SPECS: Faster Test-Time Scaling through Speculative Drafts

Mert Cemri * [1] **Nived Rajaraman** * [1] **Rishabh Tiwari** * [1] **Xiaoxuan Liu** [1] **Kurt Keutzer** [1]
Ion Stoica [1] **Kannan Ramchandran** [1] **Ahmad Beirami** [2] **Ziteng Sun** [3]

Abstract

Scaling test-time compute has driven the recent advances in the reasoning capabilities of large language models (LLMs), typically by allocating additional computation for more thorough exploration. However, increased compute often comes at the expense of higher user-facing latency, directly impacting user experience. Current test-time scaling methods primarily optimize for accuracy based on total compute resources (FLOPS), often overlooking latency constraints. To address this gap, we propose SPECS, a latency-aware test-time scaling method inspired by speculative decoding. SPECS uses a smaller, faster model to generate candidate sequences efficiently, and evaluates these candidates using signals from both a larger target model and a dedicated reward model. We introduce new integration strategies, including reward-guided soft verification and a reward-based deferral mechanism. Empirical results on MATH500, AMC23 and OlympiadBench datasets show that SPECS matches or surpasses beam search accuracy while reducing latency by up to ∼19.1%. Our theoretical analysis shows that our algorithm converges to the solution of a KL-regularized reinforcement learning objective with increasing beam width.

1. Introduction

Modern LLMs excel at multi-step reasoning, and scaling test-time compute has played a major role in achieving these reasoning capabilities by letting these models tackle harder problems with extra "thinking" compute resources (Cobbe et al., 2021; Wei et al., 2022; Nakano et al., 2022; Beirami et al., 2024; Brown et al., 2024; Beeching et al., 2024; Qiu et al., 2024; OpenAI, 2024; DeepSeek-AI, 2025).

To date, test-time scaling methods have primarily optimized performance based on total compute, demonstrating improved downstream task performance with increased FLOPS or generated tokens (OpenAI, 2024; Snell et al., 2025). However, user experience often depends more directly on serving *latency*, especially in low-throughput scenarios like personalized interactions or applications serving few users (Patel et al., 2024; Agrawal et al., 2024).

Consequently, we address the critical research question:

Can we design efficient test-time scaling methods to improve the latency-utility tradeoff?

Latency in autoregressive sampling from transformer-based LLMs is often limited by memory loading rather than total FLOPS (Tiwari et al., 2025; Yuan et al., 2024). This has led to the development of latency-focused techniques such as speculative decoding (Leviathan et al., 2022; Chen et al., 2023). Speculative decoding uses a smaller, faster draft model to propose candidate tokens, which are verified in parallel by a larger target model, reducing sequential bottlenecks and response times. However, this reduction in latency typically incurs higher total computation.

Our investigation on test-time scaling builds on the family of beam search or tree search (Mudgal et al., 2024; Beeching et al., 2024; Sun et al., 2024; Qiu et al., 2024; Snell et al., 2025) algorithms, which optimize performance by generating multiple candidate trajectories and selecting the best ones based on reward signals. Beam search's structured exploration makes it particularly effective at balancing accuracy and computational effort. Importantly, this structured exploration also enables us to extend speculative decoding beyond simple distribution matching.

For example, with the reward information, we could keep good draft trajectories with high rewards, which might have been discarded in naive speculative decoding if they do not appear like samples that the large model would have generated. Moreover, the multi-step reasoning process also offers us the flexibility to choose the reasoning model to use for each step.

Motivated by this, we present SPECS (SPECulative drafting for faster test-time Scaling), which directly optimizes

*Equal contribution [1]UC Berkeley [2]Google DeepMind [3]Google Research. Correspondence to:
Mert Cemri <cemri@berkeley.edu>,
Nived Rajaraman <nived.rajaraman@berkeley.edu>,
Rishabh Tiwari <rishabhtiwari@berkeley.edu>,
Ahmad Beirami <ahmad.beirami@gmail.com>,
Ziteng Sun <zitengsun@google.com>.

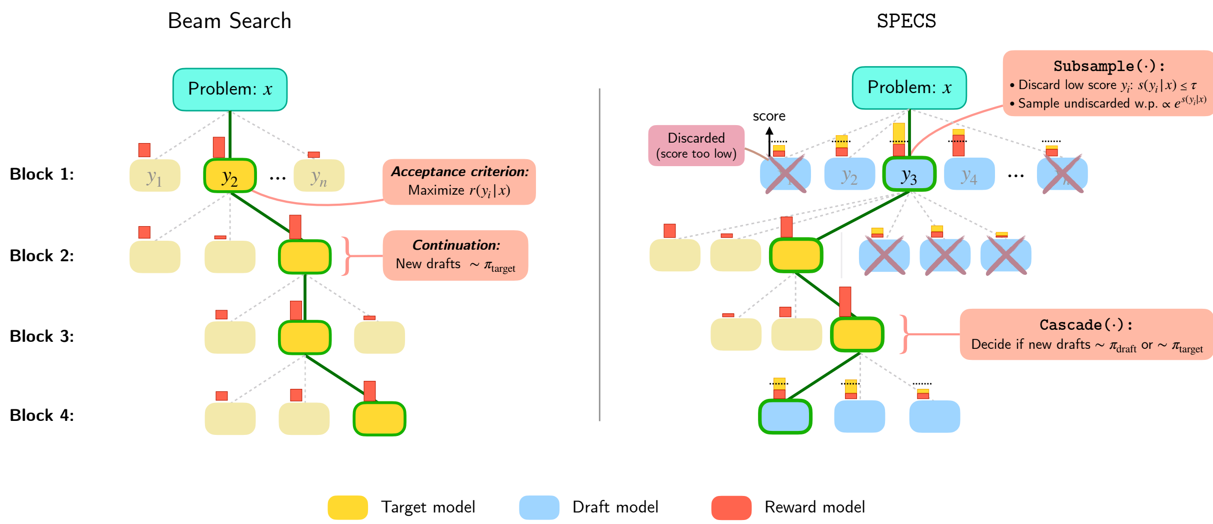

Figure 1. Visualization of Beam Search vs SPECS. In beam search the trajectories are generated by the target model (p) and scored using a PRM (r). In contrast, in SPECS the beams are primarily generated using a smaller and faster draft model (q) and scored by a combination of target and PRM model resulting in better latency-performance tradeoff.

the latency-utility tradeoff. SPECS utilizes a fast, smaller model to propose candidate reasoning steps, and uses a larger, more capable model, and a reward model to select from the candidates. We show that SPECS exhibits nice theoretical properties, and also demonstrate its strong empirical performance. In particular, our contributions can be summarized as follows:

- We propose a novel test-time scaling algorithm, SPECS (Algorithm 1), which utilizes a draft model to produce speculative drafts in steps, which are scored and selected by a large model, and a process reward model (PRM) to reach favorable latency-utility tradeoff.
- Theoretically, we analyze inference-time optimization of the KL-regularized reward maximization objective using a draft model alongside target/reward models, proving that our soft verification approach converges gracefully to the optimal solution as beam-width increases.
- Empirically, we evaluate on MATH-500, AMC23 and OlympiadBench datasets, demonstrating that SPECS achieves up to 19.1% reduction in latency while achieving on-par or higher accuracy compared to beam search and other baselines, using Qwen-1.5B-Instruct and Qwen-7B-Instruct models with Qwen-7B-Math-PRM.

2. Speculative drafting with reward-guided soft verification

In this section, we present our main algorithm SPECS. The algorithm follows an overall draft-and-select framework. The response is generated in blocks iteratively. In each iteration, the algorithm uses a fast, draft model to propose candidate blocks (reasoning steps) through autoregressive sampling. The candidate blocks are then scored by both a larger, more capable model, and a reward model, which are used to select from the candidates. The selected candidate block is appended to the existing response to start the next iteration.

To specifically optimize for utility-latency tradeoff, we develop a soft verification procedure optimized of reward-guided draft selection, summarized in the SUBSAMPLE subroutine. We further combine our approach with ideas from cascades (Dohan et al., 2022; Jitkrittum et al., 2023; Yue et al., 2024; Narasimhan et al., 2024) and construct a reward-aware deferral rule to adaptively decide which model to generate candidate steps. This aims to ensure that the larger model generates candidates for the harder steps of a problem, while the smaller draft model is used for the easier steps, summarized in the CASCADE subroutine.

Next, we break down the details of our algorithm into three parts. We first discuss the meta-algorithm in Section 2.1. Then we move to the subroutines in Section 2.2 and Section 2.3.

2.1. SPECS meta-algorithm

We detail our SPECS meta algorithm in Algorithm 1, and present a demonstration of the algorithm in Figure 1.

At a high level, in each iteration the algorithm generates n draft blocks $\{Y_i\}_{i=1}^n$ from a draft model π_{draft}, and uses the SUBSAMPLE$(\cdot)$ subroutine to determine one to select, or whether to reject all of them. This process utilizes a "score" computed using the draft, target and reward models. If a

Algorithm 1 SPECS meta algorithm

Input: Prompt X, Beam width n, Block size γ, draft model π_{draft}, target model π_{target}, sampling and cascade routines: SUBSAMPLE$(\cdot)$, CASCADE$(\cdot)$
Initialize: Current partial response, $\mathsf{tr} = X$
repeat
Step I. Sample n draft blocks from the draft model π_{draft},

$$\mathbf{Y}_{\text{draft}} = \{Y_i\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} \pi_{\text{draft}}(\cdot|\mathsf{tr})$$

Step II. Run SUBSAMPLE$(\cdot)$ for draft selection:
// SUBSAMPLE$(\cdot)$ selects a draft using $\pi_{\text{draft}}, \pi_{\text{target}}$
// and r according to the scores in Equation (1). If all
// scores are low, all drafts are rejected and return $\perp$.

if SUBSAMPLE$(\mathbf{Y}_{\text{draft}}, \beta_0{=}\frac{\beta}{2}, \mathsf{R}{=}1|\mathsf{tr}) \neq \perp$ **then**
$Y_\star = $ SUBSAMPLE$(\mathbf{Y}_{\text{draft}}, \beta_0{=}\frac{\beta}{2}, \mathsf{R}{=}1|\mathsf{tr})$
$\mathsf{tr} \leftarrow (\mathsf{tr}, Y_\star)$
Return to *Step I*.
else
// If all drafts rejected, switch to using π_{target}.
Step III. Generate n drafts from the target model,

$$\mathbf{Y}_{\text{target}} = \{Y_i'\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} \pi_{\text{target}}(\cdot|\mathsf{tr})$$

Step IV. Select a draft using SUBSAMPLE$(\cdot, \mathsf{R} = 0)$
// $\mathsf{R} = 0$ disallows rejecting all drafts.

$$Y_\star' \leftarrow \text{SUBSAMPLE}(\mathbf{Y}_{\text{target}}, \beta_0{=}\beta, \mathsf{R}{=}0|\mathsf{tr}),$$

Update $\mathsf{tr} \leftarrow (\mathsf{tr}, Y_\star')$.
Step V. Run CASCADE$(Y_1', \cdots, Y_n')$:
Decide whether to go to *Step I* or *Step III*.
// Deferral rule CASCADE$(\cdot)$ uses reward signal
// r to determine whether to generate blocks from
// π_{draft} or π_{target} in the next iteration.
until Target sequence length is achieved
Return: tr

Algorithm 2 SUBSAMPLE subroutine

Input: Context tr, Draft blocks $\{Y_i\}_{i=1}^{n}$, Target model π_{target}, Process reward model r, Draft generating model $\pi_{\text{base}} \in \{\pi_{\text{draft}}, \pi_{\text{target}}\}$, Rejection mode $\mathsf{R} \in \{0, 1\}$, Inverse temperature β_0
Hyperparameters: Threshold τ
Initialize: $L \leftarrow [n]$
for $i = 1$ **to** n **do**
Define $S_i = \log\left(\frac{\pi_{\text{target}}(Y_i|\,\text{tr})}{\pi_{\text{base}}(Y_i|\,\text{tr})}\right) + \beta_0 r(Y_i|\,\text{tr})$
// "Score" of each draft block i
if $\mathsf{R} = 1$ **and** $S_i \leq \tau$ **then**
$L \leftarrow L \setminus \{i\}$.
// Prune away low score responses if $\mathsf{R} = 1$. If
// $\mathsf{R} = 0$, no draft is discarded.
end if
end for
if $L = \emptyset$ **then**
Return $\perp$ // If all responses rejected, return $\perp$
else
Sample $i^\star \in [n]$ from dist. $P(i^\star = i) \propto e^{S_i}\mathbb{I}(i \in L)$
// Sample an undiscarded response proportional
// to its normalized exponentiated score
end if
Return: $Y_{i^\star}$

Algorithm 3 CASCADE subroutine

Input: Context tr, Draft blocks $\{Y_i\}_{i=1}^{n}$, Target model π_{target}, Draft model π_{draft}, Process reward model r
Hyperparameters: Cascade threshold τ_2
if $\max_i r(Y_i|\,\text{tr}) \geq \tau_2$ **then**
Return to "*Step I*" in Algorithm 1.
// i.e., sample from π_{draft} in the next iteration
else
Return to "*Step III*" in Algorithm 1.
// i.e., sample from π_{target} in the next iteration
end if

draft is accepted, the algorithm proceeds to the next iteration. If all of them are rejected, n fresh draft blocks are generated from target model instead. One of them is selected using SUBSAMPLE$(\cdot)$, this time not allowing for all responses to be rejected. Finally, the algorithm determines which (draft or target) model to use for draft generation in the next iteration: this is done using the CASCADE subroutine.

2.2. SUBSAMPLE subroutine

The SUBSAMPLE procedure first scores all the candidate blocks using the target, and reward models. Specifically, for each generated one-block completion of a reasoning trace tr, the following score function is computed,

$$S(y_i|\mathsf{tr}) = \log\left(\frac{\pi_{\text{target}}(Y_i|\mathsf{tr})}{\pi_{\text{draft}}(Y_i|\mathsf{tr})}\right) + \beta r(Y_i|\mathsf{tr}). \quad (1)$$

The score of each candidate is thresholded and one of the candidate blocks with scores exceeding a threshold τ are sampled with probability proportional to $\exp(S(y_i|\mathsf{tr}))$. In case all the candidates have low scores, none exceeding τ, then the subroutine returns a special symbol $\perp$ to indicate that none of the generations were deemed to be good, and for the remainder of the iteration, control is passed to the target model which instead generates candidate blocks.

2.2.1. Motivation behind scores in SUBSAMPLE

The design of the score (Equation (1)) in SUBSAMPLE is motivated by the objective of solving a one-step KL-regularized reward maximization problem. In each step of generation, given a base model π_{target} and a reward function r, the objective is to maximize,

$$\mathcal{L}_\beta(\pi) = \mathbb{E}_{\rho,\pi}[r(Y|\mathsf{tr})] - \frac{1}{\beta} D_{\text{KL}}(\pi \| \pi_{\text{target}}). \quad (2)$$

The optimal solution to this optimization problem has an explicit form (Korbak et al., 2022b;a; Yang et al., 2024b):

$$\pi^\star_\beta(y|\,\text{tr}) \propto \pi_{\text{target}}(y|\,\text{tr}) e^{\beta r(y|\,\text{tr})}. \quad (3)$$

To involve distributions and expectations under the draft model, we set $S(y|\,\text{tr}) = \log(\pi_{\text{target}}(y|\,\text{tr})/\pi_{\text{draft}}(y|\,\text{tr})) + \beta r(y|\,\text{tr})$. And this optimal policy can be written down in a different form by importance reweighting, Namely,

$$\pi^\star_\beta(y|\,\text{tr}) \propto \pi_{\text{draft}}(y|\,\text{tr}) e^{S(y|\,\text{tr})}. \quad (4)$$

The challenge of sampling from this distribution is computing the *partition function*, $Z(\text{tr}) = \mathbb{E}_{Y\sim\pi_{\text{draft}}(\cdot|\,\text{tr})}[e^{S(Y|\,\text{tr})}]$, the normalizing constant of the probability distribution. With this, we may view the SUBSAMPLE routine we consider in Algorithm 2 as a combination of two approaches, (i) approximating the partition function $Z(\text{tr})$ with samples, and (ii) rejection sampling, as detailed below.

Sampling by approximating the partition function $Z(\text{tr})$. The partition function $Z(\text{tr})$ can be approximated by a discrete summation.

$$Z(\text{tr}) = \mathbb{E}_{Y\sim\pi_{\text{draft}}(\cdot|\,\text{tr})}[e^{S(Y|\,\text{tr})}] \approx \frac{1}{n}\sum_{i=1}^{n} e^{S(Y_i|\,\text{tr})} = \widehat{Z}(x).$$

With this, we may sample a response Y by sampling from the distribution $P(Y = Y_i) = e^{S(Y_i|\,\text{tr})}/\widehat{Z}(x)$.

Rejection sampling. Another approach toward sampling from $\pi^\star_\beta(y|\,\text{tr})$ is to carry out rejection sampling with respect to the optimal distribution. Formally, for each draft index i, we may generate a uniform random variable $\eta_i \sim \mathsf{Unif}([0,1])$ and accept the i^{th} draft if,

$$\begin{aligned} \eta_i \le \max\left\{1, \frac{\pi^\star_\beta(Y_i|\,\text{tr})}{\pi_{\text{draft}}(Y_i|\,\text{tr})}\right\} &\overset{(a)}{\le} \frac{e^{S(Y_i|\,\text{tr})}}{e^{\|S\|_\infty}} \\ \iff \log(\eta_i) \le S(Y_i|\,\text{tr}) - \|S\|_\infty. & \quad (5) \end{aligned}$$

Where $\|S\|_\infty$ is the maximum score across all prompts and responses (this can be a tuned hyperparameter).

In Algorithm 2, the operation of pruning away the low score responses is an empirical approximation to rejection sampling (cf. Equation (5)). In the subsequent step of sampling a response (among the ones which survive the rejection step) proportional to their score, we are effectively constructing an approximation to the partition function. Thus, the SUBSAMPLE subroutine we consider in Algorithm 2 can be interpreted as a combination of sampling by approximating the partition function, and rejection sampling.

2.3. CASCADE subroutine

The CASCADE subroutine decides whether the next iteration must begin with candidate blocks drawn from the draft model, or from the target model. The motivation for having this step is that in the event that all the candidate blocks from the draft model are rejected, the meta-algorithm resamples a new set of candidates from the target model. This incurs the cost of generation from both the target as well as draft models, which is wasteful. In order to mitigate this issue, we use the previous generations to determine whether to use the draft model to generate the next set of candidate blocks, or directly use the target model. We design a heuristic CASCADE subroutine which thresholds the rewards of the previous candidates to determine whether to use the draft model or target model in the next iteration. The intuition is that trajectories with lower intermediate rewards may correspond to steps that are harder to complete, hencing requiring a more capable model.

3. Theoretical results

In this section, we describe theoretical guarantees for SPECS. We will show that under certain assumptions, as the beam size increases, the distribution of output of SPECS converges to the optimal policy for a KL-regularized reward maximization problem (Equation (6)). First we formalize some notation below.

Notation. Denote the space of prompts as $\mathcal{X}$, the space of responses as $\mathcal{Y}$, the draft model $\pi_{\text{draft}} : \mathcal{X} \to \Delta_{\mathcal{Y}}$ and the target model $\pi_{\text{target}} : \mathcal{X} \to \Delta_{\mathcal{Y}}$. Prompts and responses are composed of sequences of tokens, and responses are assumed to be complete: ending in a special "End-Of-Sentence" (EOS) token. We will assume that there exists a target distribution over prompts, denoted ρ, and the notation $\mathbb{E}_{\rho,\pi}[\cdot]$ is shorthand for $\mathbb{E}_{X\sim\rho, Y\sim\pi(\cdot|X)}[\cdot]$, where the random variable Y corresponds to full responses drawn from π under the prompt X. The goal is to design an algorithm that samples from the policy that maximizes the KL regularized expected reward,

$$\mathcal{L}_\beta(\pi) = \mathbb{E}_{\rho,\pi}[r(y|x)] - \frac{\mathbb{E}_\rho[D_{\text{KL}}(\pi(\cdot|x)\|\pi_{\text{target}}(\cdot|x))]}{\beta} \quad (6)$$

3.1. Warm-up: Infinite-block length regime

In this section we will build up our theoretical understanding of SPECS by considering the infinite block-length setting. In this setting, the drafts generated by the draft model are assumed to be full responses. In this case, the algorithm is simply select one candidate among a number of draft responses using the SUBSAMPLE subroutine.

To aid our analysis, we will study SPECS in the Poisson sampling model, where the number of drafts generated is randomized and distributed according to a Poisson random variable.

Assumption 3.1 (Poisson sampling model)**.** For a parameter $n > 0$, we assume that the number of drafts K, is a random variable drawn from the distribution $K \sim \mathrm{Poi}(n)_{>0}$ (i.e., a $\mathrm{Poi}(n)$ RV conditioned on being non-zero). In particular, as n grows to be large, the number of drafts sampled, K, concentrates tightly around n.

Assumption 3.2 (Soft rejection)**.** The subsampling subroutine considered in SUBSAMPLE (Algorithm 2) is implemented with "hard" rejection: all drafts with score below a threshold (specified by τ) are discarded immediately. We will analyze a "soft rejection" version of SPECS: For each draft Y_i, a uniform random variable $\eta_i \sim \mathrm{Unif}([0,1])$ is generated. The draft Y_i is then rejected if $e^{S(Y_i|\mathrm{tr})} \le e^{\tau}\eta_i$. The empirical implementation of Algorithm 2 implicitly corresponds to setting $\eta_i = 1$ deterministically, i.e., hard rejection. We will assume that $\tau = \|S(\cdot|\cdot)\|_\infty$.

Our first result establishes a convergence rate of $O(1/n^2)$ in the infinite-block setting. Increasing n scales the amount of test-time compute used.

Theorem 3.3 (Guarantees for SPECS)**.** *Suppose $n \ge 3$ and assume that the reward function lies in the range $[0, R]$ pointwise. Then, the policy π_{SPECS} returned by SPECS in the infinite block-length setting (Definition D.1), assuming the Poisson sampling model (Assumption 3.1) and soft rejection (Assumption 3.2), satisfies,*

$$D_{KL}(\pi_{SPECS}\|\pi^\star_\beta) \le \widetilde{\mathcal{O}}_n\left(C_{\mathtt{seq}}^2 \frac{e^{2\beta R}}{n^2}\right) \tag{7}$$

where, the "sequence-level coverage coefficient" $C_{\mathtt{seq}}$ is,

$$C_{\mathtt{seq}} = \left\|\frac{\pi_{\text{target}}(\cdot|x)}{\pi_{\text{draft}}(\cdot|x)}\right\|_\infty \left\|\frac{\pi_{\text{draft}}(\cdot|x)}{\pi_{\text{target}}(\cdot|x)}\right\|_\infty. \tag{8}$$

Remark 3.4. We in fact prove a stronger bound, where the LHS of Eq. (7) is replaced by $\log\Big(1 + D_{\chi^2}(\pi_{\mathtt{SPECS}}\|\pi^\star_\beta)\Big)$.

3.2. Finite-block length with idealized process feedback

In this section we extend the previous discussion to the finite-block setting. In this case, SPECS (Algorithm 1) is inherently iterative and the reward model considered in SUBSAMPLE (Algorithm 2) must be implemented with a process reward model (PRM) which is trained to provide block-level feedback. Prior to discussing our main results, we introduce some additional notation.

Notation. We assume responses can be broken down into smaller blocks. In practice, blocks may correspond to subsequences with a fixed length or subsequences which end in a special "End-Of-Block" (EOB) token. Given a (partial) response y, let $(y_1, \cdots, y_H)$ denote its blocks and let $y_{\le t} = (y_1, \cdots, y_t)$ denote the prefix of the first t blocks. By default, we will let $y_{\le 0} = \emptyset$. We will let $\mathcal{Y}$ denote the space of full responses, $\mathcal{Y}_t$ to denote the collection of blocks the model can generate at time t, and $\mathcal{Y}_{\le t} = \otimes_{t' \le t}\mathcal{Y}_{t'}$. H denotes the horizon: the maximum number of blocks that the model is allowed to generate.

Next, we introduce the "idealized" notion of PRM which we assume access to in implementing the finite-block version of SPECS. We begin with the notion of an optimal KL-regularized value function (Ziebart et al., 2008; Zhou et al., 2025).

Definition 3.5 (Optimal KL-regularized value function)**.** For a policy π, and any prompt $x \in \mathcal{X}$ and partial trace $y_{\le t}$, the optimal KL-regularized value of a policy is defined as,

$$V^\pi_\beta(x, y_{\le t}) = \frac{1}{\beta}\log\left(\mathbb{E}_{Y\sim\pi(\cdot|x,y_{\le t})}\left[e^{\beta r(x,Y)}\right]\right)$$

where Y is a full completion of the partial response $y_{\le t}$. Note that for any full response $y \in \mathcal{Y}$, $V^\pi_\beta(x, y) = r(x, y)$.

Definition 3.6 (Idealized Process Reward Model / optimal KL-regularized advantage function)**.** Let $r_{\mathtt{PRM}}$ denote the PRM corresponding to the KL-regularized advantage function with respect to some policy π. Formally, given any prefix of tokens / blocks $y_{\le t-1}$ of length $t-1$ and a subsequent token / block y_t,

$$r^\pi_{\mathtt{PRM}}(y_t|x, y_{\le t-1}) = V^\pi_\beta(x, y_{\le t}) - V^\pi_\beta(x, y_{\le t-1})$$

where, the KL-regularized value function V^π_β is defined in Definition 3.5. The cumulative advantage till block t is denoted, $r^\pi_{\mathtt{PRM}}(y_{\le t}|x) = \sum_{t'=1}^t r^\pi_{\mathtt{PRM}}(y_{t'}|x, y_{\le t'-1})$. The idealized PRM we consider is $r^{\text{target}}_{\mathtt{PRM}} = r^\pi_{\mathtt{PRM}}$ for $\pi = \pi_{\text{target}}$.

This above notion of PRM has also been used in recent works (Mudgal et al., 2024; Zhou et al., 2025; Brantley et al., 2025) for solving the KL-regularized reward maximization problem. The latter work provides an efficient and scalable approach for learning this PRM from offline datasets, and shows that policies trained by carrying out RL against this PRM perform well empirically, even on low generation budgets. For the purpose of keeping the theoretical presentation cleanest, we analyze SPECS when implemented with an exact version of this PRM.

In the infinite block-length setting, the guarantee of SPECS depended on a strong notion of coverage, $C_{\mathtt{seq}} = \|\pi_{\text{target}}(\cdot|x)/\pi_{\text{draft}}(\cdot|x)\|_\infty \|\pi_{\text{draft}}(\cdot|x)/\pi_{\text{target}}(\cdot|x)\|_\infty$, where the distributions are over full responses, a quantity which may scale exponentially with H. Our next result shows that a PRM defined in Definition 3.6 allows guarantees that scale polynomially with the horizon H, where the dependency on $C_{\mathtt{seq}}$ is relaxed to the maximum "per-block" ratio of densities between the target and draft models.

Table 1. Performance comparison of methods across datasets and beam search settings. Here n denotes per-step beam width, 'Lat. (s)' is average per problem latency in seconds and '% B' represents the percentage of steps generated by the big model B averaged across problems. (S: small model, B: big model)

Method	n	Acc (%) ↑	Lat (s) ↓	Lat/step (s) ↓	% B
AMC23					
BS (S)	4	38.3 ± 4.7	10.3 ± 0.6	0.80 ± 0.01	0
BS (B)	4	59.2 ± 1.2	18.3 ± 1.5	1.46 ± 0.06	100
SPECS	4	**59.2 ± 3.1**	14.8 ± 0.5	1.42 ±0.13	67.9 ± 1.8
BS (S)	8	46.7 ± 3.1	17.6 ± 2.3	1.14 ± 0.01	0
BS (B)	8	59.2 ± 5.1	21.6 ± 1.1	2.00 ± 0.06	100
SPECS	8	**62.5 ± 2.0**	20.7 ± 2.0	1.94 ±0.11	62.8 ± 2.1
BS (S)	16	50.8 ± 3.1	25.6 ± 3.1	1.81 ± 0.03	0
BS (B)	16	68.3 ± 4.2	35.0 ± 2.6	2.80 ± 0.13	100
SPECS	16	**68.3 ± 2.4**	28.8 ± 3.5	2.70 ± 0.05	70.3 ± 1.7
MATH500					
BS (S)	4	64.3 ± 2.9	7.9 ± 1.0	0.65 ± 0.04	0
BS (B)	4	69.3 ± 4.1	11.2 ± 3.7	1.13 ± 0.01	100
SPECS	4	**69.7 ± 1.2**	9.2 ± 0.8	1.12 ± 0.14	42.3 ± 0.8
BS (S)	8	66.7 ± 2.1	10.7 ± 0.6	1.01 ± 0.03	0
BS (B)	8	71.3 ± 0.5	13.6 ± 1.9	1.51 ± 0.00	100
SPECS	8	**72.5 ± 0.4**	12.6 ± 0.6	1.49 ± 0.56	41.4 ± 0.9
BS (S)	16	74.5 ± 0.5	19.9 ± 1.9	1.57 ± 0.10	0
BS (B)	16	75.5 ± 0.5	21.3 ± 0.5	2.15 ± 0.02	100
SPECS	16	**79.3 ± 1.9**	19.7 ± 0.3	2.23 ± 0.04	40.2 ± 0.8
OlympiadBench					
BS (S)	4	25.0 ± 1.6	7.9 ± 0.2	0.81 ± 0.2	0
BS (B)	4	39.3 ± 2.1	13.1 ± 0.3	1.32 ± 0.3	100
SPECS	4	**41.3 ± 0.9**	13.1 ± 0.3	1.30 ± 3.4	83.1 ± 0.4
BS (S)	8	31.0 ± 0.8	13.4 ± 0.3	1.22 ± 0.3	0
BS (B)	8	41.3 ± 2.1	17.9 ± 0.5	1.81 ± 0.5	100
SPECS	8	**41.7 ± 1.2**	16.9 ± 0.5	1.80 ± 1.5	80.3 ± 0.4
BS (S)	16	31.3 ± 3.3	21.1 ± 1.1	2.02 ± 1.1	0
BS (B)	16	43.7 ± 0.5	25.4 ± 0.9	2.57 ± 0.9	100
SPECS	16	**44.0 ± 2.9**	24.6 ± 1.0	2.56 ± 6.2	61.5 ± 0.3

Theorem 3.7. *Suppose SPECS is implemented with the idealized PRM as defined in Definition 3.6. Then, for reasoning problems over H blocks, in the finite-block length setting, assuming the Poisson sampling model (Assumption 3.1), the policy $\pi_{\beta,n}$ returned by SPECS satisfies,*

$$D_{KL}(\pi_{\beta,n}, \pi^\star_\beta) \leq \widetilde{\mathcal{O}}_n \left(H \cdot C^2_{\texttt{block}} \frac{e^{2\beta R}}{n^2} \right)$$

Here, we assume that the PRM rewards are in the range $[0, R]$ and, the block-level coverage coefficient $C_{\texttt{block}}$ is,

$$C_{\texttt{block}} \tag{9}$$
$$= \sup_{t \geq 0} \sup_{y_{\leq t} \in \mathcal{Y}_{\leq t}} \left\| \frac{\pi_{\text{target}}(\cdot|x, y_{\leq t})}{\pi_{\text{draft}}(\cdot|x, y_{\leq t})} \right\|_\infty \left\| \frac{\pi_{\text{draft}}(\cdot|x, y_{\leq t})}{\pi_{\text{target}}(\cdot|x, y_{\leq t})} \right\|_\infty$$

where $\pi(\cdot|x, y_{\leq t})$ denotes the next-step distribution given the partial trace $y_{\leq t}$ and prompt x.

4. Experiments

We present empirical evaluation results on the *latency-accuracy tradeoff* of SPECS.

4.1. Experimental Setup

Dataset: We conduct experiments on three mathematical reasoning benchmarks, AMC23 (Faires & Wells, 2022), MATH500 (Hendrycks et al., 2021) and OlympiadBench dataset (He et al., 2024). AMC23 dataset comprises of 40 mathematical reasoning problems, testing concepts in algebra, geometry, number theory, and combinatorics, with an emphasis on precision and strategic problem-solving. MATH500 is a widely used benchmark for evaluating mathematical problem-solving capabilities of language models. OlympiadBench consists of challenging olympiad-level problems, requiring deep reasoning. Following the prior works (Qiu et al., 2024; Zhang et al., 2024), we use a randomly selected subset of 100 questions from MATH500 and OlympiadBench datasets to make evaluation faster.

Models: We use several models from the Qwen family as our base generation models: Qwen-1.5B-Instruct and Qwen-7B-Instruct (Yang et al., 2024a). These models vary significantly in size, allowing us to test the scalability of our approach. For reward-guided selection in both beam search and SPECS, we utilize the Qwen-7B-Math-PRM (Yang et al., 2024a), a process reward model specifically trained for mathematical reasoning tasks.

Baselines: Our primary baseline is the standard beam search guided by the Qwen-7B-Math-PRM reward model. This involves generating multiple candidate sequences (beams) blockwise using the base model and selecting the highest scoring sequence according to the reward model (Snell et al., 2025; Beeching et al., 2024). We compare SPECS against baselines using one of the base models (Qwen-1.5B, 7B) and the same reward model.

We evaluate performance based on accuracy and latency of the algorithms. Since all of our datasets are on math, we calculate the accuracy with the percentage of problems correctly solved. For latency, we measure the wall-clock time required to generate the solution for each problem. We report average latency. Our main focus is the latency-accuracy trade-off, visualizing how accuracy changes as a function of latency for different methods and configurations (e.g., varying beam widths).

Further details regarding the experiments are presented in Appendix G, such as the hardware setup, the hyperparameter choice and additional experiments.

4.2. Results

We present the comparative results on latency and accuracy for the AMC23, MATH500 and OlympiadBench datasets in

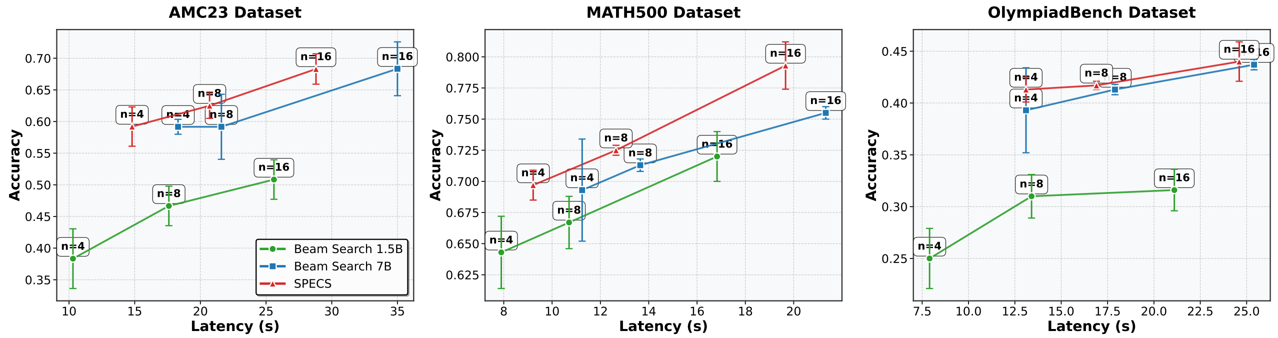

Figure 2. Accuracy vs per-query average latency curves for SPECS and beam search with draft or target model alone on AMC23, MATH500 and OlympiadBench datasets. The error bars show the standard deviation computed over 3 independent runs. The draft model is Qwen-1.5B-Instruct and the target model is Qwen-7B-Instruct.

Table 1.

For a fixed beam size n, SPECS is able to achieve comparable or even higher accuracy compared to the strong baseline of the beam search with the large model. We attribute this accuracy gain to the strength that SPECS is able to utilize the ability of both the small and large models. Note that the performance comes with the benefit of up to *∼19.1%* latency savings, demonstrating a favorable latency-accuracy trade-off of our proposed SPECS method.

We also report the percentage of steps generated by the large model in the table. The percentage is higher for the more challenging dataset of AMC23, and OlympiadBench compared to MATH500, suggesting that SPECS is able to adapt the usage of large model to the hardness of the questions. Additional plots regarding the target model usage histograms of these runs are in Appendix G.4.

5. Conclusion

In this work, we address the critical challenge of balancing latency and utility in test-time scaling for large language models. We propose SPECS, a latency-aware test-time scaling method for large language models (LLMs) that significantly improves the latency-accuracy tradeoff through speculative drafting. SPECS utilizes a smaller, faster draft model to generate candidate sequences, which we then efficiently evaluate using a larger target model and a dedicated PRM. We propose novel integration strategies to merge the draft model, target model, and the PRM, including reward-guided soft verification and a reward-based deferral mechanism, to dynamically balance computational resources with response latency. Empirically, we evaluate SPECS across the MATH500, AMC23, and OlympiadBench datasets, demonstrating its superior performance. SPECS consistently achieves accuracy comparable to or exceeding traditional beam search methods while delivering substantial reductions in latency—up to approximately 19.1%. Additionally, our theoretical analysis confirms that SPECS converges to the optimal solution of a KL-regularized reinforcement learning objective as the beam width increases.

References

Agrawal, A., Kedia, N., Panwar, A., Mohan, J., Kwatra, N., Gulavani, B., Tumanov, A., and Ramjee, R. Taming Throughput-Latency tradeoff in LLM inference with Sarathi-Serve. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 117–134, 2024.

Beeching, E., Tunstall, L., and Rush, S. Scaling test-time compute with open models, December 2024. URL https://huggingface.co/spaces/HuggingFaceH4/blogpost-scaling-test-time-compute. Accessed: 2025-04-10.

Beirami, A., Agarwal, A., Berant, J., D'Amour, A., Eisenstein, J., Nagpal, C., and Suresh, A. T. Theoretical guarantees on the best-of-n alignment policy. *arXiv preprint arXiv:2401.01879*, 2024.

Block, A. and Polyanskiy, Y. The sample complexity of approximate rejection sampling with applications to smoothed online learning. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 228–273. PMLR, 2023.

Brantley, K., Chen, M., Gao, Z., Lee, J. D., Sun, W., Zhan, W., and Zhang, X. Accelerating rl for llm reasoning with optimal advantage regression. *arXiv preprint arXiv:2505.20686*, 2025.

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. Large language monkeys: Scaling inference compute with repeated sampling, 2024. URL https://arxiv.org/abs/2407.21787.

Cai, T., Li, Y., Geng, Z., Peng, H., Lee, J. D., Chen, D., and Dao, T. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024.

Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., and Jumper, J. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318.

Chen, Z., May, A., Svirschevski, R., Huang, Y., Ryabinin, M., Jia, Z., and Chen, B. Sequoia: Scalable, robust, and hardware-aware speculative decoding. *arXiv preprint arXiv:2402.12374*, 2024.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Dohan, D., Xu, W., Lewkowycz, A., Austin, J., Bieber, D., Lopes, R. G., Wu, Y., Michalewski, H., Saurous, R. A., Sohl-dickstein, J., Murphy, K., and Sutton, C. Language model cascades, 2022. URL https://arxiv.org/abs/2207.10342.

Faires, J. D. and Wells, D. *The Contest Problem Book VIII: American Mathematics Competitions (AMC 10) 2000–2007*, volume 19. American Mathematical Society, 2022.

Geuter, J., Mroueh, Y., and Alvarez-Melis, D. Guided speculative inference for efficient test-time alignment of llms, 2025. URL https://arxiv.org/abs/2506.04118.

He, C., Luo, R., Bai, Y., Hu, S., Thai, Z. L., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Huang, A., Block, A., Liu, Q., Jiang, N., Krishnamurthy, A., and Foster, D. J. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment, 2025. URL https://arxiv.org/abs/2503.21878.

Jitkrittum, W., Gupta, N., Menon, A. K., Narasimhan, H., Rawat, A., and Kumar, S. When does confidence-based cascade deferral suffice? *Advances in Neural Information Processing Systems*, 36:9891–9906, 2023.

Korbak, T., Elsahar, H., Kruszewski, G., and Dymetman, M. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220, 2022a.

Korbak, T., Perez, E., and Buckley, C. RL with KL penalties is better viewed as Bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1083–1091, 2022b.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Leviathan, Y., Kalman, M., and Matias, Y. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2022.

Liao, B., Xu, Y., Dong, H., Li, J., Monz, C., Savarese, S., Sahoo, D., and Xiong, C. Reward-guided speculative decoding for efficient llm reasoning, 2025. URL https://arxiv.org/abs/2501.19324.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Miao, X., Oliaro, G., Zhang, Z., Cheng, X., Wang, Z., Zhang, Z., Wong, R. Y. Y., Zhu, A., Yang, L., Shi, X., Shi, C., Chen, Z., Arfeen, D., Abhyankar, R., and Jia, Z. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. ASPLOS '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703867. doi: 10.1145/3620666.3651335. URL https://doi.org/10.1145/3620666.3651335.

Mudgal, S., Lee, J., Ganapathy, H., Li, Y., Wang, T., Huang, Y., Chen, Z., Cheng, H.-T., Collins, M., Strohman, T., Chen, J., Beutel, A., and Beirami, A. Controlled decoding from language models. *International Conference on Machine Learning*, 2024.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. Webgpt: Browser-assisted question-answering with human feedback, 2022. URL https://arxiv.org/abs/2112.09332.

Narasimhan, H., Jitkrittum, W., Rawat, A. S., Kim, S., Gupta, N., Menon, A. K., and Kumar, S. Faster cascades via speculative decoding. *arXiv preprint arXiv:2405.19261*, 2024.

OpenAI. Learning to reason with llms. 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

Patel, P., Choukse, E., Zhang, C., Shah, A., Goiri, Í., Maleki, S., and Bianchini, R. Splitwise: Efficient generative llm inference using phase splitting. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pp. 118–132. IEEE, 2024.

Qiu, J., Lu, Y., Zeng, Y., Guo, J., Geng, J., Wang, H., Huang, K., Wu, Y., and Wang, M. Treebon: Enhancing inference-time alignment with speculative tree-search and best-of-n sampling. *arXiv preprint arXiv:2410.16033*, 2024.

Snell, C. V., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Sun, H., Haider, M., Zhang, R., Yang, H., Qiu, J., Yin, M., Wang, M., Bartlett, P., and Zanette, A. Fast best-of-n decoding via speculative rejection, 2024. URL https://arxiv.org/abs/2410.20290.

Sun, Z., Suresh, A. T., Ro, J. H., Beirami, A., Jain, H., and Yu, F. Spectr: Fast speculative decoding via optimal transport. *Advances in Neural Information Processing Systems*, 36:30222–30242, 2023.

Tiwari, R., Xi, H., Tomar, A., Hooper, C., Kim, S., Horton, M., Najibi, M., Mahoney, M. W., Keutzer, K., and Gholami, A. Quantspec: Self-speculative decoding with hierarchical quantized kv cache. *arXiv preprint arXiv:2502.10424*, 2025.

Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., and Higgins, I. Solving math word problems with process- and outcome-based feedback, 2022. URL https://arxiv.org/abs/2211.14275.

Verdun, C. M., Oesterling, A., Lakkaraju, H., and Calmon, F. P. Soft best-of-n sampling for model alignment, 2025. URL https://arxiv.org/abs/2505.03156.

Wang, J., Su, Y., Li, J., Xia, Q., Ye, Z., Duan, X., Wang, Z., and Zhang, M. Opt-tree: Speculative decoding with adaptive draft tree structure. *Transactions of the Association for Computational Linguistics*, 13:188–199, 2025a.

Wang, P., Li, L., Shao, Z., Xu, R. X., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. Math-shepherd: Verify and reinforce llms step-by-step without human annotations, 2024. URL https://arxiv.org/abs/2312.08935.

Wang, Z., Azmat, M., Li, A., Horesh, R., and Yurochkin, M. Speculate, then collaborate: Fusing knowledge of language models during decoding, 2025b. URL https://arxiv.org/abs/2502.08020.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

Yang, J. Q., Salamatian, S., Sun, Z., Suresh, A. T., and Beirami, A. Asymptotics of language model alignment. *arXiv preprint arXiv:2404.01730*, 2024b.

Yuan, Z., Shang, Y., Zhou, Y., Dong, Z., Zhou, Z., Xue, C., Wu, B., Li, Z., Gu, Q., Lee, Y. J., Yan, Y., Chen, B., Sun, G., and Keutzer, K. Llm inference unveiled: Survey and roofline model insights, 2024. URL https://arxiv.org/abs/2402.16363.

Yue, M., Zhao, J., Zhang, M., Du, L., and Yao, Z. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning, 2024. URL https://arxiv.org/abs/2310.03094.

Zhang, R., Haider, M., Yin, M., Qiu, J., Wang, M., Bartlett, P., and Zanette, A. Accelerating best-of-n via speculative rejection. In *ICML 2024 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2024.

Zhang, Z., Zheng, C., Wu, Y., Zhang, B., Lin, R., Yu, B., Liu, D., Zhou, J., and Lin, J. The lessons of developing process reward models in mathematical reasoning, 2025. URL https://arxiv.org/abs/2501.07301.

Zheng, C., Zhang, Z., Zhang, B., Lin, R., Lu, K., Yu, B., Liu, D., Zhou, J., and Lin, J. Processbench: Identifying process errors in mathematical reasoning, 2024. URL https://arxiv.org/abs/2412.06559.

Zhou, J. P., Wang, K., Chang, J., Gao, Z., Kallus, N., Weinberger, K. Q., Brantley, K., and Sun, W. Q#: Provably optimal distributional rl for llm post-training. *arXiv preprint arXiv:2502.20548*, 2025.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A. Related work

Reward-guided search. Our algorithm is related to the general literature of process reward modeling (Uesato et al., 2022; Lightman et al., 2023; Wang et al., 2024; Snell et al., 2025; Beeching et al., 2024; Zheng et al., 2024; Zhang et al., 2025) and reward-guided search (Mudgal et al., 2024; Beeching et al., 2024; Sun et al., 2024; Qiu et al., 2024; Snell et al., 2025). Among these, the closest to our is reward-guided speculative decoding (Liao et al., 2025), which also uses a draft model to propose each reasoning step and uses a process reward model to decide on whether to keep the proposal. SPECS generalizes this by proposing multiple candidate reasoning and combine signals from both the large model and reward model to decide on the candidate reasoning step to keep. Empirically, we observe our method achieves better latency / utility tradeoff compared to reward-guided speculative decoding.

Concurrent work of (Geuter et al., 2025) also considers soft draft verification by sampling from an exponentially tilted distribution. While their algorithm closes part of the accuracy gap between the draft model and the target model, our algorithm is able to achieve even higher accuracy than the target model alone. Moreover, we also provide convergence guarantees on the multi-step version of our algorithm.

Tree-based speculative decoding. Obtaining one sequence from multiple draft sequences has been studied in the speculative decoding literature with the goal of maximizing the likelihood/length of draft tokens accepted while maintaining the same distribution as the large model (Sun et al., 2023; Miao et al., 2024; Cai et al., 2024; Wang et al., 2025a; Chen et al., 2024). Compared to these, the goal of our soft verification algorithm is to maximize the success rate of downstream applications such as math-reasoning as guided by both the target model and the reward model. Other soft verification methods for speculative decoding (Wang et al., 2025b) consider token-level verification while we focus on step-level candidate selection.

Formal analysis of best-of-N variants. The theoretical analysis of our soft verification algorithm is related to (Verdun et al., 2025; Huang et al., 2025), which consider alternative candidate selection methods other than Top-1 selection. Our analysis captures the convergence rate in presence of draft models and multiple decoding steps.

Cascading. Cascading techniques (Dohan et al., 2022; Jitkrittum et al., 2023; Yue et al., 2024; Narasimhan et al., 2024) also utilize multiple language models and decide on the model to used based on certain deferral rule. The switching step in our algorithm can be viewed as imposing a deferral rule based on small, target, and reward models. Additionally, our deferral rule operates at the step-level while previous work usually focuses on individual tokens or the entire generation.

B. Theory

First, we begin with a well-known connection between the KL regularized reward maximization objective and the KL divergence between the learnt policy and the optimal aligned policy (Korbak et al., 2022b;a; Yang et al., 2024b). We include the proof for the sake of completeness.

Lemma B.1 (Equivalence of KL minimization and KL-regularized reward maximization)**.** *For any policy π,*

$$\mathcal{L}_\beta(\pi^\star_\beta) - \mathcal{L}_\beta(\pi) = D_{KL}(\pi \| \pi^\star_\beta) \tag{10}$$

where $\pi^\star_\beta$ is the maximizer of $\mathcal{L}_\beta(\pi)$.

Proof. By (Yang et al., 2024b), the optimal solution to Equation (6), $\pi^\star_\beta$ takes the structural form,

$$\pi^\star_\beta(y|x) = \frac{\pi_{\text{target}}(y|x)e^{\beta r(y|x)}}{Z(x)}$$

where $Z(x) = \mathbb{E}_{y\sim\pi_{\text{target}}(y|x)}[e^{\beta r(y|x)}]$. Thus,

$$\begin{aligned}
\mathcal{L}_\beta(\pi) &= \mathbb{E}_{\rho,\pi}[r(y|x)] - \frac{1}{\beta}\mathbb{E}_\rho[\mathbb{E}_{y\sim\pi(\cdot|x)}[\log(\pi(y|x)/\pi_{\text{target}}(y|x))]] \\
&= -\frac{1}{\beta}\mathbb{E}_\rho\big[\mathbb{E}_{y\sim\pi(\cdot|x)}\big[\log\Big(\pi(y|x)/(\pi_{\text{target}}(y|x)e^{\beta(r(y|x))})\Big)\big]\big] \\
&= -\frac{1}{\beta}\mathbb{E}_\rho[\log(Z(x))] - \frac{1}{\beta}\mathbb{E}_\rho\big[\mathbb{E}_{y\sim\pi(\cdot|x)}\big[\log\big(\pi(y|x)/\pi^\star_\beta(y|x)\big)\big]\big] \\
&= \mathcal{L}_\beta(\pi^\star_\beta) - \frac{1}{\beta}\mathbb{E}_\rho\big[D_{\text{KL}}(\pi(\cdot|x)\|\pi^\star_\beta(\cdot|x))\big]
\end{aligned}$$

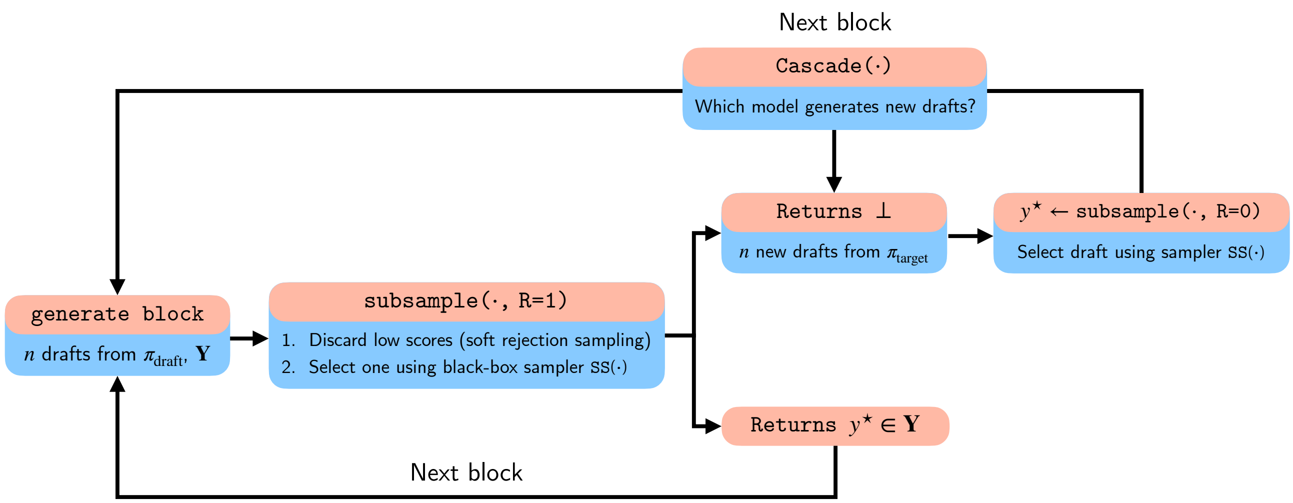

Figure 3. A flowchart of the SPECS meta-algorithm

Rearranging completes the proof. □

Therefore, up to an additive error that doesn't depend on π, the loss $\mathcal{L}_\beta(\pi)$ measures the KL divergence between the distribution over responses induced by π and $\pi^\star_\beta$ under prompts generated from ρ.

The purpose of introducing this result is to motivate the reason for choosing the error measure we study throughout the paper as the *reverse* KL divergence, rather than the forward KL divergence. The former is closely connected with the KL regularized reward maximization objective (it measures the suboptimality gap), while the latter is not so clearly connected. Indeed, these objectives penalize the learnt policy differently: the latter more strongly penalizes policies for placing too high mass on certain responses while the former more strongly penalizes policies which place too low mass on responses.

Having introduced this connection, we move on to analyzing SPECS. However, prior to jumping into this, we first break down the approach and study the heart of the algorithm which is the SUBSAMPLE(·) subroutine: this determines how to subselect one out of a set of responses (or reject all of them) in a way which takes into account signals from the target, draft and reward models.

C. Analysis of key sampling strategies

Recall that the SUBSAMPLE(·) procedure we consider in Algorithm 2 is a combination of two simpler approaches: (a) rejection sampling and (b) sampling by approximating the partition function. Building up to our analysis of SPECS in the infinite and finite block-length settings, we analyze these approaches in this section. Let us first introduce some notation to make the presentation simpler.

Definition C.1 (Score function)**.** Define the score function,

$$\phi_\beta(y|x) = \frac{\pi_{\text{target}}(y|x)}{\pi_{\text{draft}}(y|x)} e^{\beta r(y|x)}.$$

Furthermore, define $\phi_{\max} = \max_{x\in\mathcal{X},y\in\mathcal{Y}} \phi_\beta(y|x)$ and $\phi_{\min} = \min_{x\in\mathcal{X},y\in\mathcal{Y}} \phi_\beta(y|x)$. Note that the score ϕ_β is simply the exponentiated version of the score $S(y|x)$ defined in Equation (1).

Next we define rejection sampling and sampling by approximating the partition function formally. In Appendix D we will show how to combine their analyses to result in a guarantee for SPECS in the infinite block-length regime.

C.1. Analysis of rejection sampling

In this section we introduce and analyze rejection sampling. Analyses of the form of Theorem C.3 (and extensions thereof) have been carried out a number of times previously in the literature, such as in (Block & Polyanskiy, 2023).

Definition C.2 ((Soft) rejection sampling)**.** Given a prompt x and a set of draft responses $\{Y_i\}_{i=1}^n$ generated from π_{draft}, rejection sampling returns a value in $\{Y_i\}_{i=1}^n \cup \{\perp\}$. For each i, a random variable $\eta_i \sim \text{Unif}([0,1])$ is drawn, and Y_i is accepted if $\eta_i \le e^{S(Y_i|x) - \|S(\cdot|\cdot)\|_\infty}$. The first accepted response among the Y_i's is returned, and if all of them are rejected, a special symbol $\perp$ is returned to indicate the same.

The main result we establish is that rejection sampling returns a sample from the optimal solution to KL regularized reward maximization problem if none of the drafts are rejected. Furthermore, the probability of all the drafts being rejected decays gracefully as the number of drafts n grows.

Theorem C.3. *Consider a prompt x, let $Y^\star$ be the response returned by rejection sampling when n is drawn from any distribution (or is deterministic) and the n responses $\{Y_i\}_{i=1}^n$ are drawn independently from $\pi_{\text{draft}}(\cdot|x)$. Then,*

$$\Pr(Y^\star = y | Y^\star \neq \perp, x, n) = \pi^\star_\beta(y|x).$$

In other words, conditioned on n (which may be random) and on the event that at least one of the samples was accepted, $Y^\star$ is distributed exactly as the optimal solution to the KL regularized reward maximization objective (Equation (2)). On the other hand, the probability that the i^{th} draft is rejected equals,

$$\Pr(Y_i \textit{ is rejected} | x, n) = 1 - Z(x) e^{-\|S(\cdot|\cdot)\|_\infty}$$

where $Z(x) = \mathbb{E}_{Y\sim\pi_{\text{target}}(\cdot|x)}\left[e^{\beta r(Y|x)}\right]$. By independence of the drafts, $\Pr(Y^\star = \perp | x, n) = \left(1 - Z(x)e^{-\|S(\cdot|\cdot)\|_\infty}\right)^n$

Proof. We prove the second statement first. For any index $i \in [n]$, if $Y_i = y$, the probability with which Y_i is accepted is $e^{S(y|x) - \|S(\cdot|\cdot)\|_\infty}$ (which is ≤ 1 by design). Therefore,

$$\begin{aligned}
\Pr(Y_i \text{ is accepted } | Y_1 = y, x, n) &= e^{S(y|x) - \|S(\cdot|\cdot)\|_\infty} \\
\implies \Pr(Y_i \text{ is accepted } | x, n) &= \mathbb{E}_{Y\sim\pi_{\text{draft}}(\cdot|x)}\left[e^{S(Y|x) - \|S(\cdot|\cdot)\|_\infty}\right] \\
&= \mathbb{E}_{Y\sim\pi_{\text{target}}(\cdot|x,n)}\left[e^{\beta r(Y|x)}\right] e^{-\|S(\cdot|\cdot)\|_\infty} \\
&= Z(x) e^{-\|S(\cdot|\cdot)\|_\infty}
\end{aligned}$$

This implies that,

$$\begin{aligned}
\Pr(Y^\star = \perp | x, n) &= \prod_{i=1}^n (1 - \Pr(Y_i \text{ is accepted} | x, n)) \\
&= \left(1 - Z(x) e^{-\|S(\cdot|\cdot)\|_\infty}\right)^n
\end{aligned}$$

To prove the first statement, observe that by Bayes rule, for any index $i \in [n]$,

$$\begin{aligned}
\Pr(Y_i = y | Y_i \text{ is accepted}, x, n) &= \frac{\Pr(Y_i = y, Y_i \text{ is accepted} | x, n)}{\Pr(Y_i \text{ is accepted} | x, n)} \\
&= \frac{\pi_{\text{draft}}(y|x) e^{S(y|x) - \|S(\cdot|\cdot)\|_\infty}}{Z(x) e^{-\|S(\cdot|\cdot)\|_\infty}} \\
&= \frac{\pi_{\text{target}}(y|x) e^{\beta r(y|x)}}{Z(x)} \\
&= \pi^\star_\beta(y|x)
\end{aligned}$$

By independence of the drafts,

$$\Pr(Y_i = y | \{Y_i \text{ is accepted}\} \cap \{Y_1, \cdots, Y_{i-1} \text{ are rejected}\} \cap \{x, n\}) = \pi^\star_\beta(y|x)$$

This implies that,

$$\begin{aligned}\Pr(Y^\star = y | \{Y^\star \neq \perp\} \cap \{x, n\}) &= \frac{\sum_{i=1}^n \Pr(\{Y_i = y\} \cap \{Y_1, \cdots, Y_{i-1} \text{ are rejected}\} \cap \{Y_i \text{ is accepted}\} | x, n)}{\sum_{i=1}^n \Pr(\{Y_1, \cdots, Y_{i-1} \text{ are rejected}\} \cap \{Y_i \text{ is accepted}\} | x, n)} \\ &= \frac{\sum_{i=1}^n \pi^\star_\beta(y|x) \cdot \Pr(\{Y_1, \cdots, Y_{i-1} \text{ are rejected}\} \cap \{Y_i \text{ is accepted}\} | x, n)}{\sum_{i=1}^n \Pr(\{Y_1, \cdots, Y_{i-1} \text{ are rejected}\} \cap \{Y_i \text{ is accepted}\} | x, n)} \\ &= \pi^\star_\beta(y|x).\end{aligned}$$

Completing the proof of the first statement. □

C.2. Sampling by approximating the partition function

Definition C.4 (Sampling by partition function approximation)**.** Given a prompt x and a set of draft responses $\mathbf{Y}_{\text{draft}} = \{Y_i\}_{i=1}^n$ generated i.i.d. from π_{draft}, sampling by partition function approximation returns a value in $\{Y_i\}_{i=1}^n$. This approach returns Y_i with probability proportional to $e^{S(Y_i|x)}$. In other words, denoting $Y^\star$ as the output of the procedure,

$$\Pr(Y^\star = Y_i) = \frac{e^{S(Y_i|x)}}{Z_{\mathbf{Y}_{\text{draft}},\beta}(x)},$$
$$\text{where, } Z_{\mathbf{Y}_{\text{draft}},\beta}(x) = \sum_{j\in[n]} e^{S(Y_j|x)}.$$

Theorem C.5. *Suppose $n \geq 3$ in the Poisson sampling model (Assumption 3.1). In the infinite block-length setting, the policy $\pi_{\texttt{part-apx},\beta,n}$ induced by approximating the partition function (Definition C.4) satisfies,*

$$D_{KL}(\pi_{\texttt{part-apx},\beta,n} \| \pi^\star_\beta) \leq \mathcal{O}\left(\frac{\log^2(n)}{n^2}\left(\frac{\phi_{\max}}{\phi_{\min}}\right)^2\right).$$

$\phi_{\max}$ and $\phi_{\min}$ are defined in Definition C.1.

Since it is more involved, we defer the proof of Theorem C.5 to Appendix F. We have the following two corollaries of Theorem C.5.

Corollary C.6 (Target sampling)**.** *Suppose $n \geq 3$ and suppose drafts are generated from π_{target} (i.e., we choose $\pi_{\text{draft}} \leftarrow \pi_{\text{target}}$). In this case the score function $\phi_\beta(y|x) = e^{\beta r(y|x)}$. Assuming rewards lie in the range $[0, R]$, in the infinite block-length setting (Definition D.1) under the Poisson sampling model (Assumption 3.1),*

$$D_{KL}(\pi_{\texttt{part-apx},\beta,n} \| \pi^\star_\beta) \leq \widetilde{\mathcal{O}}_n\left(\frac{e^{2\beta R}}{n^2}\right)$$

Corollary C.7 (Speculative sampling)**.** *Suppose $n \geq 3$. Then, in the infinite block-length setting (Definition D.1) under the Poisson sampling model (Assumption 3.1), and assuming rewards lie in the range $[0, R]$,*

$$D_{KL}(\pi_{\texttt{part-apx},\beta,n} \| \pi^\star_\beta) \leq \widetilde{\mathcal{O}}_n\left(C_{\texttt{seq}}^2 \frac{e^{2\beta R}}{n^2}\right)$$

where $C_{\texttt{seq}} = \|\pi_{\text{target}}(\cdot|x)/\pi_{\text{draft}}(\cdot|x)\|_\infty \|\pi_{\text{draft}}(\cdot|x)/\pi_{\text{target}}(\cdot|x)\|_\infty$, as defined in Equation (8).

Remark C.8. In all three prior results, Theorem C.5 and Corollaries C.6 and C.7, the proof will argue the stronger upper bound which replaces the LHS by the strictly larger quantity, $\log\Big(1 + D_{\chi^2}(\pi_{\texttt{part-apx},\beta,n} \| \pi^\star_\beta)\Big)$.

When the target and draft models are identical, then $C_{\texttt{seq}} = 1$. We also show that the rate of decay in Corollary C.7 is essentially tight when the target and draft models are identical.

Theorem C.9. *There exists a target model π_{target} and reward model over a 2-ary response space $\mathcal{Y} = \{y_0, y_1\}$, with rewards lying in the range $[0, R]$, such that, the policy $\pi_{\texttt{part-apx},\beta,n}$ returned by sampling by partition function approximation Definition C.4 in the infinite block-length setting (Definition D.1) with $\pi_{\text{draft}} = \pi_{\text{target}}$, under the Poisson sampling model (Assumption 3.1) satisfies,*

$$D_{KL}(\pi_{\texttt{part-apx},\beta,n} \| \pi^\star_\beta) \geq \Omega_n\left(\frac{e^{2\beta R}}{n^2}\right),$$

assuming that $n \geq e^{\beta R}$.

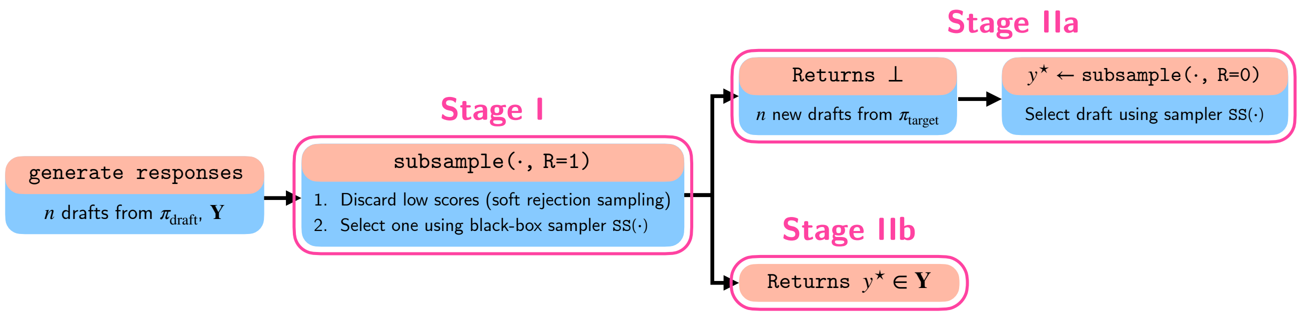

Figure 4. A flowchart of the SPECS meta-algorithm in the infinite block regime: we abstract away the black-box sampler as SS(·), which is chosen as the sampling approach induced by approximating the partition function (Definition C.4)

D. Analysis of SPECS in the infinite block-length regime: Proof of Theorem 3.3

In this section we will build up our theoretical understanding of SPECS by considering the infinite block-length setting, where $\gamma \to \infty$. In this setting, the drafts generated by the draft model are assumed to be full responses, and responses are not truncated to be at most γ tokens long. In this case, the cascade subroutine plays no role. The algorithm is described by the flowchart illustrated in Figure 4.

Definition D.1 (Infinite block-length regime)**.** The infinite block-length regime corresponds to taking $\gamma \to \infty$. Operationally, the draft/target model generates full responses and an outcome reward model is used to score responses.

In order to analyze SPECS in this setting, we will first provide a view of the algorithm from a more abstract lens.

D.1. Two-stage sampling procedure

Given a set of $N \sim \text{Poi}(n)|_{>0}$ candidate responses sampled i.i.d. from a draft model, $\mathbf{Y}_{\text{draft}} = \{Y_i\}_{i=1}^N \overset{\text{i.i.d.}}{\sim} \pi_{\text{draft}}(\cdot|x)$, we will consider a two-stage approach to select one among them. To keep the presentation relatively general, we assume that we are given access to a black-box subsampling routine $\texttt{SS}(\cdot;\cdot,\cdot) : \mathcal{Y}^\star \times \Pi \times \mathbb{R} \to \mathcal{Y}$, which picks out a response among a set of drafts (and is not allowed to reject all of them at once) and is parameterized by the policy which generates these drafts (the space of policies is denoted Π) and a scalar parameter (akin to the sampling temperature). The procedure we consider is precisely described in the flowchart in Figure 4, with the only distinction that n is not fixed beforehand and is drawn from a distribution. The two-stage procedure is described below,

1. **Stage I.** Carry out soft rejection sampling (Assumption 3.2) to determine which responses to accept. Namely, for each $i \in [N]$, generate a uniform random variable $\eta_i \sim \text{Unif}([0,1])$ and say $Y_i \in \mathbf{Y}_{\text{draft}}$ is "accepted" if,
$$\eta_i \leq e^{S_{\beta_0}(Y_i|x) - \|S_{\beta_0}(\cdot|\cdot)\|_\infty}$$
Here we make the temperature in the score $S_{\beta_0}(y|x) = \log\left(\frac{\pi_{\text{target}}(y|x)}{\pi_{\text{draft}}(y|x)}\right) + \beta_0 r(y|x)$ (Equation (1)) explicit.

   Let $\mathbf{Y}_{\text{accept}} = \{Y_i'\}_{i=1}^{N'}$ denote the subset of responses which were accepted by rejection sampling with the convention $\mathbf{Y}_{\text{accept}} = \emptyset$ if all samples were rejected.
2. **Stage IIa.** If $\mathbf{Y}_{\text{accept}} = \emptyset$, generate N fresh drafts from $\pi_{\text{target}}(\cdot|x)$, denoted $\mathbf{Y}_{\text{target}}$. Return $\texttt{SS}(\mathbf{Y}_{\text{target}}; \pi_{\text{target}}, \pi^\star_\beta)$.
3. **Stage IIb.** If $\mathbf{Y}_{\text{accept}} \neq \emptyset$, return the sample $\texttt{SS}(\mathbf{Y}_{\text{accept}}; \pi^\star_{\beta_0}, \pi^\star_\beta)$. This returns one among the responses that was accepted by soft rejection sampling.

We will choose $\beta_0 \leq \beta$. Let the policy induced by this approach be denoted $\pi_{\texttt{two-stage},\beta_1,\beta}$.

Remark D.2. $\pi_{\texttt{two-stage},\beta_1,\beta}$ is precisely the SPECS meta-algorithm instantiated for the case where the block size γ is infinite.

We will show that this two-stage process results in a subselection approach that is approximately optimal, in the sense that the resulting sample is generated from a distribution $\pi_{\texttt{two-stage},\beta_1,\beta}$ which is close to $\pi^\star_\beta$. Equivalently this may be stated as saying that the resulting policy approximately maximizes the KL-regularized expected reward (Equation (6)). We first introduce an assumption which controls the correctness of the black-box sampler $\texttt{SS}(\cdot)$.

Assumption D.3 (Consistent black-box sampler)**.** Suppose that drafts $\mathbf{Y} = \{Y_i : i \in [N']\}$ are generated by any policy π_{base}. We will assume that $N' \sim \mathsf{Poi}(\lambda)|_{>0}$, and assume that the Y_i's are sampled independently conditioned on N'. Let $\pi_{\texttt{SS}}$ be the policy corresponding to the distribution over responses induced by the subsampling procedure $\texttt{SS}(\cdot; \pi_{\text{base}}, \pi^\star_\beta)$ (marginalizing over $\mathbf{Y}$). Suppose $\pi_{\texttt{SS}}$ satisfies the following "consistency" condition,

$$\forall x \in \mathcal{X}, \quad D_f(\pi_{\texttt{SS}}(\cdot|x) \| \pi^\star_\beta(\cdot|x)) \le \varepsilon(\lambda; \pi_{\text{base}}, \pi^\star_\beta)$$

for some f-divergence D_f, and error functional $\varepsilon : \mathbb{R} \times \Pi^2 \to \mathbb{R}$ which goes to 0 as $\lambda \to \infty$.

Next we establish a black-box bound on the objective value of the two-stage sampling procedure.

Theorem D.4. *Suppose the subsampler $\texttt{SS}(\cdot; \beta)$ used satisfies Assumption D.3. Then, the two-stage sampling approach described above satisfies,*

$$\forall x \in \mathcal{X}, \quad D_f(\pi_{\texttt{two-stage},\beta_1,\beta}(\cdot|x), \pi^\star_\beta(\cdot|x)) \le \max_{p \le p^\star} \left\{ (1-p)\varepsilon(n_0, \pi^\star_{\beta_0}, \pi^\star_\beta) + p\varepsilon(n, \pi_{\text{target}}, \pi^\star_\beta) \right\} \tag{11}$$

where $n_0 = n\left(Z_{\beta_0}(x) e^{-\|S_{\beta_0}(\cdot|\cdot)\|}\right)$ and $p^\star = \left(1 - Z(x) e^{-\|S(\cdot|\cdot)\|_\infty}\right)^n$ upper bounds the probability that $\mathbf{Y}_{accept} = \emptyset$.

Proof. The proof of this result follows from two observations,

Observation 1. The analysis of rejection sampling (Theorem C.3) shows us that in the first stage of the two-stage sampling procedure, the probability that the i^{th} sample is accepted equals $Z_{\beta_0}(x) e^{-\|S_{\beta_0}(\cdot|\cdot)\|_\infty}$. Since the number of drafts N is distributed as a Poisson random variable $\sim$ Poi(n), the distribution of the number of drafts which are accepted is also Poisson, and is distributed as $N' \sim$ Poi(n') where $n' = n Z_{\beta_0}(x) e^{-\|S_{\beta_0}(\cdot|\cdot)\|_\infty}$. These samples are distributed exactly as $\pi^\star_{\beta_0}(\cdot|x)$, with no approximation error.

Observation 2. Recall that $\pi^\star_{\beta_0}(y|x) \propto \pi_{\text{target}}(y|x) e^{\beta_0 r(y|x)}$. Then, we have that,

$$\pi^\star_\beta(y|x) = \frac{\pi^\star_{\beta_0}(y|x) e^{(\beta-\beta_0) r(y|x)}}{\mathbb{E}_{Y \sim \pi^\star_{\beta_0}(\cdot|x)} \left[e^{(\beta-\beta_0) r(y|x)} \right]}.$$

The simplest intuitive explanation for this is that we may construct the tilted policy $\propto \pi_{\text{target}}(y|x) e^{\beta r(y|x)}$ by first "tilting the target policy by temperature β_0" and then subsequently tilting it by $\beta - \beta_0$.

The exactness of rejection sampling means that we may treat the N' samples which were accepted in the first stage as samples from the "base policy" $\pi_{\text{base}}(\cdot|x) = \pi^\star_{\beta_0}(\cdot|x)$. However, we must be careful about the case where $N' = 0$ and all drafts from π_{draft} were rejected. In this case, the two stage sampling procedure generates N fresh drafts from $\pi_{\text{target}}(\cdot|x)$ and subselects $\texttt{SS}(\mathbf{Y}_{\text{target}}; \pi_{\text{target}}, \pi^\star_\beta)$. The overall KL divergence of the two stage policy can be decomposed as follows. Let $\pi_{\texttt{SS},\text{reject}}$ denote the policy induced by $\texttt{SS}(\mathbf{Y}_{\text{target}}; \pi_{\text{target}}, \pi^\star_\beta)$ in the case that $N' = 0$ (and in this case $N' \sim \mathsf{Poi}(n$, and $\pi_{\texttt{SS},\text{accept}}$ denote the policy induced by $\texttt{SS}(\mathbf{Y}_{\text{accept}}; \pi^\star_{\beta_0}, \pi^\star_\beta)$ in the case that $N' > 0$. Then, by the convexity of f divergences,

$$\begin{aligned} D_f(\pi_{\texttt{two-stage},\beta_1,\beta} \| \pi^\star_\beta) &\le \Pr(N' > 0) D_f(\pi_{\texttt{SS},\text{accept}} \| \pi^\star_\beta) + \Pr(N' = 0) D_f(\pi_{\texttt{SS},\text{reject}} \| \pi^\star_\beta) \\ &\overset{(a)}{\le} \Pr(N' > 0) \varepsilon(n_0; \pi^\star_{\beta_0}, \pi^\star_\beta) + \Pr(N' = 0) \varepsilon(n; \pi_{\text{target}}, \pi^\star_\beta) \\ &\le \Pr(N' > 0) \varepsilon(n_0; \pi^\star_{\beta_0}, \pi^\star_\beta) + \Pr(N' = 0) \varepsilon(n; \pi_{\text{target}}, \pi^\star_\beta), \end{aligned}$$

where (a) relies on the the consistency property of the black-box sampler (Assumption D.3). The proof concludes by the upper bound on $\Pr(N' = 0)$ from Theorem C.3. ☐

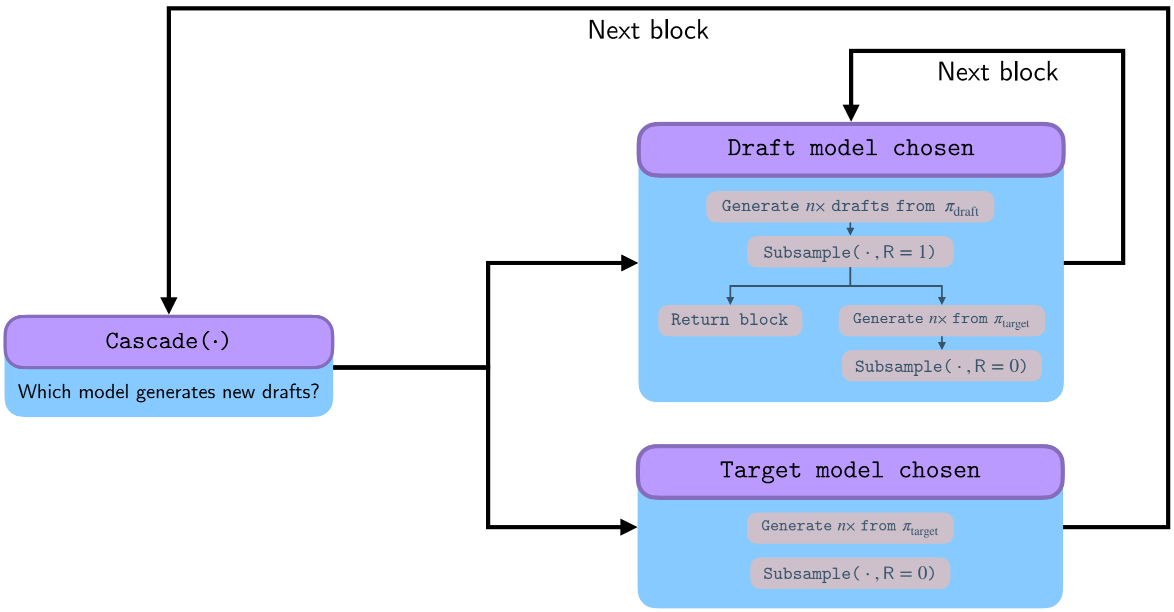

Figure 5. Flow of logic in the SPECS meta-algorithm in the finite-block regime: the description is in such a way so as to clarify the role of the CASCADE subroutine. The block corresponding to when the draft model is chosen and when the target model are executed identically to the $\gamma \to \infty$ case, with the only distinction that a PRM is used instead of an outcome reward model.

D.2. Proof of Theorem 3.3

Observe that $\pi_{\texttt{two-stage},\beta_0,\beta}$ is precisely the version of Algorithm 1, with the choice of the black-box sampler $\texttt{SS}(\cdot)$ as the sampling procedure induced by partition function approximation (Definition C.4). It is also worth mentioning again, the two changes: (a) Poisson sampling (Assumption 3.1), and (b) the use of soft rejection instead of hard rejection sampling (Assumption 3.2). Furthermore note that in Algorithm 1, we instantiate $\beta_0 = \beta/2$. Combining the guarantee on the two-stage procedure in Theorem D.4 with Corollaries C.6 and C.7 (which provide guarantees for sampling by partition function approximation),

$$D_{\chi^2}(\pi_{\texttt{two-stage},\beta_1,\beta}, \pi^\star_\beta) \le \max_{p \le p^\star} \left\{ (1-p) \cdot \exp\left(\widetilde{\mathcal{O}}_{n_0} \left(C^2_{\texttt{seq}} \frac{e^{2(\beta-\beta_0)R}}{n_0^2} \right) \right) + p \cdot \exp\left(\widetilde{\mathcal{O}} \left(\frac{e^{\beta R}}{n^2} \right) \right) \right\} - 1$$

where $n_0 = n \left(Z_{\beta_0}(x) e^{-\|S_{\beta_0}(\cdot|\cdot)\|} \right)$ and $p^\star = \left(1 - Z(x) e^{-\|S(\cdot|\cdot)\|_\infty} \right)^n$. Note that $n_0 \ge n e^{-\beta_0 R}$ and therefore,

$$D_{\chi^2}(\pi_{\texttt{two-stage},\beta_1,\beta}, \pi^\star_\beta) \le \exp\left(\widetilde{\mathcal{O}}_{n_0} \left(C^2_{\texttt{seq}} \frac{e^{2\beta R}}{n^2} \right) \right) - 1$$

Taking logarithms on both sides and noting that $D_{\mathrm{KL}}(\cdot\|\cdot) \le \log\big(1 + D_{\chi^2}(\cdot\|\cdot)\big)$ completes the proof.

E. Analysis of SPECS in the finite block-length setting: Proof of Theorem 3.7

Having worked out the analysis of SPECS in the infinite block-length setting, we next move on to analyzing the finite block-length case, corresponding to $\gamma < \infty$. In this setting, the drafts generated by the draft model are partial responses, and demarcated into "blocks". Not only is the algorithm iterative now, but control of draft generation is shared between the draft and target models additionally through the CASCADE subroutine. This is described in the flowchart in Figure 5.

The CASCADE subroutine allows for switching between π_{draft} and π_{target} while generating drafts. For the "hard" steps of a problem, it is perhaps beneficial to use the target model for draft generation, while for the easy steps, it may suffice to just use the cheaper draft model itself. While the CASCADE subroutine tries to switch between the two models most effectively, our theory will not model or be prescriptive for how best to carry out this switching. We instead focus on understanding the interaction between the PRM and the iterative nature of sampling in the finite block-length regime.

Let us first introduce the finite block-length setting more formally.

Definition E.1 (Finite block-length setting)**.** The finite block-length regime corresponds to taking $\gamma \to \infty$. Operationally, in each iteration the draft/target model generates draft blocks which are scored by a combination of the target, draft and reward models to generate the output block. The resulting partial trace may be iteratively extended using the above interaction protocol, treating the new "prompt" as this trace.

Remark E.2. In the finite block-length setting, the Poisson sampling model corresponds to letting the number of partial traces, N, which corresponds to the beam size, in each round of interaction with the draft sampling model be randomized and sampled as $N \sim \text{Poi}(n)_{>0}$.

We will reiterate the definitions of the process reward model we consider. In practice this model is trained and inexact, but for the purpose of a clean theoretical discussion we avoid this challenge and discuss the case where the process reward is exact. The following two definitions instantiate the PRM.

Definition E.3 (Optimal KL-regularized value function: restatement of Definition 3.5)**.** For a policy π, and any prompt $x \in \mathcal{X}$ and partial trace $y_{\le t}$, the optimal KL-regularized value of a policy is defined as,

$$V_\beta^\pi(x, y_{\le t}) = \frac{1}{\beta} \log \left(\mathbb{E}_{Y \sim \pi(\cdot|x, y_{\le t})} \left[e^{\beta r(x,Y)} \right] \right)$$

where Y is a full completion of the partial response $y_{\le t}$. Note that for any full response $y \in \mathcal{Y}$, $V_\beta^\pi(x, y) = r(x, y)$.

Definition E.4 (Idealized Process Reward Model: restatement of Definition 3.6)**.** Let $r_{\texttt{PRM}}^\pi$ denote the PRM corresponding to the optimal KL-regularized advantage function with respect to some policy π. Formally, given any prefix of tokens / blocks $y_{\le t-1}$ of length $t-1$ and a subsequent token / block y_t,

$$r_{\texttt{PRM}}^\pi(y_t|x, y_{\le t-1}) = V_\beta^\pi(x, y_{\le t}) - V_\beta^\pi(x, y_{\le t-1})$$

where, the KL-regularized value function V_β^π is defined in Definition 3.5. The cumulative advantage till block t is denoted, $r_{\texttt{PRM}}^\pi(y_{\le t}|x) = \sum_{t'=1}^t r_{\texttt{PRM}}^\pi(y_{t'}|x, y_{\le t'-1})$. The idealized PRM we consider is $r_{\texttt{PRM}}^{\text{target}} = r_{\texttt{PRM}}^\pi$ for $\pi = \pi_{\text{target}}$.

Our first result will be to establish that the optimal aligned model can be decomposed into a sequence of models corresponding to solving "one-block" regularized reward maximization problems over the reward function induced by $r_{\texttt{PRM}}^{\text{target}}$.

Lemma E.5. *The optimal aligned model over full responses can be written as a product of conditional per-block models,*

$$\pi_\beta^\star(y_t|x, y_{\le t-1}) = \prod_{t=1}^H \pi_\beta^\star(y_t|x, y_{\le t-1})$$

Where, we define,

$$\pi_\beta^\star(y_t|x, y_{\le t-1}) \propto \pi_{\text{target}}(y_t|x, y_{\le t-1}) e^{\beta [r_{\texttt{PRM}}^{\textit{target}}(y_t|x, y_{\le t-1})}$$

Is the solution to a one-block KL-regularized reward maximization problem. Namely, for every prompt x and partial trace $y_{\le t-1}$, $\pi_\beta^\star(\cdot|x, y_{\le t-1})$ is supported over one-block completions and is the solution to the problem,

$$\min \left\{ \mathbb{E}_{Y_t \sim \pi(\cdot|x, y_{\le t-1})} \left[r_{\texttt{PRM}}^{\textit{target}}(y_t|x, y_{t-1}) \right] - \frac{1}{\beta} D_{KL}(\pi(\cdot|x, y_{\le t-1}), \pi_{\text{target}}(\cdot|x, y_{\le t-1})) \right\}$$

where $\pi : \mathcal{X} \times \mathcal{Y}_{\le t-1} \to \Delta_{\mathcal{Y}_t}$.

Proof. By definition, the optimal aligned policy can be written down as,

$$\begin{aligned}
\pi^\star_\beta(y|x) &= \frac{\pi_{\text{target}}(y|x)e^{\beta r(y|x)}}{\mathbb{E}_{Y\sim\pi_{\text{target}}(\cdot|x)}[e^{\beta r(Y|x)}]} \\
&= \pi_{\text{target}}(y|x)e^{\beta[V_\beta^{\pi_{\text{target}}}(x,y)-V_\beta^{\pi_{\text{target}}}(x,\emptyset)]} \\
&= \pi_{\text{target}}(y|x)\prod_{t'=1}^{H} e^{\beta[V_\beta^{\pi_{\text{target}}}(x,y_{\le t'})-V_\beta^{\pi_{\text{target}}}(x,y_{\le t'-1})]} \\
&= \prod_{t=1}^{H}\frac{\pi_{\text{target}}(y_t|x,y_{\le t-1})e^{\beta[V_\beta^{\pi_{\text{target}}}(x,y_{\le t})}}{\mathbb{E}_{Y_t\sim\pi_{\text{target}}(\cdot|x,y_{\le t-1})}[e^{\beta V_\beta^{\pi_{\text{target}}}(x,[y_{\le t-1},Y_t])}]} \\
&= \prod_{t=1}^{H}\frac{\pi_{\text{target}}(y_t|x,y_{\le t-1})e^{\beta[r_{\texttt{PRM}}^{\text{target}}(y_t|x,y_{\le t-1})}}{\mathbb{E}_{Y_t\sim\pi_{\text{target}}(\cdot|x,y_{\le t-1})}[e^{\beta r_{\texttt{PRM}}^{\text{target}}(Y_t|x,y_{\le t-1})}]}.
\end{aligned}$$

□

Next, we show that policies which can guarantee that they realize a one-block distribution which is close to $\pi^\star_\beta(\cdot|x,y_{\le t-1})$ for every t correspond to policies which is close to $\pi^\star_\beta$ over full responses in the aggregate.

Lemma E.6 (Local-to-global guarantee)**.** *Consider a policy with next-block distribution given by $\pi(\cdot|x,y_{t-1})$ which satisfies the following guarantee: for every prompt x and prefix y_{t-1},*

$$D_{\chi^2}(\pi(\cdot|x,y_{\le t-1}),\pi^\star_\beta(\cdot|x,y_{\le t-1}))\le\varepsilon.$$

where the induced distributions $\pi(\cdot|x,y_{\le t-1})$ and $\pi^\star_\beta(\cdot|x,y_{\le t-1})$ are one-block completions of the partial response $(x,y_{\le t-1})$. Then, we have that,

$$1+D_{\chi^2}(\pi,\pi^\star_\beta)\le(1+\varepsilon)^H$$

Proof. By definition

$$\begin{aligned}
&1+D_{\chi^2}(\pi\|\pi^\star_\beta) \\
&=\mathbb{E}_\rho\left[\mathbb{E}_{Y\sim\pi^\star_\beta(\cdot|x)}\left[\left(\frac{\pi(Y|x)}{\pi^\star_\beta(Y|x)}\right)^2\right]\right] \\
&=\mathbb{E}_\rho\left[\mathbb{E}_{Y\sim\pi^\star_\beta(\cdot|x)}\left[\prod_{t=1}^{H}\left(\frac{\pi(Y_t|x,Y_{\le t-1})}{\pi^\star_\beta(Y_t|x,Y_{\le t-1})}\right)^2\right]\right] \\
&=\mathbb{E}_\rho\left[\mathbb{E}_{Y_{\le H-1}\sim\pi^\star_\beta(\cdot|x)}\left[[1+D_{\chi^2}(\pi(\cdot|x,Y_{\le H-1})\|\pi^\star_\beta(\cdot|x,Y_{\le H-1}))]\cdot\prod_{t=1}^{H-1}\left(\frac{\pi(Y_t|x,Y_{\le t-1})}{\pi^\star_\beta(Y_t|x,Y_{\le t-1})}\right)^2\right]\right] \\
&\overset{(a)}{\le}(1+\varepsilon)\mathbb{E}_\rho\left[\mathbb{E}_{Y_{\le H-1}\sim\pi^\star_\beta(\cdot|x)}\left[\prod_{t=1}^{H-1}\left(\frac{\pi(Y_t|x,Y_{\le t-1})}{\pi^\star_\beta(Y_t|x,Y_{\le t-1})}\right)^2\right]\right],
\end{aligned}$$

where (a) follows from the local assumption on π. By recursing on the RHS, we arrive at the statement of the lemma. □

This results in a guarantee for SPECS in the finite-block length setting where the algorithm scores responses using a PRM.

E.1. Proof of Theorem 3.7

Consider some partial trace $y_{\le t-1}$ and let us consider the process of generating the t^{th} block. Let $Z_t\in\{0,1\}$ denote the random variable which is a measurable function of $y_{\le t-1}$ which determines whether π_{draft} generates the drafts or π_{target}

generates drafts (i.e., $Z_t = 1$ means that iteration t begins at Step I of Algorithm 1, while $Z_t = 0$ means that it begins at Step III).

By the observation in Remark C.8 for the infinite block-length instantiation of the SPECS policy $\pi_{\texttt{SPECS}}$, the following local guarantee is also true when instantiated with $r = r_{\texttt{PRM}}^{\text{target}}$ and used to generate a single block: given any prompt x and partial trace $y_{\le t-1}$,

$$D_{\chi^2}(\pi_{\texttt{SPECS}}(\cdot|x, y_{\le t-1})|\{Z_t = 0\} \| \pi^\star_\beta(\cdot|x, y_{\le t-1})) \le \exp\left(\widetilde{\mathcal{O}}_n\left(\frac{e^{2\beta R}}{n^2}\right)\right) - 1$$
$$D_{\chi^2}(\pi_{\texttt{SPECS}}(\cdot|x, y_{\le t-1})|\{Z_t = 1\} \| \pi^\star_\beta(\cdot|x, y_{\le t-1})) \le \exp\left(\widetilde{\mathcal{O}}_n\left(C_{\texttt{block}}^2\frac{e^{2\beta R}}{n^2}\right)\right) - 1$$

Note in particular that the second guarantee only depends on $C_{\texttt{block}}$ by virtue of the fact that we only generate a single block and not entire responses ($\pi_{\texttt{SPECS}}(\cdot|x, y_{t-1})$ is a distribution over single blocks). By the convexity of the χ^2-divergence, we have that $\pi_{\texttt{SPECS}}(\cdot|x, y_{\le t-1})$, which is a mixture between $\pi_{\texttt{SPECS}}(\cdot|x, y_{\le t-1})|\{Z_t = 0\}$ and $\pi_{\texttt{SPECS}}(\cdot|x, y_{\le t-1})|\{Z_t = 1\}$ satisfies,

$$D_{\chi^2}(\pi_{\texttt{SPECS}}(\cdot|x, y_{\le t-1}) \| \pi^\star_\beta(\cdot|x, y_{\le t-1})) \le \exp\left(\widetilde{\mathcal{O}}_n\left(C_{\texttt{block}}^2\frac{e^{2\beta R}}{n^2}\right)\right) - 1$$

The proof finishes by combining these guarantees across t via Lemma E.6 to give the inequality,

$$1 + D_{\chi^2}(\pi_{\texttt{SPECS}} \| \pi^\star_\beta) \le \exp\left(\widetilde{\mathcal{O}}_n\left(H \cdot C_{\texttt{block}}^2\frac{e^{2\beta R}}{n^2}\right)\right)$$

Using the inequality, $D_{\mathrm{KL}}(\pi_{\texttt{SPECS}} \| \pi^\star_\beta) \le \log\Big(1 + D_{\chi^2}(\pi_{\texttt{SPECS}} \| \pi^\star_\beta)\Big)$ completes the proof.

F. Sampling by partition function approximation: Proof of Theorem C.5 and Theorem C.9

We begin with the proof Theorem C.5; the last part of this section will be dedicated toward proving Theorem C.9. Prior to diving into the proof, we will first introduce some additional notation. Let N_y denote the number of occurrences of a particular $y \in \mathcal{Y}$ in the collection of $N \sim \mathsf{Poi}(n)_{>0}$ responses, $\mathbf{Y}_{\text{draft}}$, drawn i.i.d. from $\pi_{\text{draft}}(\cdot|x)$. Recall the definition of the score function $\phi_\beta(\cdot|\cdot)$ as defined in Definition C.1. Furthermore, define,

$$\Phi_\beta(s) = \mathbb{E}_{Y \sim \pi_{\text{draft}}(\cdot|x)}\left[e^{-s\,\phi_\beta(Y|x)}\right]$$

as the Laplace transform of $\phi_\beta(Y|x)$ for $Y \sim \pi_{\text{draft}}(\cdot|x)$. For the sake of brevity, we will abbreviate $\pi_{\texttt{part-apx},\beta,n}$ by $\pi_{\beta,n}$. Observe that,

$$\pi_{\beta,n}(y|x) = \frac{1}{\Pr(N_0 \ge 1)}\mathbb{E}\left[\frac{\mathbb{I}(N_0 \ge 1) \cdot N_y \phi_\beta(y|x)}{Z_{\mathbf{Y}_{\text{draft}},\beta}(x)}\right] \tag{12}$$

where $N_0 \sim \mathsf{Poi}(n)$.

Note that with our definition of scores (Definition C.1), we have,

$$\pi^\star_\beta(y|x) = \frac{\pi_{\text{target}}(y|x)e^{\beta r(y|x)}}{\mathbb{E}_{Y \sim \pi_{\text{target}}(\cdot|x)}[e^{\beta r(Y|x)}]} = \frac{\pi_{\text{draft}}(y|x)\phi_\beta(y|x)}{\mathbb{E}_{Y \sim \pi_{\text{draft}}(\cdot|x)}[\phi_\beta(Y|x)]} = \frac{\pi_{\text{draft}}(y|x)\phi_\beta(y|x)}{-\Phi'_\beta(0)}.$$

Our first result will be to provide a formula for the probability that a response y is sampled by $\pi_{\texttt{part-apx},\beta,n}$ marginalized over the underlying set of drafts generated, $\mathbf{Y}_{\text{draft}}$.

Lemma F.1. *Under the Poisson sampling model (Assumption 3.1), $\pi_{\beta,n}$ can be written as the following integral,*

$$\pi_{\beta,n}(y|x) = \frac{n\pi_{\text{draft}}(y|x)\phi_\beta(y|x)}{1 - e^{-n}}\int_0^\infty e^{n(\Phi_\beta(s)-1)} \cdot e^{-s\phi_\beta(y|x)}\mathrm{d}s.$$

Proof. Our first step we will be to arrive at an integral representation of $1/Z_{\mathbf{Y}_{\text{draft}},\beta}(x)$. For any $A > 0$, $A^{-1} = \int_0^\infty e^{-sA}\mathrm{d}s$. With the choice $A = Z_{\mathbf{Y}_{\text{draft}},\beta}(x)$, by Fubini's theorem,

$$\mathbb{E}\left[\frac{\mathbb{I}(N_0 \geq 1) \cdot N_y \phi_\beta(y|x)}{Z_{\mathbf{Y}_{\text{draft}},\beta}(x)}\right] = \phi_\beta(y|x) \int_0^\infty \mathbb{E}\left[N_y e^{-s\, Z_{\mathbf{Y}_{\text{draft}},\beta}(x)}\, \mathbb{I}(N \geq 1)\right] \mathrm{d}s = \int_0^\infty \Psi_y(s)\, \mathrm{d}s \tag{13}$$

where we define,

$$\Psi_y(s) = \phi_\beta(y|x)\mathbb{E}\left[N_y e^{-s\, Z_{\mathbf{Y}_{\text{draft}},\beta}(x)}\, \mathbb{I}(N_0 \geq 1)\right] = \phi_\beta(y|x)\mathbb{E}\left[N_y e^{-s\, Z_{\mathbf{Y}_{\text{draft}},\beta}(x)}\right],$$

which uses the fact that $N_y = 0$ if $N_0 = 0$. The remainder of this proof will be to analyze $\Psi_y(s)$.
Define $N_y = \sum_{y' \in \mathbf{Y}_{\text{draft}}} \mathbb{I}(y' = y)$ as the number of times y appears in $\mathbf{Y}_{\text{draft}}$. Then,

$$\begin{aligned}
\Psi_y(s) &= \mathbb{E}\left[N_y e^{-s\, \sum_{y_0 \in \mathcal{Y}} N_{y_0} \phi_\beta(y_0|x)}\right] \\
&= \mathbb{E}\left[N_y e^{-sN_y \phi_\beta(y|x)}\right] \cdot \prod_{y_0 \in \mathcal{Y} \setminus \{y\}} \mathbb{E}\left[e^{-sN_{y_0} \phi_\beta(y_0|x)}\right]
\end{aligned}$$

where the second equation uses independence admitted by the Poisson structure. Note that,

$$\begin{aligned}
\mathbb{E}\left[e^{-sN_{y_0} \phi_\beta(y_0|x)}\right] &= \sum_{j=0}^\infty e^{-sj\phi_\beta(y_0|x)} \Pr[N_y = j] \\
&= \sum_{j=0}^\infty e^{-sj\phi_\beta(y_0|x)} \cdot \frac{(n\pi_{\text{draft}}(y_0|x))^j}{j!} e^{-n\pi_{\text{draft}}(y_0|x)} \\
&= \exp\Big(n\pi_{\text{draft}}(y_0|x)\big(e^{-s\phi_\beta(y_0|x)} - 1\big)\Big)
\end{aligned}$$

A similar calculation results in,

$$\begin{aligned}
\mathbb{E}\left[N_y e^{-sN_y \phi_\beta(y|x)}\right] &= \sum_{j=0}^\infty j e^{-sj\phi_\beta(y|x)} \Pr[N_y = j] \\
&= \sum_{j=1}^\infty e^{-sj\phi_\beta(y|x)} \cdot \frac{(n\pi_{\text{draft}}(y|x))^j}{(j-1)!} e^{-n\pi_{\text{draft}}(y|x)} \\
&= n\pi_{\text{draft}}(y|x) e^{-s\phi_\beta(y|x)} \sum_{j=0}^\infty \cdot \frac{\big(n\pi_{\text{draft}}(y|x) e^{-s\phi_\beta(y|x)}\big)^j}{j!} e^{-n\pi_{\text{draft}}(y|x)} \\
&= n\pi_{\text{draft}}(y|x) e^{-s\phi_\beta(y|x)} \exp\Big(n\pi_{\text{draft}}(y|x)\big(e^{-s\phi_\beta(y|x)} - 1\big)\Big)
\end{aligned}$$

Multiplying everything out, we get,

$$\begin{aligned}
\Psi_y(s) &= n\pi_{\text{draft}}(y|x) e^{-s\phi_\beta(y|x)} \exp\Big(n\pi_{\text{draft}}(y|x)\big(e^{-s\phi_\beta(y|x)} - 1\big)\Big) \cdot \prod_{y_0 \in \mathcal{Y} \setminus \{y\}} \exp\Big(n\pi_{\text{draft}}(y_0|x)\big(e^{-s\phi_\beta(y_0|x)} - 1\big)\Big) \\
&= n\pi_{\text{draft}}(y|x) e^{-s\phi_\beta(y|x)} e^{n(\Phi_\beta(s) - 1)}
\end{aligned}$$

Plugging into Equation (13) completes the proof. ∎

As a sanity check, we can check that the probability calculated in Lemma F.1 integrates to 1 over $y \in \mathcal{Y}$. Notably,

$$\begin{aligned}
\sum_{y \in \mathcal{Y}} \pi_{\beta,n}(y|x) &= \sum_{y \in \mathcal{Y}} \frac{n \pi_{\text{draft}}(y|x) \phi_\beta(y|x)}{1 - e^{-n}} \int_0^\infty e^{n(\Phi_\beta(s)-1)} \cdot e^{-s\phi_\beta(y|x)} \mathrm{d}s \\
&= \frac{n}{1-e^{-n}} \int_0^\infty e^{n(\Phi_\beta(s)-1)} \cdot \mathbb{E}_{Y \sim \pi_{\text{draft}}(\cdot|x)}[\phi_\beta(Y|x) e^{-s\phi_\beta(Y|x)}] \mathrm{d}s \\
&= \frac{-n}{1-e^{-n}} \int_0^\infty e^{n(\Phi_\beta(s)-1)} \cdot \Phi'_\beta(s) \mathrm{d}s \\
&\overset{(a)}{=} \frac{1}{1-e^{-n}} \int_0^n e^{-t} \cdot \mathrm{d}t \\
&= 1
\end{aligned}$$

where in (a), $t = n(1 - \Phi_\beta(s))$.

F.1. Bounding the KL-/χ^2-divergence: Proof of Theorem C.5

We bound $D_{\mathrm{KL}}(\pi_{\beta,n} \| \pi^\star_\beta) \le \log\Big(1 + D_{\chi^2}(\pi_{\beta,n} \| \pi^\star_\beta)\Big)$ and analyze the latter. Observe that,

$$\begin{aligned}
&1 + D_{\chi^2}(\pi_{\beta,n} \| \pi^\star_\beta) && (14)\\
&= \sum_{y \in \mathcal{Y}} \frac{(\pi_{\beta,n}(y|x))^2}{\pi^\star_\beta(y|x)} \\
&= \sum_{y \in \mathcal{Y}} \frac{\pi_{\text{draft}}(y|x) \phi_\beta(y|x) n^2 (-\Phi'_\beta(0))}{(1-e^{-n})^2} \left[\int_0^\infty e^{n(\Phi_\beta(s)-1)} \cdot e^{-s\phi_\beta(y|x)} \mathrm{d}s \right]^2 \\
&= \sum_{y \in \mathcal{Y}} \frac{\pi_{\text{draft}}(y|x) \phi_\beta(y|x) n^2 (-\Phi'_\beta(0))}{(1-e^{-n})^2} \int_0^\infty \int_0^\infty e^{n(\Phi_\beta(s_1)-1) + n(\Phi_\beta(s_2)-1)} \cdot e^{-(s_1+s_2)\phi_\beta(y|x)} \mathrm{d}s_1 \mathrm{d}s_2 \\
&= \frac{n^2(-\Phi'_\beta(0))}{(1-e^{-n})^2} \int_0^\infty \int_0^\infty e^{n(\Phi_\beta(s_1)-1) + n(\Phi_\beta(s_2)-1)} \cdot (-\Phi'_\beta(s_1+s_2)) \mathrm{d}s_1 \mathrm{d}s_2 && (15)
\end{aligned}$$

We split the integral into four components; denoting $p_n = \log\big(n^3/4\big)/n$,

$$\begin{aligned}
\mathcal{R}_{0,0} &= \{(s_1,s_2) \in \mathbb{R}^2 : \Phi_\beta(s_1), \Phi_\beta(s_2) \ge 1 - p_n\} \\
\mathcal{R}_{1,0} &= \{(s_1,s_2) \in \mathbb{R}^2 : \Phi_\beta(s_1) \le 1 - p_n, \text{ and } \Phi_\beta(s_2) \ge 1 - p_n\} \\
\mathcal{R}_{0,1} &= \{(s_1,s_2) \in \mathbb{R}^2 : \Phi_\beta(s_1) \ge 1 - p_n, \text{ and } \Phi_\beta(s_2) \le 1 - p_n\} \\
\mathcal{R}_{1,1} &= \{(s_1,s_2) \in \mathbb{R}^2 : \Phi_\beta(s_1), \Phi_\beta(s_2) \le 1 - p_n\}.
\end{aligned}$$

Correspondingly, for $(i,j) \in \{0,1\}^2$, we will let,

$$\mathcal{I}_{i,j} = \frac{n^2(-\Phi'_\beta(0))}{(1-e^{-n})^2} \int_{\mathcal{R}_{i,j}} e^{n(\Phi_\beta(s_1)-1) + n(\Phi_\beta(s_2)-1)} \cdot (-\Phi'_\beta(s_1+s_2)) \mathrm{d}s_1 \mathrm{d}s_2.$$

And note that $\mathcal{I}_{0,1} = \mathcal{I}_{1,0}$.

Lemma F.2 (Bounding the integral $\mathcal{I}_{0,0}$)**.** *Assume that n is sufficiently large that $0 \le p_n \le \frac{1}{\sqrt{2}}$. Then,*

$$\mathcal{I}_{0,0} \le \exp\left(p_n^2 \left(\phi_{\max}/\phi_{\min} - 1\right)^2\right).$$

Recall here, that $\phi_{\max} \triangleq \max_{x \in \mathcal{X}, y \in \mathcal{Y}} \phi_\beta(y|x)$ and $\phi_{\min} \triangleq \min_{x \in \mathcal{X}, y \in \mathcal{Y}} \phi_\beta(y|x)$.

Proof. Observe that $\mathcal{I}_{0,0}$ corresponds to integrating around a small neighborhood of $(0,0)$, and to this end, we first bound $(-\Phi'_\beta(0))(-\Phi'_\beta(s_1+s_2))$ for $(s_1,s_2) \in \mathcal{R}_{0,0}$, showing that it is approximately equal to $(-\Phi'_\beta(s_1))(-\Phi'_\beta(s_2))$ in this regime. As s_1 and s_2 become smaller, which is the case as n grows, the approximation becomes better: this essentially follows from the fact that $f(s) = \log\big(-\Phi_\beta'(s)\big)$ is a bounded, smooth convex function and behaves locally linearly. Formally,

$$\begin{aligned}
f(s_1+s_2)+f(0)-f(s_1)-f(s_2) &= \int_0^{s_1}\int_0^{s_2} f''(u_1+u_2)\mathrm{d}u_1\mathrm{d}u_2 \\
&\overset{(a)}{=} \int_0^{s_1}\int_0^{s_2} \frac{\Phi_\beta'''(u)\Phi_\beta'(u)-(\Phi_\beta''(u))^2}{(\Phi_\beta'(u))^2}\mathrm{d}u_1\mathrm{d}u_2 \\
&= \int_0^{s_1}\int_0^{s_2} \frac{\Phi_\beta'''(u)\Phi_\beta'(u)-(\Phi_\beta''(u))^2}{(\Phi_\beta'(u))^2\Phi_\beta'(u_1)\Phi_\beta'(u_2)}\mathrm{d}\Phi_\beta(u_1)\mathrm{d}\Phi_\beta(u_2) \\
&\le \max_{u_1\in[0,s_1]}\max_{u_2\in[0,s_2]} \frac{\Phi_\beta'''(u)\Phi_\beta'(u)-(\Phi_\beta''(u))^2}{(\Phi_\beta'(u))^2\Phi_\beta'(u_1)\Phi_\beta'(u_2)} \cdot \int_0^{s_1}\int_0^{s_2} \mathrm{d}\Phi_\beta(u_1)\mathrm{d}\Phi_\beta(u_2) \\
&\le p_n^2 \cdot \max_{u_1\in[0,s_1]}\max_{u_2\in[0,s_2]} \frac{\Phi_\beta'''(u)\Phi_\beta'(u)-(\Phi_\beta''(u))^2}{(\Phi_\beta'(u))^2\Phi_\beta'(u_1)\Phi_\beta'(u_2)}
\end{aligned} \tag{16}$$

where in (a), $u = u_1+u_2$. In the $\mathcal{R}_{0,0}$ regime, letting $Y, Y' \overset{\text{i.i.d.}}{\sim} \pi_{\text{draft}}(\cdot|x)$,

$$\begin{aligned}
&\max_{u_1\in[0,s_1]}\max_{u_2\in[0,s_2]} \frac{\Phi_\beta'''(u)\Phi_\beta'(u)-(\Phi_\beta''(u))^2}{(\Phi_\beta'(u)^2\Phi_\beta'(u_1)\Phi_\beta'(u_2)} \\
&= \frac{\mathbb{E}\big[\phi_\beta(Y|x)\phi_\beta(Y'|x)\cdot(\phi_\beta(Y|x)-\phi_\beta(Y'|x))^2\cdot e^{-u(\phi_\beta(Y|x)+\phi_\beta(Y'|x))}\big]}{2\mathbb{E}\big[\phi_\beta(Y|x)\cdot e^{-u(\phi_\beta(Y|x)}\big]^2\cdot\mathbb{E}\big[\phi_\beta(Y|x)\phi_\beta(Y'|x)\cdot e^{-u(\phi_\beta(Y|x)+\phi_\beta(Y'|x))}\big]} \\
&\overset{(a)}{\le} \frac{\mathbb{E}\big[\phi_\beta(Y|x)\phi_\beta(Y'|x)\cdot(\phi_\beta(Y|x)-\phi_\beta(Y'|x))^2\cdot e^{-u(\phi_\beta(Y|x)+\phi_\beta(Y'|x))}\big]}{2\phi_{\min}^2(1-p_n)^2\cdot\mathbb{E}\big[\phi_\beta(Y|x)\phi_\beta(Y'|x)\cdot e^{-u(\phi_\beta(Y|x)+\phi_\beta(Y'|x))}\big]} \\
&\le \frac{(\phi_{\max}-\phi_{\min})^2}{2\phi_{\min}^2(1-p_n)^2} \\
&\le (\phi_{\max}/\phi_{\min}-1)^2,
\end{aligned}$$

where (a) uses the assumption that $\Phi_\beta(u_1), \Phi_\beta(u_2) \ge 1-p_n$ in this regime, and the last inequality assumes that n is sufficiently large that $p_n \le \frac{1}{\sqrt{2}}$. Combining with Equation (16),

$$\frac{(-\Phi_\beta'(s_1+s_2))(-\Phi_\beta'(0))}{(-\Phi_\beta'(s_1))(-\Phi_\beta'(s_2))} \le \exp\left(p_n^2\left(\frac{\phi_{\max}}{\phi_{\min}}-1\right)^2\right).$$

This results in the bound,

$$\begin{aligned}
\mathcal{I}_{0,0} &= \frac{n^2(-\Phi_\beta'(0))}{(1-e^{-n})^2}\int_{\mathcal{R}_{0,0}} e^{n(\Phi_\beta(s_1)-1)+n(\Phi_\beta(s_2)-1)}\cdot(-\Phi_\beta'(s_1+s_2))\mathrm{d}s_1\mathrm{d}s_2 \\
&\le \exp\left(p_n^2\left(\frac{\phi_{\max}}{\phi_{\min}}-1\right)^2\right)\cdot\frac{n^2}{(1-e^{-n})^2}\int_{\mathcal{R}_{0,0}} e^{n(\Phi_\beta(s_1)-1)+n(\Phi_\beta(s_2)-1)}\cdot(-\Phi_\beta'(s_1))(-\Phi_\beta'(s_2))\mathrm{d}s_1\mathrm{d}s_2 \\
&\overset{(a)}{\le} \exp\left(p_n^2\left(\frac{\phi_{\max}}{\phi_{\min}}-1\right)^2\right)\cdot\frac{1}{(1-e^{-n})^2}\int_0^{\log(n^3/4)}\int_0^{\log(n^3/4)} e^{-t_1}e^{-t_2}\mathrm{d}t_1\mathrm{d}t_2 \\
&\le \exp\left(p_n^2\left(\frac{\phi_{\max}}{\phi_{\min}}-1\right)^2\right),
\end{aligned}$$

where in (a) we substitute $t_i = n(1-\Phi_\beta(s_i))$ for $i \in \{1,2\}$, and use the definition of $\mathcal{R}_{0,0}$, and in the last inequality, the fact that $1-4/n^3 \le 1-e^{-n}$. □

Lemma F.3 (Bounding $\mathcal{I}_{1,0}$ and $\mathcal{I}_{0,1}$)**.** *Suppose n is sufficiently large that $e^{-n} \leq 1/10$. Then,*

$$\mathcal{I}_{1,0} = \mathcal{I}_{0,1} \leq \frac{\phi_{\max}}{\phi_{\min}} \cdot \frac{5}{n^2}$$

Proof. Note that $\mathcal{I}_{1,0} = \mathcal{I}_{0,1}$ by symmetry, and so we will bound the former. Since $n(1 - \Phi_\beta(s_1)) \geq \log(n^3/4)$ in the $\mathcal{R}_{1,0}$ regime,

$$\begin{aligned}
\mathcal{I}_{1,0} &= \frac{n^2(-\Phi'_\beta(0))}{(1-e^{-n})^2} \int_{\mathcal{R}_{1,0}} e^{n(\Phi_\beta(s_1)-1)+n(\Phi_\beta(s_2)-1)} \cdot (-\Phi'_\beta(s_1+s_2)) \mathrm{d}s_1 \mathrm{d}s_2 \\
&\leq \frac{4(-\Phi'_\beta(0))}{n(1-e^{-n})^2} \int_{\mathcal{R}_{1,0}} e^{n(\Phi_\beta(s_2)-1)} \cdot (-\Phi'_\beta(s_1+s_2)) \mathrm{d}s_1 \mathrm{d}s_2 \\
&= \frac{4(-\Phi'_\beta(0))}{n(1-e^{-n})^2} \int_{\mathcal{R}_{1,0}} e^{n(\Phi_\beta(s_2)-1)} \cdot \mathbb{E}\big[\phi_\beta(Y|x) e^{-(s_1+s_2)\phi_\beta(Y|x)}\big] \mathrm{d}s_1 \mathrm{d}s_2 \\
&\overset{(a)}{\leq} \frac{4(-\Phi'_\beta(0))}{n(1-e^{-n})^2} \int_{\mathcal{R}_{1,0}} e^{n(\Phi_\beta(s_2)-1)} \cdot \mathbb{E}\big[e^{-s_2\phi_\beta(Y|x)}\big] \mathrm{d}s_2 \\
&\leq \frac{\phi_{\max}}{\phi_{\min}} \cdot \frac{4}{n(1-e^{-n})^2} \int_{\mathcal{R}_{1,0}} e^{n(\Phi_\beta(s_2)-1)} \cdot \mathbb{E}\big[\phi_\beta(Y|x) e^{-s_2\phi_\beta(Y|x)}\big] \mathrm{d}s_2 \\
&\leq \frac{\phi_{\max}}{\phi_{\min}} \cdot \frac{4}{n^2(1-e^{-n})^2}
\end{aligned}$$

where in (a), we use the fact that $\int_{\geq 0} e^{-sA} \mathrm{d}s = A^{-1}$, and in the last equation, the substitution $t = n(1 - \Phi_\beta(s_2))$, with $\mathrm{d}t = n\mathbb{E}\big[\phi_\beta(Y|x) e^{-s_2\phi_\beta(Y|x)}\big] \mathrm{d}s_2$. Assuming that n is sufficiently large that $1 - e^{-n} \geq 9/10$ completes the proof. □

Lemma F.4 (Bounding the integral $\mathcal{I}_{1,1}$)**.** *Suppose n is sufficiently large that $e^{-n} \leq 1/10$. Then,*

$$\mathcal{I}_{1,1} \leq \frac{\phi_{\max}}{\phi_{\min}} \cdot \frac{20}{n^4}.$$

Proof. By definition,

$$\begin{aligned}
\mathcal{I}_{1,1} &= \frac{n^2(-\Phi'_\beta(0))}{(1-e^{-n})^2} \int_{\mathcal{R}_{1,1}} e^{n(\Phi_\beta(s_1)-1)+n(\Phi_\beta(s_2)-1)} \cdot (-\Phi'_\beta(s_1+s_2)) \mathrm{d}s_1 \mathrm{d}s_2 \\
&\overset{(a)}{\leq} \frac{16n^2(-\Phi'_\beta(0))}{n^6(1-e^{-n})^2} \int_{\mathcal{R}_{1,1}} \mathbb{E}[\phi_\beta(Y|x) e^{-(s_1+s_2)\phi_\beta(Y|x)}] \mathrm{d}s_1 \mathrm{d}s_2 \\
&\overset{(b)}{\leq} \frac{20(-\Phi'_\beta(0))}{n^4} \int_{\mathcal{R}_{1,1}} \mathbb{E}[\phi_\beta(Y|x) e^{-(s_1+s_2)\phi_\beta(Y|x)}] \mathrm{d}s_1 \mathrm{d}s_2 \\
&\overset{(c)}{\leq} \frac{20(-\Phi'_\beta(0))}{n^4} \mathbb{E}\left[(\phi_\beta(Y|x))^{-1}\right] \\
&\leq \frac{\phi_{\max}}{\phi_{\min}} \cdot \frac{20}{n^4},
\end{aligned}$$

where (a) uses the definition of $\mathcal{R}_{1,1}$, (b) uses the fact that $1 - e^{-n}$ is at least $9/10$, and (c) uses the fact that $\int_{\geq 0} e^{-sA} \mathrm{d}s = A^{-1}$. □

F.2. Proof of Theorem C.5

Proof. This is a consequence of the fact that by Equation (15),

$$1 + D_{\chi^2}(\pi_{\beta,n} \| \pi^\star_\beta) = \sum_{(i,j)\in\{0,1\}^2} \mathcal{I}_{i,j}$$

and the bounds on $\mathcal{I}_{i,j}$ for different values of i and j established in Lemmas F.2 to F.4. It's worth pointing out that when $n \geq 3$, $1 - e^{-n} \geq 9/10$ and $p_n \leq 1/\sqrt{2}$. This results in the bound,

$$1 + D_{\chi^2}(\pi_{\beta,n} \| \pi^\star_\beta) \leq \frac{\phi_{\max}}{\phi_{\min}} \cdot \frac{13}{n^2} + \exp\left(p_n^2 \left(\frac{\phi_{\max}}{\phi_{\min}} - 1\right)^2\right).$$

Simplifying this by noting that $p_n = \log\big(n^3/4\big)/n \geq 1.9/n$ and $z \leq e^z - 1$, gives the inequality,

$$1 + D_{\chi^2}(\pi_{\beta,n} \| \pi^\star_\beta) \leq 1 + 2\left(\exp\left(4p_n^2 \left(\frac{\phi_{\max}}{\phi_{\min}}\right)^2\right) - 1\right).$$

Taking a logarithm on both sides, noting that $D_{\mathrm{KL}}(p\|q) \leq \log\big(1 + D_{\chi^2}(p\|q)\big)$ and simplifying the RHS with the inequality, $\log(1 + c(e^x - 1)) \leq cx$ for $c \geq 1$ completes the proof. □

F.3. Proof of Theorem C.9

Consider the following problem instance where there is a single prompt (which we will avoid stating in the reward and policy notations), and the response space $\mathcal{Y} = \{y_0, y_1\}$. $r(y_0) = 0$ and $r(y_1) = R$, and $1 - \pi_{\text{target}}(y_0) = \pi_{\text{target}}(y_1) = \frac{1}{1+\theta}$ where we denote $\theta = e^{\beta R}$ for the sake of brevity. The optimal aligned model is,

$$\pi^\star_{w,\beta}(y_0) = \pi^\star_{w,\beta}(y_1) = \frac{1}{2}$$

For this model, the Laplace transform of the score function is,

$$\Phi_\beta(s) = \mathbb{E}_{Y \sim \pi_{\text{target}}(\cdot|x)}[e^{-s\phi_\beta(Y|x)}] = \frac{\theta e^{-s} + e^{-s\theta}}{1+\theta} \tag{17}$$

From Lemma F.1,

$$\begin{aligned}
\pi_{\beta,n}(y_0) &= \frac{\theta}{1+\theta} \cdot \frac{n}{(1-e^{-n})} \int_0^\infty e^{n(\Phi_\beta(s)-1)} \cdot e^{-s} \mathrm{d}s \\
&= \frac{\theta}{1+\theta} \cdot \frac{n}{(1-e^{-n})} \int_0^1 e^{n\left(\frac{\theta t + t^\theta}{1+\theta} - 1\right)} \mathrm{d}t
\end{aligned}$$

where the last equation uses the structure of the Laplace transform of the score function in Equation (17). Let $h = g(t) = \frac{\theta t + t^\theta}{1+\theta}$, and thereby, $t = g^{-1}(h)$ and $\mathrm{d}t = 1/g'(g^{-1}(h))\mathrm{d}h$.

$$\begin{aligned}
\pi_{\beta,n}(y_0) &= \frac{\theta}{1+\theta} \cdot \frac{n}{(1-e^{-n})} \int_0^1 \frac{e^{n(h-1)}}{g'(g^{-1}(h))} \mathrm{d}h \\
&\overset{(b)}{=} \frac{\theta}{1+\theta} \cdot \frac{1}{(1-e^{-n})} \int_0^n \frac{e^{-h_1}}{g'(g^{-1}(1-h_1/n))} \mathrm{d}h_1
\end{aligned}$$

where in (b) we plug in $h_1 = n(1-h)$. Observe that $g'(t) = \frac{\theta + \theta t^{\theta-1}}{1+\theta} \leq \frac{2\theta}{1+\theta}$ is an increasing function in t, and thereby,

$$\pi_{\beta,n}(y_0) \geq \frac{1}{2}.$$

In particular, we will show that $\left|\pi_{\beta,n}(y_0) - \frac{1}{2}\right| \geq \mathcal{O}(\frac{\theta}{n})$. Since g' and g^{-1} are increasing functions,

$$\begin{aligned}
\pi_{\beta,n}(y_0) &\geq \frac{\theta}{1+\theta} \cdot \frac{1}{(1-e^{-n})} \left[\int_0^1 \frac{e^{-h_1}}{g'(g^{-1}(1))} \mathrm{d}h_1 + \int_1^n \frac{e^{-h_1}}{g'(g^{-1}(1-1/n))} \mathrm{d}h_1\right] \\
&= \frac{\theta}{1+\theta} \cdot \frac{1}{(1-e^{-n})} \left[(1-e^{-1}) \frac{1+\theta}{2\theta} + \frac{e^{-1} - e^{-n}}{2\theta/(1+\theta)} \cdot \frac{2\theta/(1+\theta)}{g'(g^{-1}(1-1/n))}\right] \\
&= \frac{1}{2} + \frac{e^{-1} - e^{-n}}{2} \left(\frac{2\theta/(1+\theta)}{g'(g^{-1}(1-1/n))} - 1\right)
\end{aligned} \tag{18}$$

Observe that,

$$
\begin{aligned}
g\left(1-\frac{1+\theta}{2\theta n}\right) &= \frac{\theta}{1+\theta}\left(1-\frac{1+\theta}{2\theta n}\right)+\frac{1}{1+\theta}\left(1-\frac{1+\theta}{2\theta n}\right)^{\theta} \\
&\geq \frac{\theta}{1+\theta}\left(1-\frac{1+\theta}{2\theta n}\right)+\frac{1}{1+\theta}\left(1-\frac{1+\theta}{2n}\right) \\
&= 1-\frac{1}{n}
\end{aligned}
$$

Since g (and thereby g^{-1}) is increasing, this results in the inequality, $g^{-1}(1-1/n) \leq 1-\frac{1+\theta}{2\theta n}$. Since g' is also increasing, we have the inequality,

$$
\begin{aligned}
g'(g^{-1}(1-1/n)) &\leq g'\left(1-\frac{1+\theta}{2\theta n}\right) \\
&= \frac{\theta}{1+\theta}+\frac{\theta}{1+\theta}\left(1-\frac{1+\theta}{2\theta n}\right)^{\theta-1} \\
&\leq \frac{2\theta}{1+\theta}-\frac{\theta}{1+\theta}\left[1-\left(1-\frac{1+\theta}{2\theta n}\right)^{\theta-1}\right]
\end{aligned}
$$

Assuming that $n \geq \theta \geq 2$, $\left(1-\frac{1+\theta}{2\theta n}\right)^{\theta-1} \geq 1-\frac{1+\theta}{4n}$. This implies,

$$
g'(g^{-1}(1-1/n)) \leq \frac{2\theta}{1+\theta}-\frac{\theta}{4n}.
$$

Combining with Equation (18),

$$
\begin{aligned}
\pi_{\beta,n}(y_0)-\frac{1}{2} &\geq \frac{e^{-1}-e^{-n}}{2}\left(\frac{2\theta/(1+\theta)}{2\theta/(1+\theta)-\theta/4n}-1\right) \\
&\geq \frac{e^{-1}-e^{-n}}{2}\left(\frac{2}{2-\theta/4n}-1\right) \\
&\geq \frac{\theta}{50n},
\end{aligned}
$$

where the last inequality assumes that $n \geq 3$. This implies that,

$$
D_{\mathrm{TV}}(\pi_{\beta,n},\pi^{\star}_{\beta}) \geq \frac{\theta}{50n}.
$$

The proof concludes by an application of Pinsker's inequality to lower bound the KL divergence by the squared TV distance, $D^2_{\mathrm{TV}}(p,q) \leq \frac{1}{2}D_{\mathrm{KL}}(p,q)$).

G. Additional Experimental Details

G.1. Hardware and frameworks used for the experiments

All experiments are performed on a node equipped with 3 NVIDIA A100 Tensor Core GPUs. Each model is served on a separate GPU and accessed using vLLM (Kwon et al., 2023) API endpoint. For both beam search and SPECS, we vary the computational budget (e.g., by adjusting the beam width n) to trace out different latency-accuracy scalings.

G.2. Engineering and Performance Optimizations:

In order to make the SPECS algorithm more latency-efficient, we implemented some performance optimizations for increased parallelism and efficiency that are worth noting here. In particular, during the SUBSAMPLE subroutine in Algorithm 2, we need to evaluate the log-likelihoods of the generated beams under the target model as well as their scores under the reward model. We do these two operations concurrently to only pay for the latency cost of the slower of these two operations.

Moreover, during the generation of new candidate beams for a given step, ideally we would like to use prefix-caching so that we do not need to recompute the logits of the previously decoded steps. However, prefix-caching is disabled at vLLM when we also want to access to the logprobs of the prompt tokens, which is necessary for the SUBSAMPLE subroutine that requires the computation of loglikehoods of trajectories under the target model. We solved this problem by changing the vLLM source code to allow the computation of logprobs. The solution to this problem is that we do not actually need to compute the logprobs of the tokens that are prefix-cache hit, we only need the logprobs of the tokens appearing in the last step. Thus we solve the issue by bypassing the prefix-cache hit tokens for the previous steps and efficiently compute the logprobs of the newly generated tokens in the last step, avoiding redundant computations.

G.3. Hyperparameters for SPECS experiments

The hyperparameters used in the experiments in Table 1 are presented in Table 2. Importantly, there are 4 key hyperparameters in this algorithm: n is the beam-width, β is the inverse-temperature for the soft verification, τ is the threshold used in the SUBSAMPLE subroutine, and τ_2 is the cascade threshold used in the CASCADE subroutine. For MATH500 and OlympiadBench datasets, the hyperparameters are chosen on a separate set of 100 questions from that dataset, treated as a validation set. For AMC23 dataset, since there were only 40 questions, we select the hyperparameters that give the best accuracy/latency ratio for different choices of n.

Table 2. Chosen hyperparameters for AMC23 and MATH500 datasets

Dataset	n	β	τ	τ_2
	4	20	8	0.8
AMC23	8	16	8	0.8
	16	12	4	0.9
	4	20	5	0.7
MATH500	8	16	2	0.8
	16	10	8	0.7
	4	12	8	0.9
MATH500	8	8	6	0.9
	16	12	8	0.7

G.4. Histograms for target model usage

Given our algorithm that relies on a cascade using a draft model and a target model, one important question is what the target model usage of SPECS is and how much of the accruacy gains come from the target model vs. draft model. To answer these questions, for the SPECS runs presented in Table 1, we plot the histograms of target model usage in Figure 6. Note that in both of these cases, while achieving either on-par or superior results in terms of accuracy, SPECS always outperforms the baseline algorithm (beam search with the target model alone) on latency. Moreover, notice SPECS does not fully rely on the target model, as evidenced by the histograms, however it adaptively selects whether to sample from the draft model or the target model using the CASCADE subroutine.

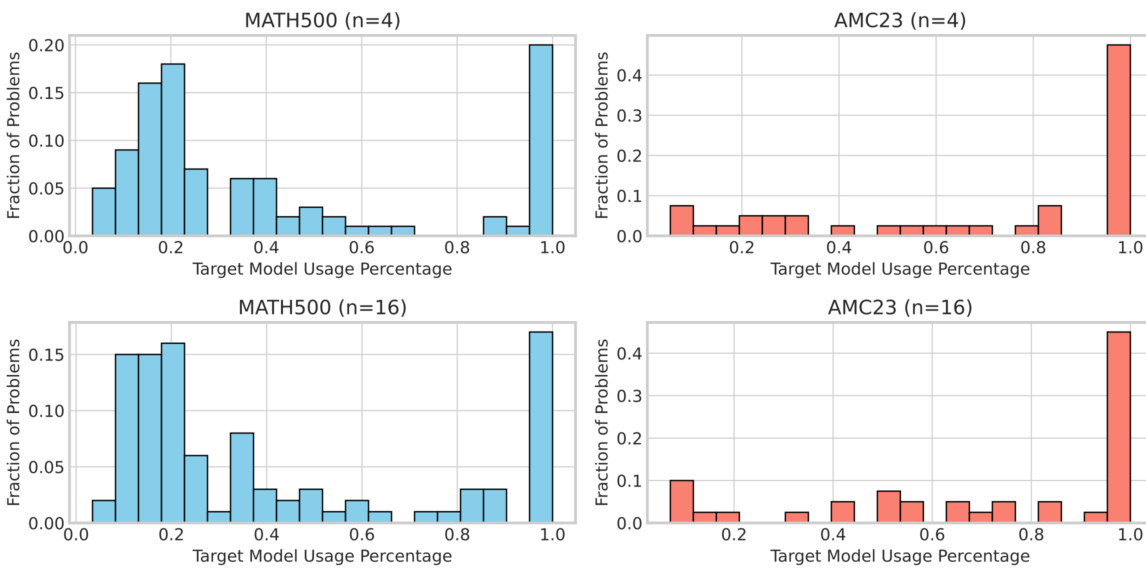

Figure 6. Target model usage histograms of SPECS in MATH500 and AMC23 datasets with n=4 and n=8.