Forward or Reverse KL? Exploring On-Policy Distillation for Speculative Decoding

Columbia COMS4705 Final Project Report
Keywords: *Speculative Decoding, On Policy Knowledge Distillation*

Rishabh Jain
Dept. of Computer Science
Columbia University
rj2790@columbia.edu

Krish Veera
Dept. of Computer Science
Columbia University
krv2123@columbia.edu

Gautam Agarwal
Dept. of Computer Science
Columbia University
ga2726@columbia.edu

Abstract

The deployment of Large Language Models is currently bottlenecked by high inference latency. Speculative Decoding (SD) offers a solution by utilizing a small "draft" model to generate candidate tokens for verification by a larger "target" model. However, the speedup from SD is strictly bounded by the distributional alignment between the draft and target models. Standard off-policy Knowledge Distillation (KD) fails to maximize this alignment due to exposure bias, where the student model drifts from its training distribution during autoregressive generation. While on-policy distillation addresses this by training on student-generated trajectories, the choice of the optimal divergence measure remains an open question, particularly across tasks with varying entropy profiles. In this work, we conduct a comparative study of three on-policy distillation objectives: Forward Kullback-Leibler (FKL), Reverse Kullback-Leibler (RKL), and Jensen-Shannon Divergence (JSD), evaluating their efficacy in training draft models for text summarization and mathematical reasoning. Utilizing a white-box, token-level distillation framework, we demonstrate that the optimal divergence is highly task-dependent. We find that mean-seeking objectives (FKL, JSD) are superior for high-entropy tasks like summarization, where preserving linguistic diversity is critical. Conversely, we show that the mode-seeking nature of RKL is essential for precision-oriented tasks like mathematical reasoning and maximizes the acceptance rate in speculative decoding setups. Our results suggest that while JSD offers a stable middle-ground, on-policy RKL provides the tightest distributional alignment required for efficient inference in reasoning-heavy domains.

1 Key Information

TA Mentor: Noah Foster; **External collaborators:** No; **Sharing project:** No

2 Introduction

A critical barrier to large scale deployment of Large language models is their substantial inference costs. The autoregressive nature of decoding necessitates a full forward pass through a large model for every generated token, resulting in significant cost and latency for applications. Speculative decoding (SD) presents a promising solution to this bottleneck: a lightweight *draft* model proposes candidate tokens that a larger *target* model verifies in parallel, thereby reducing the frequency of expensive target model evaluations [1]. Yet, the efficacy of these speedups is contingent upon the alignment between the draft and target models' distributions [2]. When alignment is poor, acceptance rates plummet, rendering SD ineffective or even detrimental to inference speed. However, the potential speedups from speculative decoding depend strongly on next-token entropy: low-entropy tasks such as mathematical reasoning yield highly deterministic continuations, whereas high-entropy tasks such as summarization allow diverse valid outputs [3].

To bridge the gap between draft and target distributions, Knowledge Distillation (KD) [4] offers a natural training paradigm. Traditionally employed for model compression, KD encourages a smaller

student model to mimic the probabilistic output of a larger, more robust teacher. Recent work, such as DistillSpec [2], adapts distillation specifically for speculative decoding, demonstrating that targeted alignment can significantly boost acceptance rates. However, it hinges on the choice of divergence metric and the specific task and decoding strategy. This observation leads to critical open questions regarding the relationship between alignment objectives and task characteristics.

Motivated by these observations, we follow their approach and focus our experiments on exploring different divergence metrics in *On-Policy Distillation* a ground-truth free training paradigm, for aligning models for speculative decoding, for aligning models for speculative decoding. We examine two model families: Qwen3 [5] for mathematical reasoning and SmolLM [6] for article summarization. Our study is delimited to two representative tasks: the GSM8K dataset [7], characterizing the low-entropy, deterministic nature of mathematical reasoning, and the CNN/DailyMail (CNNDM) dataset [8], representing the high-entropy diversity of text summarization. Within this framework, we compare three divergence objectives commonly used in distillation: Forward KL, Reverse KL, and Jensen-Shannon divergence, to systematically understand how different alignment pressures affect SD performance.

Our contributions are two-fold. First, we develop speculative decoding with dynamic batching for fast inference which allows handling of jagged output batches, and a unified evaluation framework within it that measures speculative decoding efficiency via acceptance rate and alignment behavior between the draft and target models. Second, we conduct a systematic empirical study of on-policy distillation across three divergence objectives: Forward KL, Reverse KL, and Jensen–Shannon, and evaluate their effect on lightweight draft models under both mathematical reasoning and high-entropy summarization settings.

3 Related Work

Speculative Decoding and Inference Acceleration Parallel to model compression, Speculative Decoding (SD) [3, 1] has emerged as a premier method for reducing inference latency without compromising generation quality. SD decouples generation into two distinct phases: a rapid "drafting" phase, where a small model proposes a sequence of tokens, and a parallel "verification" phase, where the target model validates these proposals. The field has rapidly evolved beyond simple draft models; recent innovations include architectural modifications like *Medusa* [9], which adds extra decoding heads to the target model to predict future tokens, and *Eagle* [10], which leverages feature-level autoregression. Despite these structural variations, the fundamental bottleneck remains the probabilistic alignment between the drafter and the verifier. The theoretical speedup is strictly bounded by the acceptance rate.

Knowledge Distillation and the Exposure Bias Challenge Knowledge Distillation (KD) [4] has established itself as a cornerstone technique for compressing LLMs [11]. Traditionally, this process is conducted *off-policy*, where the student learns to mimic the teacher's output probabilities conditioned on a fixed dataset of high-quality text. While computationally convenient, this paradigm ignores the autoregressive nature of language generation, leading to the *exposure bias* phenomenon [12, 13]. During training, the student is guided by ground-truth history (teacher forcing), but during inference, it must generate tokens based on its own predicted history. This distribution shift means that minor errors early in generation place the student in unfamiliar states, causing cascading failures.

On-Policy Distillation and Divergence Measures Recent literature [2, 14] advocates for *on-policy* distillation as a remedy for this mismatch. In this paradigm, the student generates tokens according to its own policy, and the teacher evaluates these student-generated trajectories. This approach shares conceptual similarities with imitation-learning strategies such as DAgger [15], ensuring that the student learns to recover from its own mistakes, a crucial requirement for sequential decision-making in autoregressive text generation. Prior work has further shown that the choice of divergence metric is critical for shaping student behavior [14, 16].

Distillation for Alignment in Speculative Decoding Prior work such as DistillSpec [2] shows that the effectiveness of speculative decoding is tightly governed by draft–target distributional overlap. Rather than optimizing general text quality metrics, successful SD distillation explicitly maximizes agreement with the teacher. This motivates our investigation of divergence objectives.

4 Approach

4.1 Speculative Decoding and Alignment

Speculative decoding [1] accelerates autoregressive inference by pairing a large target model with a smaller draft model. The draft model generates a short block of candidate tokens, and the target model verifies this block in a single forward pass, accepting the longest prefix whose probabilities match the draft's and rejecting the remainder [3]. This eliminates repeated computation over the shared prefix and can yield 2–3$\times$ speedups when many draft tokens are accepted at once.

The effectiveness of speculative decoding depends almost entirely on draft-target alignment. High agreement leads to long accepted prefixes and substantial acceleration, whereas misalignment sharply reduces acceptance and negates speed benefits. Consequently, improving the draft model's predictive distribution, rather than modifying the decoding algorithm itself, is a central lever for enhancing speculative decoding performance.

Token-level acceptance rate and Speed-up For speculative decoding, *token-level acceptance rate* α_{tok} is defined as the probability that a token generated by M_D is accepted by M_T. Empirically, it is the average number of tokens accepted per draft. The inference latency in this regime is strictly governed by α_{tok}. Under standard rejection sampling, the expected speedup is given by S.

$$\alpha_{tok} \approx \frac{\mathbb{E}[\text{number of accepted tokens per draft}]}{\gamma}, \quad S \propto \frac{1-\alpha^{\gamma+1}}{1-\alpha}$$

Crucially, α serves as a direct proxy for the *alignment* between the two models' distributions [3]. Unlike standard distillation which targets task metrics (e.g., perplexity or ROUGE), optimizing for SD requires maximizing the distributional overlap.

Sequence Level Acceptance Rate Following [2], sequence level acceptance rate α_{seq} is defined as the probability of the drafter generating a token that will be accepted by the target

$$\alpha_{seq} = \frac{\mathbb{E}[\text{number of accepted tokens}]}{\mathbb{E}[\text{output length}]}$$

4.2 Divergence Measures: Mode-Seeking vs. Mean-Seeking

Forward KL (D_{FKL}) Often referred to as mean-seeking or mode-covering, FKL encourages the student to cover the entire support of the teacher's distribution. In high-entropy tasks like summarization, this preserves diversity. However, it can force the student to assign non-zero probability to the teacher's "tail", potentially leading to hallucinations in reasoning tasks where precision is paramount.

Reverse KL (D_{RKL}) Known as mode-seeking. By weighing errors by the student's own probability, RKL heavily penalizes the student for generating samples the teacher deems improbable. This metric is particularly favored for reasoning and math tasks, where generating a single correct "mode" (the right answer) is preferable to covering a broad distribution of potential answers.

$$D_{\text{FKL}} : D_{KL}(\pi_T \parallel \pi_\theta) = \mathbb{E}_{y\sim\pi_T}\left[\log\left(\frac{\pi_T(y)}{\pi_\theta(y)}\right)\right], D_{\text{RKL}} : D_{KL}(\pi_\theta \parallel \pi_T) = \mathbb{E}_{y\sim\pi_\theta}\left[\log\left(\frac{\pi_\theta(y)}{\pi_T(y)}\right)\right]$$

Jensen-Shannon Divergence (JSD) A symmetrized and smoothed divergence metric where M is the average distribution. JSD offers a balance between the mode-seeking behavior of RKL and the coverage of FKL.

$$\text{JSD}(\pi_T \parallel \pi_\theta) = \frac{1}{2}D_{\text{KL}}(\pi_T \parallel M) + \frac{1}{2}D_{\text{KL}}(\pi_\theta \parallel M) \text{ where } M = \frac{1}{2}(\pi_T + \pi_\theta)$$

4.3 On-Policy vs. Off-Policy Distillation in LLMs

Traditionally, LLM distillation has relied on *off-policy* methods, where the student is trained on a fixed dataset of prompt-completion pairs generated by the teacher or a static corpus. While effective for general imitation, off-policy distillation suffers from *exposure bias*: during inference, the student generates its own trajectory of tokens, potentially drifting into distribution shifts not covered by the fixed training data [14]. Recent literature [2, 14] advocates for *on-policy* distillation, a paradigm where the student generates tokens based on its own current policy π_θ, and the teacher π_T scores these generations. This approach shares conceptual similarities with Reinforcement Learning strategies like DAgger [15], ensuring that the student learns to recover from its own mistakes.

On-Policy Distillation for Alignment in Speculative Decoding The crucial intuition behind an on-policy setup is that the training setup mimics the behavior the draft model would exhibit during speculative decoding. By optimizing the student to align with the teacher's distribution specifically on the tokens the student is likely to generate, the "draft" model becomes a more robust predictor of the target model's behavior. Zhou et al. [2] demonstrate that training a student on on-policy data with an appropriately chosen divergence metric leads to an improved acceptance rate in speculative decoding. This application motivated us to investigate the choice of divergence metric $\mathcal{D}_{\text{div}}$, and how it shapes the student's behavior.

White-box, Token-Level Distillation We employ a white-box distillation approach, operating directly on the full token-level probability distributions rather than sequence-level approximations. Unlike black-box methods that rely only on generated text, or sequence-level KD which utilizes the logit for the sampled next token, full white-box KD access allows us to utilize the teacher's entire next-token distribution. This provides a richer, dense training signal, enabling the student to fit the nuances of the teacher's distribution, including uncertainty and secondary modes more closely.

Distillation Loss Function To investigate the impact of divergence measures on task performance, we compare the distinct distribution-matching objectives mentioned above. Each objective is computed token-wise on the predictive distributions produced by the teacher and student given the student's generated prefix. The divergence is calculated directly over the output probability distributions after temperature scaling. Since we do not use any gold labels from the datasets while training, our loss function is purely the divergence metric.

$$\mathcal{L}_{\text{on-policy}} = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}[\mathcal{D}_{\text{div}}(\pi_T(\cdot|x,y)||\pi_\theta(\cdot|x,y))] \tag{1}$$

4.4 Task and Model Selection

We selected two distinct datasets to test the universality of our distillation methods across different types of generation distributions. These models were chosen for their strong performance on math and summarization tasks relative to their size.

Mathematical Reasoning (GSM8k) Represents a logic-driven task, where the output distribution is often multi-modal but sharp. We hypothesize that Reverse KL (RKL), known for its mode-seeking behavior, will excel here by encouraging the student to commit to a single high-probability reasoning path rather than "hedging" across multiple options.

Summarization (CNN/DailyMail) Represents an open-ended linguistic task. Summarization involves high stylistic variance (e.g., choosing between synonymous phrasings or sentence structures). We anticipate that the teacher's distribution will be flatter (higher entropy) due to this inherent ambiguity. We hypothesize that FKL or JSD, which balances mode-seeking and mean-seeking behaviors, may offer better stability in these high-entropy regions compared to the aggressive mode-seeking of RKL.

4.5 Evaluation Metrics: Alignment and Speedup Proxies

To rigorously assess the effectiveness of our distilled models, we prioritize distributional alignment over noisy wall-clock measurements. We utilize two key proxies that theoretically bound the achievable speedup:

Token-Level Acceptance Rate (α_{tok}) This measures the probability that any individual token generated by the draft model matches the target distribution. Crucially, Speculative Decoding employs a "prefix verification" scheme: if a draft sequence of length K fails at index i, the first $i-1$ tokens are still accepted. Therefore, improvements in α_{tok} directly translate to a higher Mean Accepted Length (MAL) per step, providing the primary contribution to inference acceleration.

Sequence-Level Consistency (α_{seq}) We also report the acceptance rate of full drafts (where all K tokens are accepted). While less directly correlated with linear speedup than α_{tok} due to the partial acceptance mechanism, this metric serves as a measure of the model's stability and ability to capture long-range dependencies without "drifting" off-policy.

Entropy-Aware Analysis: Standard aggregate metrics (e.g., mean acceptance rate) often obscure critical failure modes. To rigorously analyze where draft models succeed or fail, we decompose performance based on Teacher Entropy $H(\pi_T)$. We therefore analyze α_{seq} as a function of Teacher Entropy. This reveals whether a model's improvements are universal or specific to "easy" (low entropy) or "hard" (high entropy) tokens. While running speculative decoding on the evaluation set, we store next-token entropy for all accepted, and only the *first rejected token* (to simulate rejection sampling), assign them to discretize entropy bins and then plot them to analyse the impact of different divergence metrics.

Hypotheses Based on the theoretical properties of the loss functions, we aim to validate the following expectations: (i) The "Entropy Tax": We expect all draft models to suffer performance degradation as Teacher Entropy increases, reflecting the inherent difficulty of predicting the next token when the teacher itself is uncertain. (ii) Mode-Seeking Advantage: We hypothesize that RKL will outperform FKL and the Baseline on GSM8k by effectively ignoring the "tails" of the teacher's distribution and focusing the student's capacity on the most likely tokens. (iii) Robustness in Ambiguity: We hypothesize that JSD will demonstrate superior robustness in the high-entropy "confusion zones" of CNN/DailyMail, avoiding the catastrophic performance drops seen in baseline models when the teacher's confidence wavers.

Implementation of Dynamic Speculative Decoding A critical prerequisite for this study was the development of a robust Speculative Decoding (SD) evaluation harness capable of handling dynamic draft lengths within a single batch. Most open-source SD implementations operate on a "synchronized" paradigm, where the speculative length is fixed across the batch, or strictly limited by the worst-performing sequence. Our code adapts [17] repository, and we extend it to support batched inference required substantial engineering effort to manage jagged sequence lengths. Since different sequences in a batch accept varying numbers of tokens at each verification step, the resulting active sequence lengths become non-uniform. We implemented complex masking and padding logic to maintain tensor contiguity without corrupting the causal attention masks of individual sequences. Furthermore, we encountered a architectural limitation in the standard Hugging Face `transformers` library, which does not allow for dynamic Key-Value cache across batch elements. Consequently, we were forced to bypass KV-cache optimization, leading to inference overhead. Due to these resource constraints, we restricted our final evaluations to a subset of the test datasets, ensuring rigorous analysis while maintaining computational feasibility.

4.6 Model Selection and Experimental Setup

Mathematical Reasoning (GSM8k) For the precision-centric task of mathematical reasoning, we employ the **Qwen** model family, utilizing *Qwen3-4B-Instruct* as the target (teacher) and *Qwen3-0.6B-Instruct* as the draft (student) model. The Qwen series was selected for its demonstrated state-of-the-art reasoning capabilities relative to parameter count.

Text Summarization (CNN/DailyMail) Abstractive summarization represents a high-entropy generation task, necessitating extensive training data to capture diverse linguistic alignments. To maintain computational feasibility without sacrificing model quality, we utilize the **SmolLM** family: *SmolLM-1.7B-Instruct* serves as the target, with *SmolLM-360M-Instruct* as the drafter. These compact models allow for efficient iteration on the large-scale CNN/DailyMail dataset while maintaining coherent instruction-following behavior.

Training Protocol: Pure On-Policy Distillation A critical deviation in our approach is the omission of Supervised Fine-Tuning (SFT) on the downstream datasets. Given that both the teacher and student models are already instruction-tuned and exhibit strong zero-shot performance on their respective tasks, we focus exclusively on the *alignment* phase. Consequently, no ground-truth labels from the datasets were utilized during training; the distillation process relied entirely on student-generated trajectories scored by the frozen teacher, adhering to a strict on-policy paradigm.

5 Experiments

5.1 Data

GSM8k Mathematical Reasoning We use GSM8k math dataset, consisting of approx. 8000 training prompts. We distill Qwen0.6b from Qwen-4b-Instruct on this dataset, and limit output max tokens to 312 following [2]. For eval, we choose a subset of 1000 prompts from test set. Prompts consisted of math word problems requiring full reasoning trajectories and numeric answers.

CNN Daily Mail Summarization CNN/DailyMail consists of articles of varying sizes. To limit computational requirements, we filter the dataset to remove texts longer than 512 tokens. For generation in training and evaluation, we limit model output to 128, which we believe is sufficient for a short summary for an article of length 512. This left us with a dataset with roughly 50,000 prompts. For eval, we choose a subset of 2000 prompts from test set. Prompts contained document snippets requiring short summaries.

5.2 Experimental Details

On-Policy Distillation We use GKDTrainer[14] from Huggingface TRL library for our experiments. In all scenarios, *distillation relied solely on student-generated* trajectories; gold answers and summaries were excluded entirely from the training process. The teacher model remained frozen throughout, while the student was trained for a 1 epoch on the chosen divergence objective.

We trained all models using the HuggingFace TRL GKDTrainer on a single NVIDIA H100 (80GB) GPU on GCP. Qwen models were trained with 8-bit quantization via bitsandbytes, while SmolLM models were trained in full precision. A typical run required approximately 2 hours for Qwen and 1 hour for SmolLM, amounting to a total compute budget of roughly 17 GPU-hours across all experiments.

5.3 Evaluation method

We evaluate models via our batched speculative decoding implementation with block size $\gamma = 5$ and nucleus sampling (task-specific temperature, $p = 0.95$). Following the definitions in Section 4.5, we report token-level (α_{tok}) and sequence-level (α_{seq}) acceptance rates as our primary measures of **distributional alignment**. Because the target model remains frozen and speculative decoding (via rejection sampling) preserves the target model's output distribution, we do not report standard task metrics such as accuracy or ROUGE. Mismatch between the student and teacher manifests would reflect as a reduction in acceptance rates, which in turn bounds the achievable inference speedup [3]. While running speculative decoding on the evaluation set, we store next-token entropy for all accepted, and only the *first rejected token* (to simulate rejection sampling), assign them to discretize entropy bins and then plot them to analyse the impact of different divergence metrics.

Draft Acceptance Rate vs Teacher Entropy Analysis To rigorously evaluate alignment across the full spectrum of uncertainty, we analyzed α_{seq} against Teacher Entropy. We divided entropy into bins, from highly deterministic states ($H < 10^{-4}$) to highly ambiguous states ($H > 10^{0}$), and calculated the empirical α_{seq} per bin. Note that since majority of tokens are low-entropy, the collection distribution is inherently skewed - and we therefore ignore those entropy bins which have tokens less than 2000 for statistical significance of α_{seq}. We also overlay a bubble chart over each bin to show the relative number of tokens in that bin.

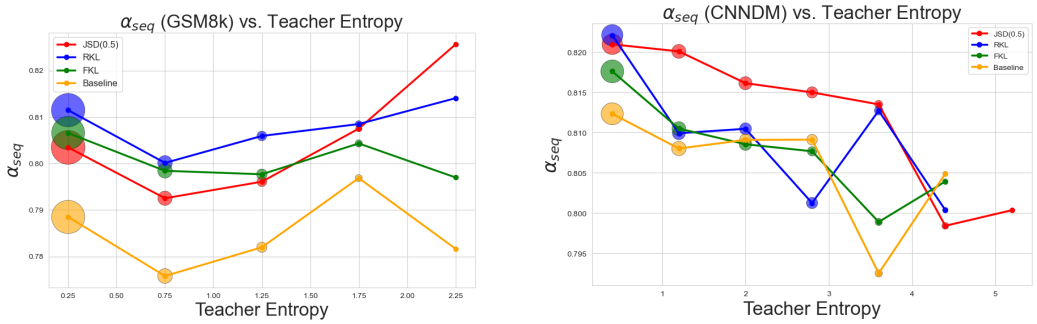
5.4 Results

Model Name	Metric	Baseline	RKL	JSD	FKL
Qwen3 (GSM8k)	α_{seq}	70.94	**72.67**	72.18	72.33
	α_{tok}	49.52	**53.93**	52.63	52.98
SmolLM (CNN/DM)	α_{seq}	71.82	71.93	**72.35**	72.08
	α_{tok}	53.91	54.43	**55.35**	54.47

Table 1: Acceptance Rates (α_{tok}: Token-Level; α_{seq}: Sequence-Level) across Divergence Methods.

Table 1 presents a comprehensive comparison of acceptance rates across mathematical reasoning (GSM8k) and summarization (CNN/DM) tasks, highlighting the task-dependent efficacy of different divergence objectives. On the deterministic GSM8k benchmark, the Reverse KL (RKL) objective achieves the highest performance, improving the token-level acceptance rate (α_{tok}) by over 4.4 percentage points compared to the baseline (53.93% vs. 49.52%). This reinforces the hypothesis that mode-seeking alignment is optimal for logic-heavy tasks where precision is paramount. Conversely, on the high-entropy CNN/DM summarization task, the Jensen-Shannon Divergence (JSD) objective yields the best results, achieving a token-level acceptance rate of 55.35% and a sequence-level rate (α_{seq}) of 72.35%. This suggests that for open-ended generation, the balanced nature of JSD - avoiding the extremes of zero-forcing provides a more robust signal for aligning the drafter with the teacher's diverse output distribution. Across both domains, all distilled models consistently outperform the baseline, confirming the general effectiveness of on-policy distillation for speculative decoding.

The Superiority of Mode-Seeking in GSM8k As illustrated in Figure 1a, the performance on the mathematical reasoning task (GSM8k) reveals a clear hierarchy. The Reverse KL (RKL) model (Blue) consistently outperforms both the Forward KL (Green) and JSD (Red) models, particularly



(a) α_{seq} vs Teacher entropy for GSM8k.

(b) α_{seq} vs Teacher entropy for CNNDM.

Figure 1: Variation of α_{seq} vs. Teacher Entropy in Speculative Decoding simulations. Bubble chart indicates relative number of points per bin, which naturally reduce with increasing teacher entropy. RKL wins for GSM8k, where JSD wins for CNNDM.

in the low-entropy regime ($H < 10^{-2}$). At these near-deterministic points—which constitute the majority of mathematical reasoning steps. Notably, we observe a distinct U-shaped recovery in the high-entropy region ($H > 10^{-1}$). This likely corresponds to function words or stylistic choices where the teacher's distribution is flat; here, the student's top choice, while potentially different from the teacher's greedy path, often falls within the teacher's acceptable top-k set, leading to a resurgence in acceptance rates. However, since the number of points at the tail are relatively low, the difference in acceptance rates is less statistically significant.

Robustness of JSD in High-Entropy Regimes The summarization task (CNN/DM), shown in Figure 1b, presents a markedly different landscape characterized by higher overall entropy. While RKL (Blue) performs competitively in low-entropy regions ($H < 10^{-1}$), it suffers a sharp performance degradation as entropy exceeds 1, dropping below both FKL and JSD. In contrast, the Jensen-Shannon Divergence (JSD) model (Red) demonstrates remarkable stability in these high-entropy regimes. While RKL collapses under the ambiguity of open-ended summarization, likely due to its zero-forcing penalty punishing valid synonyms, JSD maintains a higher α_{seq} well into the high-entropy tail.

6 Analysis

While the demonstrated gains are marginal, we analyse the observed patterns below that indicate strength in our hypothesis about the three divergence metrics under consideration.

Task-Dependent Divergence Superiority As presented in Table 1, our results demonstrate that the optimal on-policy divergence measure is task-dependent, aligning with our theoretical expectations regarding entropy and mode-coverage. For mathematical reasoning (GSM8k), where the output distribution is typically peaked around a single correct reasoning path, the mode-seeking Reverse KL (RKL) objective proves superior. It achieves the highest token-level acceptance rate (α_{tok}) of 53.93%, a notable improvement over the baseline (49.52%) and the mean-seeking FKL (52.98%). Conversely, for summarization (CNN/DM), which requires modeling a broader, high-entropy distribution of valid linguistic variations, the balanced Jensen-Shannon Divergence (JSD) outperforms all other methods ($\alpha_{tok} = 55.35\%$). Although gains are modest (1-4%), the consistent ranking across runs supports the link between divergence geometry and task entropy.

Entropy Dynamics and Model Alignment To further investigate the mechanism of alignment, we analyze the sequence-level acceptance rate (α_{seq}) as a function of the teacher's predictive entropy. In the GSM8k domain (1a), the data is concentrated in low-entropy regions (indicated by larger bubble sizes at $H < 0.75$), reflecting the deterministic nature of mathematical reasoning. In this dense, high-confidence regime, RKL (blue) consistently outperforms the baseline and competing divergences. This confirms that RKL effectively forces the student to "snap" to the teacher's primary mode, reducing the probability of sampling valid but "unaligned" tokens. In the CNN/DM domain (1b), the teacher's entropy is significantly higher and more dispersed. Here, we observe that RKL suffers from performance degradation in higher entropy regions ($H > 3.0$) likely due to its tendency to aggressively prune valid semantic alternatives (mode-collapse), thereby drifting away from the teacher's diverse support. JSD (red), however, maintains robust alignment across the entire entropy spectrum. By interpolating between mode-seeking and mean-seeking behaviors, JSD allows the

student to cover the necessary linguistic diversity without succumbing to the "tail-chasing" behavior often seen with pure FKL. It therefore offers a more robust objective, preserving the diversity required to match the teacher's broader support during ambiguous generation steps.

Limitations of High-Entropy Analysis It is important to note a caveat in our entropy analysis: as visualized by the diminishing bubble sizes in the right-hand regions of both plots, the number of samples with extremely high teacher entropy is relatively low. Consequently, the apparent performance spikes (e.g., JSD's sudden rise at $H = 2.25$ in GSM8k) may be artifacts of data sparsity rather than robust algorithmic superiority.

7 Conclusion

In this work, we investigated the impact of divergence measures in on-policy distillation for optimizing Speculative Decoding (SD) draft models. Our findings confirm that the optimal alignment objective is intrinsic to the entropy profile of the downstream task. We demonstrate that for precision-centric tasks like mathematical reasoning (GSM8k), the mode-seeking **Reverse KL (RKL)** divergence is superior, effectively pruning the student's distribution to match the teacher's deterministic reasoning paths. Conversely, for high-entropy tasks like abstractive summarization (CNN/DailyMail), the **Jensen-Shannon Divergence (JSD)** proves more robust, balancing the need for linguistic diversity with the requirement for structural alignment.

Limitations Despite these consistent qualitative trends, we acknowledge that the quantitative gains in acceptance rates remained marginal across our experiments, typically ranging between 1% and 4% over baselines. The models could be trained for longer on larger datasets to align with their size. Furthermore, due to the engineering constraints of implementing dynamic batching without native KV-cache pruning support, our rigorous evaluations were restricted to representative subsets of the test data. While we believe these subsets are statistically significant, larger-scale validation is required to fully generalize these findings to production-grade deployments. Additional evaluation is needed to compare the outputs of the models after training to judge output quality post-distillation.

Future Work Future research should look beyond static divergence objectives. A promising avenue is *Adaptive Generalized Knowledge Distillation*, where the interpolation hyperparameter β (controlling the FKL-RKL trade-off) is dynamically adjusted based on the teacher's real-time entropy. Additionally, investigating the interplay between these on-policy objectives and architectural SD variants, such as Medusa or Eagle, could unlock significantly larger speedups by decoupling the verification logic from the drafting model's capacity constraints. Another natural extension is the incorporation of reward modeling into the speculative decoding loop. By learning a reward signal that captures draft quality or alignment with the teacher's verification preferences, the student could be guided toward trajectories with inherently higher acceptance probabilities, effectively shaping its generation behavior to minimize corrections and further improve SD efficiency.

8 Team Contributions

Rishabh Jain architected the end-to-end experimental framework, implementing dynamic batched speculative decoding and custom white-box distillation pipelines. He executed all on-policy distillation experiments and formulated the entropy-based analysis to quantify teacher-student alignment. Additionally, he managed the project infrastructure and containerized environments for reproducible testing. In prior phases, he implemented MatQuant paper and reward modeling to optimize training efficiency.

Krish Veera contributed to project design and experimentation, running early architectural mini-experiments (e.g., attention-head pruning) and multiple distillation runs for llama models that helped mold the final methodology. He also built the speculative-decoding evaluation framework for baseline SD metrics and alignment metrics and helped organize the project structure throughout the course of the project while drafting major portions of the milestone and final report.

Gautam Agrawal read multiple research papers and acivey contributed to the ideation of the project track. He contributed to experimentation, running experiments on llama and smoll models for on-policy and off-policy knowledge distillation. He also built the evaluation framework for speculative decoding.

References

[1] Chao Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *DeepMind Technical Report*, 2023.

[2] Yi Zhou, Kedi Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-Francois Kagy, and Rajiv Agarwal. Distillspec: Improving speculative decoding via knowledge distillation. In *International Conference on Learning Representations (ICLR)*, 2024.

[3] Yaniv Leviathan, Mor Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. *arXiv preprint arXiv:2211.17192*, 2022.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[5] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[6] Loubna Ben Allal, Anton Lozhkov, and Elie Bakouch. Smollm — blazingly fast and remarkably powerful. Hugging Face Blog, Jul 2024.

[7] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Łukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Chris Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[8] abisee. Cnn/dailymail dataset. Hugging Face Dataset, 2023.

[9] Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024.

[10] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024.

[11] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[12] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[13] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[14] Rishabh Agarwal, Aditya Menon, Ankit Singh Rawat, Kedi Lyu, Yi Zhou, Aishwarya Agarwal, Afshin Rostamizadeh, and Sanjiv Kumar. On-policy distillation of language models: Learning from self-generated mistakes. *arXiv preprint arXiv:2306.13649*, 2023.

[15] Stephane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. *arXiv preprint arXiv:1011.0686*, 2010.

[16] Yuqing Zhang, Xueru Qiu, Emre Temel, Xiaodong Liu, Peng Li, Jingjing Wang, and Graham Neubig. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.

[17] github.com/romsto/speculative-decoding.

Test Dataset	γ	max tokens	Avg Acceptance Rate (%)
Writing prompts	$\gamma = 5$	100	47.10%
Writing prompts	$\gamma = 5$	200	46.45%
Writing prompts	$\gamma = 10$	200	29.86%
GSM8K	$\gamma = 5$	100	71.54%
GSM8K	$\gamma = 5$	200	73.47%
GSM8K	$\gamma = 10$	200	58.00%

Table 2: Speculative Decoding Acceptance Rates for LLaMa-7b as target and LLaMA-1b as drafter

A Appendix (optional)

A.1 Model Configuration and Training

All draft models were trained for a single epoch over the task datasets using full-parameter fine-tuning. Gradients were computed directly from the divergence objectives (FKL, RKL, or JSD) in an on-policy manner. To fit GPU memory constraints, the *Qwen-4B-Instruct* student was trained using 8-bit quantization via `bitsandbytes`, while *SmolLM-360M* was trained in full precision.

A.2 Hyperparameters

Optimization used AdamW with a fixed learning rate of 5×10^{-5} and cosine decay. During the on-policy generation phase, sampling temperatures were set to $T = 0.6$ for Qwen and $T = 0.7$ for SmolLM, matching recommended inference configurations. Batch sizes and sequence lengths were constrained by hardware limits.

A.3 Computational Resources

All experiments were run on a single NVIDIA H100 GPU. Training the Qwen student required approximately 2 hours per run (batch size 16, sequence length 312), while the SmolLM student required roughly 1 hour (batch size 128, sequence length 128). Across divergence objectives and model families, total compute expenditure was 17 GPU-hours (10 for training, 7 for evaluation).

A.4 Speculative Decoding with a Draft Model

Algorithm 1 Speculative Decoding

Require: Prompt x, max length N, draft M_d, target M_t, speculation length γ

$y \leftarrow \emptyset$ {Output sequence}
while $|y| < N$ **do**
  $z \leftarrow M_d(x \parallel y)$ {Draft γ tokens autoregressively}
  $p \leftarrow M_t(x \parallel y)$ {Target logits for $|z| + 1$ tokens in parallel}
  $k \leftarrow \max\{i \leq \gamma : \forall j \leq i,\, z_j \sim p_j\}$ {Longest accepted prefix via rejection sampling}
  $y \leftarrow y \parallel z_{<k+1}$
  if $k < \gamma$ **then**
    $y \leftarrow y \parallel \text{sample}(p_{k+1})$ {Fallback sample}
  end if
end while
return y

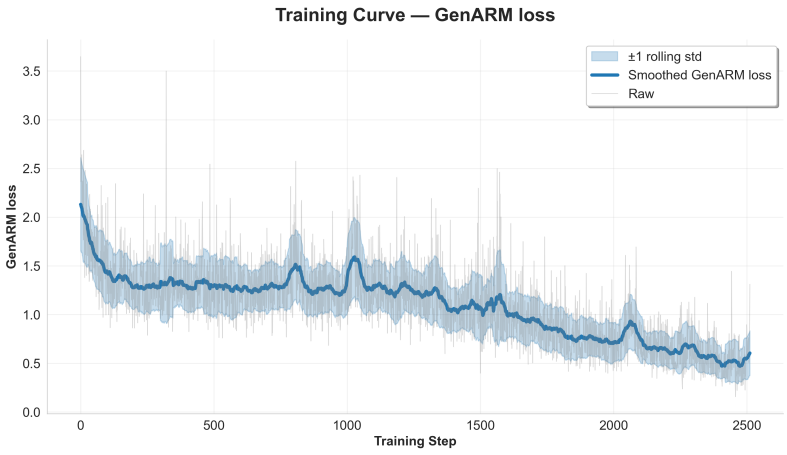


Figure 2: GenARM loss curve for 2 epochs on HH-RLHF dataset for Qwen-4b-Instruct-2507