v^*)) \\
& + \sum_{v \in \mathcal{B}} (\alpha q_v^* - \min(p_v, q_v^*)) \geq 0, \quad (10)
\end{aligned}$$

where $\mathcal{A} = \{v \in V : p_v \geq \lambda\}$ and $\mathcal{B} = \{v \in V : p_v < \lambda\}$. Suppose that the conditions of the lemma hold, then:

- For $v \in \mathcal{A}$, if $\min(p_v, q_v^*) = q_v^*$ then $\min(p_v + \alpha q_v^*, q_v^*) = q_v^*$. Otherwise if $\min(p_v, q_v^*) = p_v$ then $\min(p_v + \alpha q_v^*, q_v^*) \geq p_v$. Hence $\min(p_v + \alpha q_v^*, q_v^*) - \min(p_v, q_v^*) \geq 0$.
- For $v \in \mathcal{B}$: $p_v \leq \alpha q_v^*$ implies $\min(p_v, q_v^*) = p_v$ and $\min(\pi_v^{<\lambda}, q_v^*) = \min(\alpha q_v^*, q_v^*) = \alpha q_v^*$. Thus

$$\alpha q_v^* - \min(p_v, q_v^*) = \alpha q_v^* - p_v \geq 0.$$

Summing the nonnegative coordinate differences over $\mathcal{A}$ and $\mathcal{B}$ gives the left hand side of (10) is ≥ 0 . Hence $D_{\text{TV}}(\pi^{<\lambda}, \mathbf{q}^*) \leq D_{\text{TV}}(\mathbf{p}, \mathbf{q}^*)$. $\square$

I Dataset Examples

Following previous work, we rely on regular expressions to extract the final response form the LLM generation. In the case of GSM8K and SVAMP, we extract the last number in the model's response. In the case of MATH500, we extract the

Algorithm 1: Returns the deferral rule using primal approximation.

Data: Budget $b \in [0, 1]$, distribution p .

Result: Deferral rule $d \in \{0, 1\}^V$.

begin

$v \leftarrow -p \log p$ /* Values, p is a plug-in estimator for $\mathbb{P}$ */

$w \leftarrow p$ /* Weights, p is a plug-in estimator for $\mathbb{P}$ */

$c \leftarrow v/w$ which is equal to $-\log p$ /* Densities */

$s \leftarrow \text{argsort_descending}(c)$

$d \leftarrow 0$

$b' \leftarrow 0$

$i \leftarrow 1$

while $i \leq V$ and $b' + w_{s_i} \leq b$ **do**

$d_{s_i} \leftarrow 1$

$b' \leftarrow b' + w_{s_i}$

end

return d

end

last $\boxed{}$ number in the model’s response. In the multiple-choice question tasks, we extract the last capital letter from A to J in the response.

The models are evaluated on the question prompts alone, with no additional few-shot examples. For each task, we use the extracted answer’s accuracy to evaluate the model’s performance. We also evaluate the generation throughput as the average number of tokens generated by the model each second.

The following system prompt was used in all tasks:

Answer the question by walking through the reasoning step by step.

MATH500. Here is an example of a question/answer from MATH500 dataset:

```
{
  "role": "user",
  "content": "Convert the point $(0,3)$ in rectangular coordinates to polar coordinates. Enter your answer in the form $(r,\theta)$, where $r > 0$ and $0 \le \theta < 2\pi$. Present the answer in LaTeX format: \boxed{Your answer}"
},
{
  "role": "assistant",
  "content": "We have that $r = \sqrt{0^2 + 3^2} = 3$. Also, if we draw the line connecting the origin and $(0,3)$, this line makes an angle of $\frac{\pi}{2}$ with the positive $x$-axis. \n\n[asy]\nunitsize(0.8 cm);\n\ndraw((-0.5,0)--(3.5,0));\ndraw((0,-0.5)--(0,3.5)); \ndraw(arc((0,0),3,0,90),red,Arrow(6));\n\ndot((0,3), red);\nlabel("$$(0,3)$"
```

```
", (0,3), W);\ndot((3,0), red); \n[/asy]
\n\nTherefore, the polar coordinates are
$\boxed{\left( 3, \frac{\pi}{2} \right)}$."
}
```

The following instruction is appended at the end of the user’s query :

\n\nPresent the answer in LaTeX format:
\boxed{{Your answer}}

GSM8K. Here is an example of a question/answer from GSM8K dataset:

```
{
  "role": "user",
  "content": "Janet\u2019s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?"
},
{
  "role": "assistant",
  "content": "How many eggs does Janet sell? ** Janet sells 16 - 3 - 4 = <<16-3-4=9>>9 duck eggs a day. \nHow much does Janet make at the farmers' market? ** She makes 9 * 2 = <<9*2=18>>18 every day at the farmer\u2019s market. \n#### 18"
}
```

ARC. Here is an example of a question/answer from ARC dataset:

```
{
  "role": "user",
  "content": "Choose the correct answer to the following multiple-choice question. \n\nQuestion: An astronomer observes that a planet rotates faster after a meteorite impact. Which is the most likely effect of this increase
```



```

in rotation?\n\nA). Planetary density will
decrease.\nB). Planetary years will become longer.
\nC). Planetary days will become shorter.
\nD). Planetary gravity will become stronger.
\n\nProvide your reasoning about the answer and
finish your answer with the letter corresponding
to the correct option (e.g., A, B, C, or D).\n\n"
},
{
  "role": "assistant",
  "content": "\nAnswer: C\n\n"
}

```

The following prefix is prepended to the user’s query

Choose the correct answer to the following multiple-choice question.\n\n

The following suffix is appended to the user’s query

\nProvide your reasoning about the answer and finish your answer with the letter corresponding to the correct option (e.g., A, B, C, or D).\n\n

J Further Analysis

Conditional distribution entropy. The Shannon entropy is defined as follows:

$$H[\mathbf{p}] = - \sum_{v \in V} p_v \log p_v.$$

Entropy is a measure of uncertainty, *i.e.* high entropy means high uncertainty. In the following we consider entropies of conditional distributions, where we condition the distribution on sampling element that are deferred in $\pi^{<\lambda}$, for different values of λ , which we write as follows:

$$\begin{aligned} & [\langle \mathbf{p}, \mathbf{d} \rangle^{-1} \mathbf{p} \odot \mathbf{d}]_v \\ &= p \left(\mathbf{x}_t = v \left| \begin{array}{l} \mathbf{x}_t \in \{v \in V | d_v = 1\}, \\ \mathbf{x}_{1:t-1} = \mathbf{x}_{1:t-1} \end{array} \right. \right), \end{aligned}$$

where $\odot$ is the element-wise multiplication, $\langle \cdot, \cdot \rangle$ is the dot product. We define a similar conditional distribution for $\mathbf{p}^*$.

When the following inequality holds:

$$H[\langle \mathbf{p}, \mathbf{d} \rangle^{-1} \mathbf{p} \odot \mathbf{d}] \geq H[\langle \mathbf{p}^*, \mathbf{d} \rangle^{-1} \mathbf{p}^* \odot \mathbf{d}],$$

the low probability elements of $\mathbf{p}$ has a higher uncertainty than the same elements of $\mathbf{p}^*$. To check if this is true empirically, we plot on Figure 3 the difference of entropy between the conditional distributions based on $\mathbf{p}$ and $\mathbf{p}^*$, for different values of λ . We can see the the entropy of the conditional distribution based on $\mathbf{p}$ is much higher than that of its aligned version $\mathbf{p}^*$. This motivates the need to mask those parts of $\mathbf{p}$ and replace them with $\mathbf{q}^*$.

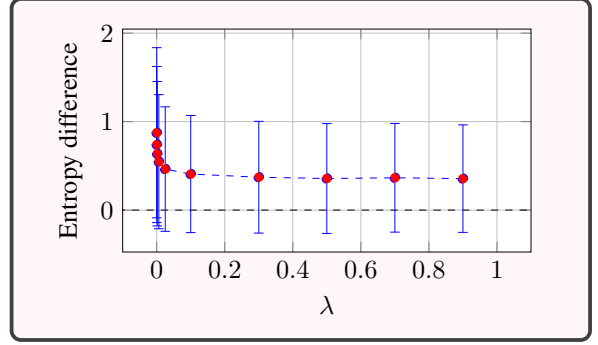


Figure 3: Entropy different $H[\langle \mathbf{p}, \mathbf{d} \rangle^{-1} \mathbf{p} \odot \mathbf{d}] - H[\langle \mathbf{p}^*, \mathbf{d} \rangle^{-1} \mathbf{p}^* \odot \mathbf{d}]$ for different values of λ . We report mean differences evaluated on OLMo-2 1B and 13B on MATH500.

$\ \mathbf{1} - \hat{\mathbf{d}}^-\ _1$	= 0	= 1	= 2	= 3
	9.5%	88.6%	1.9%	0%
$\ \mathbf{1} - \tilde{\mathbf{d}}_{(b=0.9)}^+\ _1$	= 0	= 1	= 2	= 3
	0.0%	99.9%	0.1%	0%
$\ \mathbf{1} - \tilde{\mathbf{d}}_{(b=0.6)}^+\ _1$	= 0	= 1	= 2	= 3
	0.0%	90.7%	6.3%	1.5%

Table 3: Proportions of the number of selected token-probabilities from $\mathbf{p}$ using a threshold of $\lambda = 0.4$ for $\hat{\mathbf{d}}^-$ and a budget of $b \in \{0.6, 0.9\}$ for $\tilde{\mathbf{d}}^+$. Evaluated on OLMo-2 1B and 13b on MATH500.

p-selection proportion. Table 3 shows the proportions of tokens whose mass from $\mathbf{p}$ is kept in the mixture, *i.e.* the number of non-deferred tokens. We show statistics for $\pi^{<\lambda}$ with $\lambda = 0.4$, and for primal approximation with a budget of $\{0.6, 0.9\}$. We can see that those rules select probabilities from $\mathbf{p}$ according to their contribution, measured either with a threshold or a budget, and this contribution can be split over 1 or more tokens. For example, decreasing the budget b of the primal approximation will spread the contribution of $\mathbf{p}$ across more than one token.

Budget selection. Setting a high budget does not mean that it will necessarily be fully met, as the remaining unfilled budget depends on the size of the critical element, *i.e.* the larger the critical element is, the smaller the actually filled budget can be. An example is shown in Figure 4.

	GMS8K	MATH	SVAMP	ARC	CSQA	TQA	Avg.	GMS8K	MATH	SVAMP	ARC	CSQA	TQA	Avg.
	OLMo 2 (1B and 13B)							Qwen 3 (1.7B and 14B)						
Building blocks														
p - AR	26.6	26.4	26.3	27.1	27.1	26.9	26.7	26.8	26.5	26.1	26.4	26.6	26.5	26.4
q^* - AR	65.8	65.7	64.9	66.5	66.5	65.9	65.8	39.4	39.0	39.1	39.4	38.8	39.0	39.1
Nudging (Fei et al., 2025)														
$\lambda = 0.4$ - AR	26.6	26.8	26.7	26.0	24.5	25.5	26.0	26.2	26.1	26.2	26.1	25.9	26.1	26.0
$\lambda = 0.4, \gamma = 3$	27.2	27.1	26.8	25.6	25.3	25.7	26.2	21.5	21.1	21.5	19.7	18.8	19.4	20.3
$\lambda = 0.4, \gamma = 5$	29.1	28.6	28.5	25.4	24.8	26.2	27.0	22.7	22.4	22.3	19.2	18.0	18.8	20.5
$\lambda = 0.4, \gamma = 7$	28.5	27.6	26.9	23.9	22.8	23.8	25.5	22.0	21.5	21.5	17.4	15.9	17.0	19.2
Our work: dual approximation														
$\hookrightarrow \pi^{<\lambda}$ with $\lambda = f^{-1}(\mu)$														
$\lambda = 0.4$ - AR	19.4	19.3	19.4	19.6	19.3	19.5	19.4	16.1	16.0	16.0	16.1	16.1	15.9	16.0
$\lambda = 0.4, \gamma = 3$	28.8	28.6	28.0	27.7	28.2	27.8	28.1	22.1	22.0	22.2	20.6	19.8	20.2	21.1
$\lambda = 0.4, \gamma = 5$	31.9	31.2	30.2	28.9	30.3	29.3	30.3	23.6	23.0	23.2	20.2	19.6	20.3	21.6
$\lambda = 0.4, \gamma = 7$	31.7	31.6	30.3	27.4	27.7	28.4	29.5	23.1	22.6	22.9	19.1	17.8	18.8	20.7
Our work: primal approximation														
$b = 0.9$ - AR	19.3	19.1	19.4	19.3	19.2	19.4	19.2	15.8	16.0	15.8	15.9	15.9	16.1	15.9
$b = 0.9, \gamma = 3$	28.7	28.1	28.1	26.9	26.8	26.8	27.5	22.0	21.7	22.2	20.0	19.8	20.0	21.9
$b = 0.9, \gamma = 5$	30.6	30.3	29.8	27.8	25.5	26.6	28.4	23.5	22.9	23.1	20.2	19.1	19.9	21.4
$b = 0.9, \gamma = 7$	30.1	29.8	28.7	24.6	21.6	22.4	26.2	23.4	22.4	22.5	18.2	17.4	18.2	20.3

Table 4: Model’s throughput in tokens per second. “AR” stands for auto-regressive, other lines correspond to speculative decoding with a draft size of γ . Best results on each column are in bold, second best are underlined.

	GMS8K	MATH	SVAMP	ARC	CSQA	TQA	Avg.	GMS8K	MATH	SVAMP	ARC	CSQA	TQA	Avg.
	OLMo 2 (1B and 13B)							Qwen 3 (1.7B and 14B)						
Nudging (Fei et al., 2025)														
$\lambda = 0.4, \gamma = 3$	67.6	69.2	65.9	59.0	56.5	59.3	62.9	78.4	78.1	<u>75.3</u>	65.0	59.5	64.0	70.0
$\lambda = 0.4, \gamma = 5$	56.5	57.5	53.3	46.7	44.0	48.2	51.0	69.5	69.0	65.9	53.2	47.5	52.2	59.5
$\lambda = 0.4, \gamma = 7$	48.0	48.6	43.1	38.6	36.6	38.5	42.2	61.7	60.9	59.2	44.4	38.6	43.6	51.4
Our work: dual approximation														
$\hookrightarrow \pi^{<\lambda}$ with $\lambda = f^{-1}(\mu)$														
$\lambda = 0.4, \gamma = 3$	73.7	75.6	71.4	67.7	69.3	68.8	71.0	<u>81.6</u>	81.3	79.8	69.9	66.2	70.2	74.8
$\lambda = 0.4, \gamma = 5$	64.0	65.9	59.3	55.5	60.0	57.3	60.3	73.9	72.5	71.6	59.1	54.7	59.5	65.2
$\lambda = 0.4, \gamma = 7$	55.4	57.9	52.3	47.6	48.3	49.8	51.8	67.0	65.3	64.0	50.1	46.0	50.3	57.1
Our work: primal approximation														
$b = 0.9, \gamma = 3$	<u>73.5</u>	<u>74.6</u>	<u>68.8</u>	<u>65.5</u>	<u>64.9</u>	<u>64.4</u>	<u>68.6</u>	81.7	<u>80.9</u>	79.8	<u>69.8</u>	<u>66.0</u>	<u>69.4</u>	<u>74.6</u>
$b = 0.9, \gamma = 5$	62.0	63.1	57.2	53.0	46.7	49.9	55.3	74.0	72.2	71.3	58.1	53.1	57.6	64.3
$b = 0.9, \gamma = 7$	53.5	54.5	48.0	41.2	33.8	36.4	44.5	66.8	64.3	64.2	48.5	44.1	48.1	56.0

Table 5: Speculative generation empirical acceptance rate. Best results on each column are in bold, second best are underlined.

K Generation Speed Results

Table 4 shows the throughput for all the tested models, both for autoregressive generation and speculative decoding. Table 5 shows the exact acceptance rates.

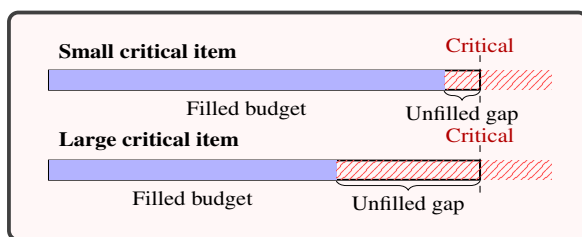


Figure 4: Illustration of the effect of the critical element size on the filled budget; a bigger critical element can leave a bigger unfilled gap.