DistillBeam: Multi-Trajectory Knowledge Distillation for Efficient Speculative Decoding

Anonymous ACL submission

Abstract

The efficacy of speculative decoding(SD) is fundamentally constrained by the alignment between the draft and target models. Existing distillation approaches for SD rely on single-trajectory supervision, which induces exposure bias and degrades acceptance rates at inference time. To address this, we introduce **DistillBeam**, a framework that optimizes draft-target alignment via multi-trajectory distillation. By aggregating supervision from multiple high-probability teacher trajectories, DistillBeam approximates the target model's full structural support, thereby mitigating sequence drift. We further tackle the prohibitive storage overhead of multi-beam distillation by demonstrating that aggressive Top-K truncation ($K = 50$) reduces offline storage by 99.9% without degrading alignment. Extensive evaluation across 20 languages reveals that DistillBeam achieves wall-clock speedups of 35-65% over autoregressive decoding, with particularly strong gains in morphologically rich languages where baseline methods struggle.

1 Introduction

Large language models (LLMs) have demonstrated strong performance across a wide range of natural language processing tasks. However, their deployment in latency-sensitive production settings remains constrained by the inherently sequential nature of autoregressive decoding (Pope et al., 2023; Shazeer, 2019; Miao et al., 2023). Each generated token requires a full forward pass through the model, making inference memory-bandwidth limited rather than compute-bound. As model sizes scale to billions of parameters, this bottleneck leads to prohibitive latency, limiting the practical use of LLMs in interactive applications such as dialogue systems, machine translation, and code generation.

Speculative decoding (SD) has recently emerged as an effective strategy for alleviating this limitation without sacrificing output quality (Leviathan et al., 2023; Chen et al., 2023; Stern et al., 2018). SD follows a draft and verify paradigm in which a lightweight draft model proposes multiple tokens that are verified in parallel by a larger target model. A modified rejection sampling procedure ensures that the resulting output distribution exactly matches that of standard autoregressive decoding from the target model. Empirical studies report inference speedups of 2-3$\times$ on production-scale models, with performance tightly coupled to the acceptance rates. (Cai et al., 2024; Li et al., 2024; Fu et al., 2024).

Improving draft-target alignment is therefore central to maximizing speculative decoding efficiency. Knowledge distillation (KD) provides a natural mechanism for aligning the draft model with the target model's predictive distribution (Hinton et al., 2015; Buciluǎ et al., 2006). While KD has been extensively studied for model compression and transfer learning, its application to speculative decoding introduces distinct challenges. Standard distillation objectives are typically optimized for downstream task accuracy and do not explicitly account for the rejection-sampling dynamics that govern speculative decoding latency.

Recent work has begun to adapt distillation techniques for speculative decoding, showing that task-aware objectives can improve acceptance rates. However, several key issues remain insufficiently explored. First, prior work lacks a systematic comparison of divergence objectives tailored to rejection-based verification, leaving the optimal training objective unclear. Second, existing approaches predominantly rely on single-trajectory supervision, which inadequately captures the multiple high probability generations of the target model distribution and exacerbates sequence drift during inference. Third, high-fidelity distillation at scale is hindered by the prohibitive storage cost of full-vocabulary teacher distributions.

We introduce **DistillBeam**, a framework de-

signed to optimize draft-target alignment via multi-trajectory distillation. DistillBeam aggregates supervision from multiple high-probability teacher trajectories to approximate the target model's structural support. To address the scalability constraints of multi-beam supervision, we employ Top-K truncation, ensuring the method remains storage-efficient.

Our contributions are summarized as follows:

1. **DistillBeam Framework.** A unified distillation framework for speculative decoding that leverages multi-trajectory supervision to improve draft-target alignment and mitigate sequence drift.

2. **Systematic Divergence Analysis.** A systematic evaluation of divergence objectives (KL, RKL, JSD, Hellinger, TVD) tailored to rejection-based verification.

3. **Storage-Efficient Scaling.** We demonstrate that Top-K teacher distributions reduce offline storage by 99.9% while preserving alignment quality.

2 Related Work

2.1 Speculative Decoding

Since the formalization of speculative decoding and sampling methods (Leviathan et al., 2023; Chen et al., 2023; Stern et al., 2018), prior work has progressed primarily along two directions: improving verification mechanisms and designing specialized draft architectures. Early approaches typically employed smaller autoregressive variants of the target model as drafters. More recent work has explored alternatives to strictly autoregressive drafting. For example, Cai et al. (2024) proposed multi-head decoding to predict multiple future tokens in parallel, while Li et al. (2024) leveraged feature-level extrapolation to reduce reliance on token-level generation.

In parallel, substantial effort has focused on verification strategies that move beyond simple rejection sampling. Token-tree verification (Miao et al., 2023) and related graph-based methods verify multiple candidate continuations simultaneously, increasing the expected number of accepted tokens per decoding step. Despite these advances, recent surveys (Xia et al., 2024) observe that improvements in verification logic yield diminishing returns when the draft model poorly approximates the target distribution. In practice, distributional mismatch between the draft and target remains a dominant bottleneck for speculative decoding efficiency.

2.2 Knowledge Distillation for Speculative Decoding

To address draft-target mismatch, several works have adapted knowledge distillation (KD) specifically for speculative decoding. Unlike conventional KD, which primarily optimizes downstream task accuracy, speculative decoding-oriented distillation directly targets high acceptance rates during verification. Zhou et al. (2023) were among the first to formalize this distinction, demonstrating that on-policy distillation using teacher-generated data substantially improves acceptance rates compared to training on human reference data.

Subsequent work has explored dynamic and hybrid distillation strategies. Zhao et al. (2024) proposed an online distillation framework that continuously aligns the draft model during inference, while Xu et al. (2024) combined distillation with n-gram matching to accelerate prediction. However, these approaches predominantly minimize standard KL divergence against a single teacher trajectory, typically obtained via greedy decoding or sampling a single sequence. Consequently, the draft model is supervised on only one specific realization of the target distribution, limiting its ability to generalize to valid alternative continuations during inference.

2.3 Divergence Objectives in Generative Modeling

The choice of divergence function is a first-order determinant of behavior in generative modeling. Literature in variational inference and GANs has extensively characterized the trade-offs between Forward KL (mass-covering) and Reverse KL (mode-seeking) (Minka et al., 2005). Beyond KL-based objectives, bounded and symmetric divergences such as Jensen-Shannon, as well as distance-based measures including Hellinger distance, offer alternative optimization landscapes regarding tail sensitivity and gradient stability (Agarwal et al., 2024; Theis et al., 2015; Nowozin et al., 2016).

In the specific context of speculative decoding, these properties have direct implications for inference latency: overestimating token probabilities leads to immediate rejection. Despite this, a systematic evaluation of how these divergence objectives interact with rejection-based verification is absent

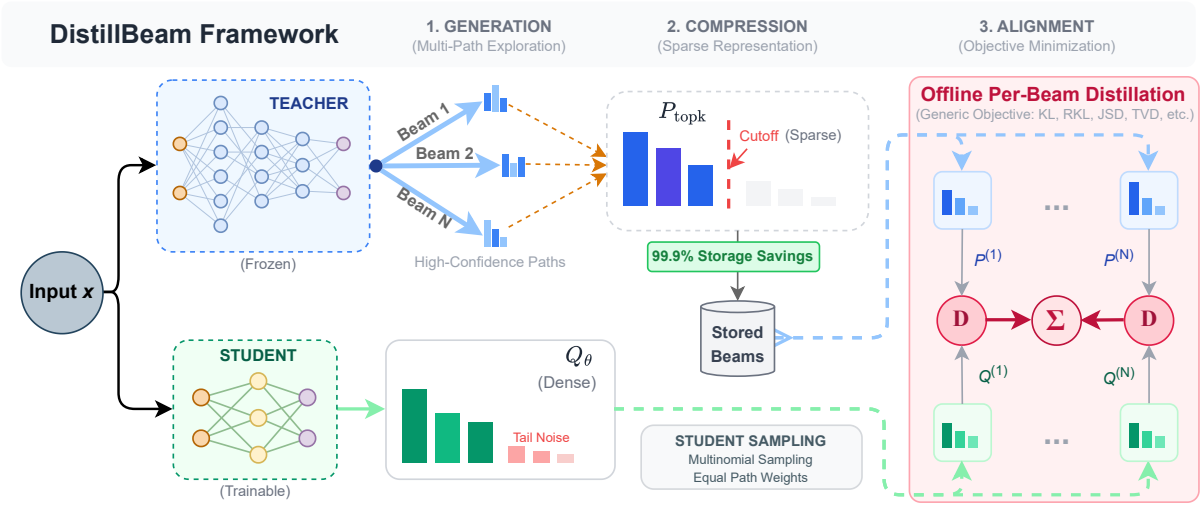


Figure 1: Overview of the **DistillBeam** framework. The pipeline consists of (1) Multi-path generation via beam search, (2) Storage-efficient Top-K compression, and (3) Offline alignment using the Multi-Trajectory Distillation (MTD) objective.

from the current literature, with most prior work defaulting to standard Forward KL without specific justification related to the verification mechanism.

2.4 Multi-Trajectory Supervision and Efficiency

Standard distillation strategies suffer from exposure bias (Bengio et al., 2015; Agarwal et al., 2024), as the draft model is trained on a constrained set of valid continuations. Sequence-level knowledge distillation (SeqKD) attempts to mitigate this by supervising the student with multiple high-probability teacher trajectories. Kim and Rush (2016) demonstrated that training on beam search outputs improves generalization by exposing the student to diverse, globally coherent sequences. This broader supervision enables recovery from local errors, a property vital for preventing cascading rejections in speculative decoding.

However, applying multi-trajectory supervision to large language models introduces a data efficiency bottleneck. Effective alignment requires soft targets to capture teacher uncertainty (Hinton et al., 2015), but storing dense logits for multiple beams across the massive vocabularies of modern LLMs induces prohibitive storage overhead. While recent efficiency-oriented work has targeted model-side optimizations such as quantization (Dettmers et al., 2023) or pruning, the optimization of the distillation data pipeline itself, specifically the trade-off between trajectory diversity and storage constraints, remains relatively unexplored.

3 Methodology

We detail the mathematical formulation, divergence analysis, and optimization strategy of the DistillBeam framework.

3.1 Problem Formulation

Let $p(\cdot \mid x)$ denote the token distribution of the target model and $q_\theta(\cdot \mid x)$ that of the draft model, given context x. In speculative decoding, a token $y \sim q_\theta$ is accepted with probability

$$\min\left(1, \frac{p(y \mid x)}{q_\theta(y \mid x)}\right).$$

This ratio represents the "confidence" the target model places in the draft's proposal; if the draft assigns a lower probability to a token than the target does, acceptance is guaranteed, whereas overestimating the probability leads to rejection.

The expected acceptance rate α therefore depends on how closely q_θ aligns with p, particularly in regions where p assigns high probability. Our objective is to train q_θ to maximize α, rather than conventional likelihood-based metrics.

3.2 Distillation Objectives

Given a target model p and a draft model q_θ parameterized by θ, we consider six distillation objectives:

Supervised Fine-Tuning (SFT) trains the draft model using standard cross-entropy loss against a reference sequence y_{ref}, equivalent to Maximum Likelihood Estimation (MLE) (Sutskever et al., 2014):

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{|y_{\text{ref}}|} \log q_\theta(y_{\text{ref},t} | x, y_{\text{ref},<t}). \quad (1)$$

The reference sequence can be either human-generated (Gold) or produced by the teacher model (Teacher-Greedy or Teacher-Multinomial).

KL Divergence (KL) minimizes the forward KL divergence between the teacher and student distributions (Kullback and Leibler, 1951; Hinton et al., 2015): $D_{\text{KL}}(p \,\|\, q_\theta)$:

$$\mathcal{L}_{\text{FKL}} = \sum_{y \in \mathcal{V}} p(y \mid x) \log \frac{p(y \mid x)}{q_\theta(y \mid x)}. \quad (2)$$

KL is mass-covering and encourages q_θ to approximate the full support of p. For capacity-limited draft models, this often flattens the distribution, reducing probability mass on the dominant modes and hurting acceptance under greedy or low-temperature decoding.

Reverse KL Divergence (RKL) minimizes the reverse KL divergence:

$$\mathcal{L}_{\text{RKL}} = \sum_{y \in \mathcal{V}} q_\theta(y \mid x) \log \frac{q_\theta(y \mid x)}{p(y \mid x)}. \quad (3)$$

RKL is mode-seeking (Minka et al., 2005; Theis et al., 2015) and penalizes the draft model for assigning probability to tokens that the target model considers unlikely. Since such tokens are frequently rejected during speculative decoding, RKL directly aligns the training objective with the maximizing acceptance rate. As noted in Section 2.3, RKL's mode-seeking property makes it particularly suitable for SD alignment.

Jensen-Shannon Divergence (JSD) provides a symmetric and smoothed alternative to KL divergence (Lin, 2002). It is defined as the average of the forward and reverse KL divergences from a mixture distribution $m = \frac{1}{2}(p + q_\theta)$:

$$\mathcal{L}_{\text{JSD}} = \frac{1}{2} D_{\text{KL}}(p \| m) + \frac{1}{2} D_{\text{KL}}(q_\theta \| m). \quad (4)$$

Unlike KL divergence, JSD is bounded $[0, \ln 2]$. In our implementation, we compute the mixture logits and minimize the divergence symmetrically. JSD offers a balance between the mass-covering behavior of KL and the mode-seeking behavior of RKL, potentially stabilizing gradients when draft and target distributions are initially disjoint.

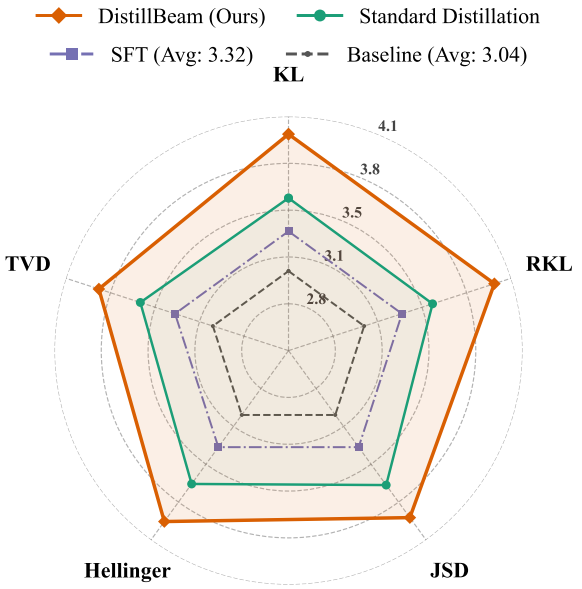


Figure 2: Radar chart comparing the Block Efficiency of DistillBeam (orange) against standard SFT and Forward KL baselines. DistillBeam achieves superior draft-target alignment across all evaluated divergence frontiers.

Hellinger Distance (HEL) is a metric satisfying the triangle inequality, defined based on the Euclidean distance between the square roots of the probability distributions (Beran, 1977; Le Cam and Yang, 2000). We minimize the squared Hellinger distance:

$$\mathcal{L}_{\text{HEL}} = \frac{1}{2} \sum_{y \in \mathcal{V}} \left(\sqrt{p(y \mid x)} - \sqrt{q_\theta(y \mid x)} \right)^2. \quad (5)$$

Geometrically, this operates on the probability simplex sphere. Because it operates on $\sqrt{p}$ rather than $\log p$, Hellinger distance is less sensitive to extreme outliers in the tail of the distribution compared to KL-based measures, potentially allowing the draft model to focus on the core probability mass without over-fitting to negligible tail probabilities.

Total Variation Distance (TVD) measures the L_1 distance between the probability distributions (Villani et al., 2008):

$$\mathcal{L}_{\text{TVD}} = \frac{1}{2} \sum_{y \in \mathcal{V}} |p(y \mid x) - q_\theta(y \mid x)|. \quad (6)$$

TVD has a direct interpretation in optimal transport and hypothesis testing: it represents the largest possible difference in probabilities the two models can assign to the same event. In the context of speculative decoding, minimizing TVD directly constrains the absolute error in probability estimation, which serves as a tight bound for the rejection probability.

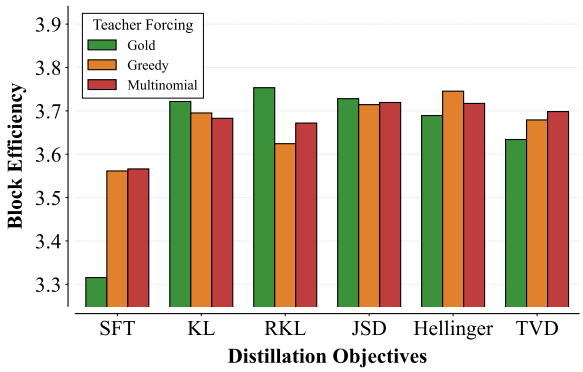


Figure 3: Impact of forcing strategies on alignment. Distilling from **Teacher-Multinomial** trajectories (red) consistently outperforms human references (Gold) and greedy decoding across all divergence objectives.

3.3 Multi-Trajectory Distillation (MTD)

Single-trajectory distillation exposes the draft model to a narrow subset of the target's valid continuations. We propose aggregating supervision from a set of trajectories $\mathcal{B}_N(x) = \{y^{(1)}, \ldots, y^{(N)}\}$ obtained via beam search (width N). At each step t, the aggregated teacher distribution is:

$$p_{\text{MTD}}(y_t \mid x) = \frac{1}{N} \sum_{i=1}^{N} p(y_t \mid x, y^{(i)}_{<t}). \quad (7)$$

Generalized Distillation Objective Multi-Trajectory Distillation is agnostic to the choice of divergence function. Let $\mathcal{D}(\cdot \,\|\, \cdot)$ denote a token-level divergence between two any distributions, The generalized MTD objective is defined as

$$\mathcal{L}_{\text{MTD}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{|y^{(i)}|} \mathcal{D}\Big(p_{\text{MTD}}(\cdot) \Big\| \, q_\theta(\cdot)\Big). \quad (8)$$

This formulation strictly generalizes single-trajectory distillation: when $N = 1$, it reduces to standard greedy or sampled teacher forcing. For $N > 1$, the draft model is trained to align with a broader, more representative subset of the target model's support.

3.4 Top-K Truncated Teacher Representations

Storing full-vocabulary teacher distributions for multi-beam trajectories is prohibitively expensive. To address this, we employ Top-K truncation, approximating the full teacher distribution p with a sparse renormalized distribution $\hat{p}_K$. Let $\mathcal{T}_K$ denote the set of the K highest-probability tokens in $p(\cdot \mid x)$. We define:

$$\hat{p}_K(y_t) = \begin{cases} \dfrac{p(y_t)}{\sum_{z \in \mathcal{T}_K} p(z)} & \text{if } y_t \in \mathcal{T}_K, \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

For a standard vocabulary of $|\mathcal{V}| \approx 100$k (e.g., Llama-3), setting $K = 50$ yields a theoretical compression ratio of $|\mathcal{V}|/K \approx 2000\times$ (a reduction of $> 99.9\%$). This design is motivated by the hypothesis that the verification signal in speculative decoding is dominated by the distribution's head, rendering the storage of long-tail logits computationally redundant. In all MTD experiments, we replace the dense target $p(y_t)$ with this truncated approximation $\hat{p}_K$.

4 Experimental Setup

4.1 Task and Dataset

We evaluate on the Flores-200 benchmark (Goyal et al., 2022; Costa-Jussà et al., 2022) for machine translation. Specifically, we consider English-to-X translation for 20 diverse languages covering multiple language families and scripts: French, Spanish, German, Portuguese, Italian, Chinese, Japanese, Korean, Arabic, Turkish, Hindi, Bengali, Tamil, Urdu, Telugu, Kannada, Malayalam, Marathi, Gujarati, and Punjabi. We use the *dev* set for training and the *devtest* set for evaluation.

4.2 Models

We use **Llama-3.1-8B-Instruct**[1] (Grattafiori et al., 2024) as the target model and **Llama-3.2-1B-Instruct**[2] as the draft model. Both are instruction-tuned (Ouyang et al., 2022) for consistent behavior.

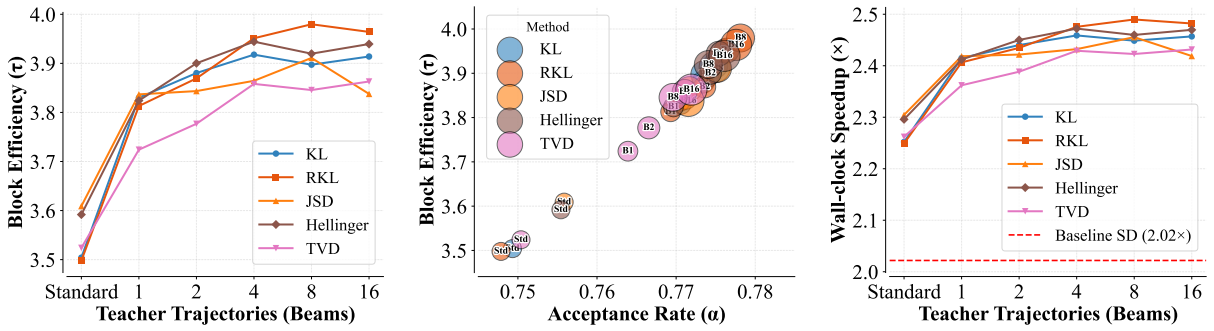
4.3 Forcing Strategies and Experimental Protocol

We employ different forcing strategies at different stages of our experiments. Forcing strategies are not mixed across settings; each experiment is designed to isolate a single factor while keeping all others fixed. Below, we specify the forcing strategy used in each experimental stage.

Forcing Strategies. We consider the following forcing strategies:

[1] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[2] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct



(a) (b) (c)

Figure 4: Decoding efficiency and speedup analysis. (a) **Efficiency scaling:** Block efficiency (τ) improves with richer teacher signals, with KL and JSD scaling best. (b) **Efficiency vs. acceptance:** Relationship between acceptance rate (α) and efficiency (τ); bubble size indicates beam width. (c) **Wall-clock speedup:** Estimated speedup over autoregressive decoding, with KL and JSD outperforming baseline speculative decoding at higher beam widths.

- **Gold**: Conditioning on human reference translations.
- **Target-Greedy**: Conditioning on sequences sampled by greedy decoding from the target model.
- **Target-Multinomial**: Conditioning on sequences sampled using Multinomial sampling (Holtzman et al., 2019) from the target model distribution.

Distribution Truncation and Temperature Analysis. All experiments analyzing Top-K truncation and temperature scaling are conducted using Gold forcing only. In this stage, we fix the forcing strategy to isolate the effects of (i) distribution effect size and (ii) temperature on distillation behavior. No target-generated forcing is used in these experiments.

Forcing Strategy Comparison. With truncation levels and temperature fixed at their optimal values, we isolate the impact of the teacher forcing strategy on distillation performance. We evaluate three distinct conditioning contexts, Gold, Target-Greedy, and Target-Multinomial, to analyze how the diversity of the teacher trajectories affects draft-target alignment when the loss function is held constant.

Supervised Fine-Tuning (SFT) Baselines. We separately train models using standard cross-entropy loss (SFT) on the Gold, Target-Greedy, and Target-Multinomial datasets. This allows us to quantify the specific benefit of distilling soft teacher distributions compared to training on hard labels (SFT), independent of the data source.

Multi-Trajectory Distillation. Beam-based distillation is performed exclusively with target-generated forcing, as beam aggregation is defined over model-generated trajectories.

4.4 Evaluation Metrics

We evaluate each distillation configuration using the following metrics:

Acceptance Rate. Following Zhou et al. (2023), we utilize the sequence-level acceptance rate $\alpha(x)$, defined as the ratio of the expected number of accepted tokens to the expected target sequence length $L_p(x)$:

$$\alpha(x) = \frac{\mathbb{E}_{y\sim p(\cdot|x)}\left[\sum_{t=1}^{|y|}\beta(x, y_{<t})\right]}{L_p(x)}, \qquad (10)$$

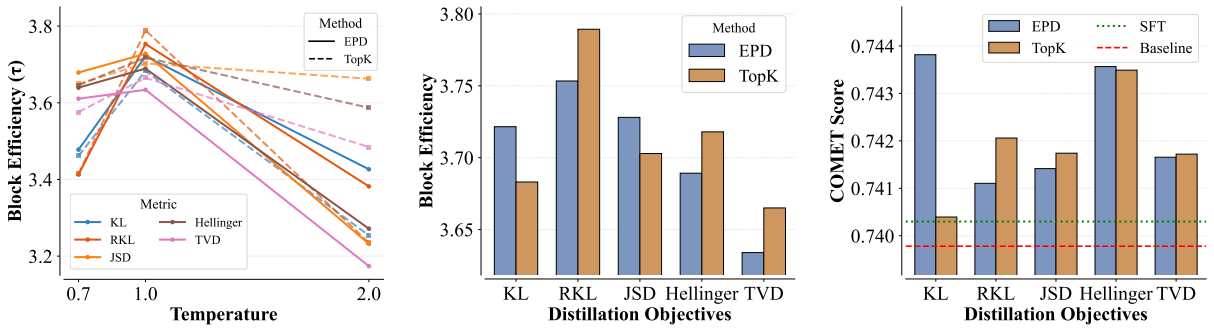
where $\beta(x, y_{<t}) = \mathbb{E}_{y_t\sim q}[\min(1, p(y_t)/q(y_t))]$ denotes the token-level acceptance probability.

Block Efficiency. For a fixed block size γ, block efficiency $\tau(x)$ measures the expected number of tokens generated per decoding step (including the target model's correction token). Under the standard assumption of i.i.d. token acceptance, this metric is derived as the sum of a truncated geometric series:

$$\tau(x) = \frac{1-\alpha(x)^{\gamma+1}}{1-\alpha(x)}. \qquad (11)$$

Speedup. The expected wall-clock speedup over standard autoregressive decoding is computed as

$$\text{Speedup} = \frac{\tau(x)}{c\gamma+1}, \qquad (12)$$



(a) (b) (c)

Figure 5: Hyperparameter and performance analysis. (a) **Temperature Sensitivity:** Distillation temperature $T = 1.0$ yields optimal block efficiency across all divergence metrics. (b) **Distillation Algorithms:** Top-50 truncation (TopK) achieves comparable efficiency to full distributions (EPD) while significantly reducing storage. (c) **Translation Quality:** COMET scores (at $T = 1.0$) indicate that KL and Reverse-KL (RKL) distillation maintain performance closest to the SFT Gold baseline.

where c denotes the ratio between draft and target model inference latency.

All metrics are averaged over 20 languages and 1000 test examples per language.

5 Results

DistillBeam consistently yields superior draft-target alignment compared to supervised fine-tuning (SFT) and standard single-trajectory distillation across all evaluated divergence objectives (Figure 2). Our optimal configuration, utilizing Reverse KL with an 8-beam width, achieves a global average block efficiency of $\tau = 3.980$ and an acceptance rate of $\alpha = 0.778$ (Table 2). This represents a significant improvement over the standard single-sequence distillation baseline ($\tau = 3.830$) and the strongest SFT baseline ($\tau = 3.566$). While Reverse KL and Hellinger distance provide the highest absolute metrics, the performance gains from multi-trajectory supervision are universal across objectives. As shown in Figure 3, distilling from teacher-sampled trajectories (Target-Multinomial) consistently outperforms training on human references (Gold) or greedy decoding across all divergence metrics.

Scaling the teacher supervision beam width N results in monotonic improvements in block efficiency (Figure 4a). Efficiency gains are steepest as N increases from 1 to 4, with performance saturating between 8 and 16 beams. Regarding storage efficiency, truncating teacher distributions to the top $K = 50$ tokens reduces offline storage requirements by 99.9% compared to full-vocabulary distillation. As shown in Figure 5b, despite this aggressive compression, the Top-K models maintain block efficiency parity with the dense-logit baselines (within 0.5% relative difference), validating the feasibility of the method under strict memory constraints.

In terms of inference latency, DistillBeam produces wall-clock speedups of 35%-65% over autoregressive decoding across the 20 evaluated languages (Figure 4c). The method demonstrates particularly strong generalization to agglutinative and complex-script languages, with the largest relative gains recorded in Marathi ($+56.1\%$), Korean ($+49.2\%$), and Turkish ($+49.1\%$) (Table 3, Figure 6). In comparison, Indo-European languages average a $+31.0\%$ improvement (detailed linguistic analysis is provided in Appendix C). Finally, hyperparameter analysis indicates that these results are robust to temperature variations around $T = 1.0$, with performance degrading only when using significantly sharper ($T = 0.7$) or more diffuse ($T = 2.0$) teacher distributions (Figure 5a).

6 Discussion

Support Coverage and Exposure Bias. The observed correlation between beam width and block efficiency demonstrates that a primary bottleneck in standard distillation is exposure bias. Single-trajectory methods condition the draft model on a narrow path, leading to narrowness when stochastic sampling introduces states unobserved during training. By aggregating supervision from multiple trajectories, DistillBeam approximates the target model's high-probability support. This reduces the covariate shift between training and inference,

ensuring alignment even when the draft model deviates from the greedy path.

Divergence Objectives and Rejection Dynamics. While the choice of divergence objective is secondary to multi-trajectory supervision, the strong performance of Reverse KL (RKL) aligns with the acceptance condition in speculative decoding. A token y is accepted when $p(y)/q(y) \geq 1$, so rejections arise primarily when the draft model overestimates probability mass ($q(y) > p(y)$). Forward KL is mass-covering and tends to spread probability into low-probability regions, increasing overestimation. In contrast, Reverse KL is mode-seeking and penalizes assigning mass where the target has little or none, directly reducing overestimation. This zero-forcing behavior improves the acceptance lower bound, leading to higher acceptance rates. We also observe that bounded divergences such as Hellinger distance yield comparable alignment by similarly penalizing distributional mismatch.

Sparsity of the Verification Signal. The stability of performance under aggressive Top-K truncation demonstrates that the effective signal for verification is highly sparse. The vocabulary's long tail adds negligible value to acceptance rates but dominates storage costs. By discarding this tail, DistillBeam overcomes the primary bottleneck of offline distillation: prohibitive memory overhead. This result transforms high-fidelity distillation from a theoretical capability into a practical solution, enabling training on massive datasets without the operational burden of managing terabytes of logit data.

Linguistic Generalization This multi-trajectory approach is particularly effective for linguistically complex settings. Agglutinative and morphologically rich languages (e.g., Marathi, Korean) exhibit the largest gains, as their high subword branching factors make single-trajectory supervision insufficient to capture the valid morphological space. By exposing the draft model to these permutations, our method suggests that multi-trajectory supervision is not merely an optimization, but a structural necessity for extending the benefits of speculative decoding to non-Latin and morphologically complex scripts.

7 Conclusion

We introduced DistillBeam, a multi-trajectory distillation framework designed to improve draft-target alignment for speculative decoding. By aggregating supervision from multiple high-probability teacher trajectories, our method mitigates the sequence drift inherent to single-path distillation. This approach is supported by Top-K truncation, which reduces storage requirements by 99.9% while preserving inference efficiency. Our empirical findings across 20 languages suggest that DistillBeam offers a scalable and principled approach for accelerating large language model inference, effectively balancing alignment quality with practical storage constraints.

Limitations and Future Work

Our study's scope is primarily focused on machine translation using the Llama-3 model family. The framework's effectiveness on other tasks, such as reasoning and code generation, remains to be validated. Furthermore, while multi-beam data generation improves alignment, it incurs a higher upfront computational cost than single-trajectory methods. We also exclusively evaluated our framework with a standard rejection sampling verifier; its interaction with more advanced mechanisms like token-tree verification is unexplored.

Future work should address these limitations. First, extending DistillBeam to a broader range of tasks and model architectures will be crucial for assessing its generality. Second, investigating adaptive beam width selection during training could optimize the trade-off between distillation cost and alignment quality. Finally, combining DistillBeam with advanced verifiers like token-tree or multi-head drafting presents a promising avenue for achieving complementary and potentially compounding gains in inference efficiency.

Acknowledgments

Computational resources were provided by Lambda.ai through their research grant program. Experiments were conducted using approximately 500 GPU hours on NVIDIA A100 GPUs.

Code Availability

To ensure reproducibility, we will release the code for this work. An anonymized repository is available at: https://anonymous.4open.science/r/ACL-DistillBeam-Code-AB20

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.

Rudolf Beran. 1977. Robust location estimates. *The Annals of Statistics*, pages 431–444.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

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

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. 2024. Break the sequential dependency of llm inference using lookahead decoding. *arXiv preprint arXiv:2402.02057*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Lucien Le Cam and Grace Lo Yang. 2000. *Asymptotics in statistics: some basic concepts*. Springer Science & Business Media.

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

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

Jianhua Lin. 2002. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, and 1 others. 2023. Specinfer: Accelerating generative large language model serving with tree-based speculative inference and verification. *arXiv preprint arXiv:2305.09781*.

Tom Minka and 1 others. 2005. Divergence measures and message passing.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2023. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624.

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018. Blockwise parallel decoding for deep autoregressive models. *Advances in Neural Information Processing Systems*, 31.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. 2015. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.

Cédric Villani and 1 others. 2008. *Optimal transport: old and new*, volume 338. Springer.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *arXiv preprint arXiv:2401.07851*.

Wenda Xu, Rujun Han, Zifeng Wang, Long T Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. 2024. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. *arXiv preprint arXiv:2410.11325*.

Weilin Zhao, Yuxiang Huang, Xu Han, Wang Xu, Chaojun Xiao, Xinrong Zhang, Yewei Fang, Kaihuo Zhang, Zhiyuan Liu, and Maosong Sun. 2024. Ouroboros: Generating longer drafts phrase by phrase for faster speculative decoding. *arXiv preprint arXiv:2402.13720*.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. 2023. Distillspec: Improving speculative decoding via knowledge distillation. *arXiv preprint arXiv:2310.08461*.

A Method

A.1 Description of Divergence Functions

In this work, we evaluate alignment using discrete probability distributions $p(\cdot|x)$ (target) and $q_\theta(\cdot|x)$ (draft) over a vocabulary $\mathcal{V}$.

Forward Kullback-Leibler (FKL) Divergence.

$$D_{\text{FKL}}(p\|q_\theta) = \sum_{y\in\mathcal{V}} p(y) \log \frac{p(y)}{q_\theta(y)} \tag{A.1}$$

Minimizing FKL corresponds to Maximum Likelihood Estimation (MLE). This objective is **mass-covering** (also known as zero-avoiding for the student); $q_\theta(y)$ is forced to be non-zero wherever $p(y) > 0$. For a capacity-constrained draft model, this often leads to "mode averaging," where the student assigns low probability to the entire support of p rather than committing to specific modes, resulting in high entropy.

Reverse Kullback-Leibler (RKL) Divergence.

$$D_{\text{RKL}}(p\|q_\theta) := D_{\text{KL}}(q_\theta\|p) = \sum_{y\in\mathcal{V}} q_\theta(y) \log \frac{q_\theta(y)}{p(y)} \tag{A.2}$$

RKL is **mode-seeking** (or zero-forcing for the student). The term is weighted by the student's probability $q_\theta(y)$. If $p(y) \to 0$, then $q_\theta(y)$ must tend to 0 to minimize the objective. This imposes a heavy penalty on *false positives* (regions where q_θ has mass but p does not).

Jensen-Shannon Divergence (JSD).

$$D_{\text{JS}}(p\|q_\theta) = \frac{1}{2}D_{\text{KL}}(p\|m) + \frac{1}{2}D_{\text{KL}}(q_\theta\|m), \quad \text{where } m = \frac{1}{2}(p + q_\theta) \tag{A.3}$$

JSD is the symmetrized, smoothed arithmetic average of forward and reverse KL divergences from the mixture distribution m. It is bounded $[0, \ln 2]$, providing gradient stability when distributions have disjoint support.

Hellinger Distance (HEL).

$$D_{\text{HEL}}(p, q_\theta) = \frac{1}{\sqrt{2}}\sqrt{\sum_{y\in\mathcal{V}}(\sqrt{p(y)} - \sqrt{q_\theta(y)})^2} \tag{A.4}$$

Hellinger distance is a metric satisfying the triangle inequality. Because it operates on the square root of probabilities, it is less sensitive to extreme outliers in the tail of the distribution compared to KL-based measures.

Total Variation Distance (TVD).

$$D_{\text{TVD}}(p, q_\theta) = \frac{1}{2}\sum_{y\in\mathcal{V}} |p(y) - q_\theta(y)| \tag{A.5}$$

TVD measures the L_1 distance between probability vectors. It represents the maximal difference in probabilities the two models can assign to the same event.

A.2 Theoretical Analysis: Acceptance Rate & Beam Coverage

Here we provide the mathematical justification for selecting Reverse KL and Multi-Trajectory Distillation to optimize the acceptance rate α.

A.2.1 Divergence and Acceptance Probability

Let the token-level acceptance probability for a candidate y sampled from q_θ be $\beta(y|x) = \min(1, \frac{p(y|x)}{q_\theta(y|x)})$. The objective of Speculative Decoding is to maximize the expected acceptance rate:

$$\alpha(x) = \mathbb{E}_{y \sim q_\theta(\cdot|x)} \left[\min \left(1, \frac{p(y|x)}{q_\theta(y|x)} \right) \right]. \tag{A.6}$$

Proposition A.1 (False Positive Minimization). *Maximizing the expected acceptance rate $\alpha(x)$ is strictly equivalent to minimizing the probability mass assigned by the draft model to the set of "over-estimated" tokens (False Positives).*

Proof. Let us partition the vocabulary $\mathcal{V}$ into two disjoint sets based on the likelihood ratio:

$$\mathcal{V}_{\leq} = \{y \in \mathcal{V} \mid q_\theta(y) \leq p(y)\} \quad \text{(Under-estimated or exact)} \tag{A.7}$$

$$\mathcal{V}_{>} = \{y \in \mathcal{V} \mid q_\theta(y) > p(y)\} \quad \text{(Over-estimated / False Positives)} \tag{A.8}$$

We expand the expectation in Eq. (A.5):

$$\alpha(x) = \sum_{y \in \mathcal{V}} q_\theta(y) \min \left(1, \frac{p(y)}{q_\theta(y)} \right) \tag{A.9}$$

$$= \sum_{y \in \mathcal{V}_{\leq}} q_\theta(y)(1) + \sum_{y \in \mathcal{V}_{>}} q_\theta(y) \left(\frac{p(y)}{q_\theta(y)} \right) \tag{A.10}$$

$$= \sum_{y \in \mathcal{V}_{\leq}} q_\theta(y) + \sum_{y \in \mathcal{V}_{>}} p(y). \tag{A.11}$$

Since $\sum_{y \in \mathcal{V}} q_\theta(y) = 1$, we can write $\sum_{y \in \mathcal{V}_{\leq}} q_\theta(y) = 1 - \sum_{y \in \mathcal{V}_{>}} q_\theta(y)$. Substituting this back:

$$\alpha(x) = \left(1 - \sum_{y \in \mathcal{V}_{>}} q_\theta(y) \right) + \sum_{y \in \mathcal{V}_{>}} p(y) \tag{A.12}$$

$$\alpha(x) = 1 - \underbrace{\sum_{y \in \mathcal{V}_{>}} (q_\theta(y) - p(y))}_{\text{Net Excess Mass in False Positive Region}}. \tag{A.13}$$

Equation (A.11) demonstrates that $\alpha(x)$ is maximized if and only if the net excess probability mass in $\mathcal{V}_{>}$ is minimized. □

Connection to Reverse KL: The Reverse KL divergence $D_{\text{KL}}(q_\theta \| p) = \sum q_\theta(y) \log \frac{q_\theta(y)}{p(y)}$ is dominated by terms where $q_\theta(y) \gg p(y)$ (i.e., tokens in $\mathcal{V}_{>}$). In these regions, the ratio $\frac{q_\theta}{p}$ is large, leading to a large penalty. Conversely, Forward KL $D_{\text{KL}}(p \| q_\theta)$ is dominated by regions where $p(y) \gg q_\theta(y)$ (False Negatives), which do *not* negatively impact the acceptance rate for generated tokens (as shown in Eq. A.9, tokens in $\mathcal{V}_{\leq}$ are always accepted). Thus, RKL is geometrically aligned with Eq. (A.11).

A.2.2 Support Coverage via Multi-Trajectory Distillation

Proposition A.2 (Prevention of Support Collapse). *Single-trajectory distillation on the mode (greedy target) induces Support Collapse, restricting the draft model to a proper subset of the target's support. Multi-Trajectory Distillation approximates the union of high-probability supports, minimizing sequence drift.*

Proof (Sketch). Let the target distribution p be multi-modal. A greedy teacher creates a dataset $\mathcal{D}_{greedy} = \{(x, y^*)\}$ where $y^* = \text{argmax}_y p(y|x)$. Minimizing divergence against a deterministic target (Dirac delta δ_{y^*}) yields the optimal student $q^*(y) = \delta_{y^*}(y)$.

$$\text{supp}(q^*) = \{y^*\} \subset \text{supp}(p). \tag{A.14}$$

However, during inference, the target model p samples from its full support. If p samples a valid token $y' \in \text{supp}(p)$ such that $y' \neq y^*$, the student q^* (having zero mass on y') incurs an infinite rejection penalty. More critically, for the subsequent token $t+1$, the student is conditioned on context (x, y'), a state it has never observed during training (Covariate Shift).

In Multi-Trajectory Distillation, the target is defined as a mixture over beams $\mathcal{B}$: $\hat{p}(y) \propto \sum_{y^{(i)} \in \mathcal{B}} p(y|x, y^{(i)}_{<t})$. The optimal student q_{MTD} satisfies:

$$\text{supp}(q_{MTD}) \approx \bigcup_{y^{(i)} \in \mathcal{B}} \text{supp}(p(\cdot|\dots y^{(i)})). \tag{A.15}$$

This ensures q_θ places non-zero mass on all likely paths. Consequently, if the target model samples a non-greedy token y' present in the beams, $q_\theta(y'|x)$ is non-negligible, preventing rejection, and the student has been trained on the trajectory continuation given y', minimizing drift. □

A.3 DistillBeam Algorithms

We formally present the inference and training procedures. Algorithm 1 details the standard Speculative Decoding step with rejection sampling. Algorithm 2 details the **Online DistillBeam** training procedure. Unlike standard distillation which uses a fixed dataset, this algorithm generates diverse teacher trajectories (beams) on-the-fly, ensuring the student is continuously aligned with the target's structural support.

Algorithm 1 Speculative decoding step

Require: Target model $\mathcal{M}_p$, draft model $\mathcal{M}_q$, context $\rho = \{x, y_{<t}\}$.
Require: Block size γ, Random uniform generator $\mathcal{U}[0, 1]$.

$q_t(y), \ldots, q_{t+\gamma-1}(y) \leftarrow \emptyset$
for $i = 0$ to $\gamma - 1$ **do**
  $q_{t+i}(y) \leftarrow \mathcal{M}_q(y \mid x, y_{<t+i})$ ▷ Draft autoregressively.
  $y_{t+i} \sim q_{t+i}(y)$ ▷ Sample candidate token.
end for
$(p_t(y), \ldots, p_{t+\gamma}(y)) \leftarrow (\mathcal{M}_p(\cdot|x, y_{<t}), \ldots, \mathcal{M}_p(\cdot|x, y_{<t+\gamma}))$ ▷ Run $\mathcal{M}_p$ in parallel.
$u_t \sim \mathcal{U}[0, 1], \ldots, u_{t+\gamma-1} \sim \mathcal{U}[0, 1]$ ▷ Generate γ random values.
$n \leftarrow \min\left(\{i \mid 0 \le i < \gamma, u_{t+i} > \min(1, \frac{p_{t+i}(y_{t+i})}{q_{t+i}(y_{t+i})})\} \cup \{\gamma\}\right)$ ▷ Find first rejection index.
if $n < \gamma$ **then**
  $p'_{\text{adj}}(y) \leftarrow \max(0, p_{t+n}(y) - q_{t+n}(y))$ ▷ Compute residual distribution.
  $y_{t+n} \sim p'_{\text{adj}}(y) / \sum_{z \in \mathcal{V}} p'_{\text{adj}}(z)$ ▷ Resample from adjusted distribution.
else
  $y_{t+n} \sim p_{t+n}(y)$ ▷ Accept all; sample next step from $\mathcal{M}_p$.
end if
return $\{x, y_{<t+n+1}\}$ ▷ Append n accepted tokens plus one corrected token.

Algorithm 2 Online DistillBeam (Multi-Trajectory Distillation)

Require: Target $\mathcal{M}_p$, Student $\mathcal{M}_q^\theta$, Dataset $\mathcal{D}$.
Require: Beam Width N, Top-K K, Divergence $\mathcal{D}_{\text{div}}(\cdot\|\cdot)$.

while not converged **do**
  Sample batch of inputs X_{batch} from $\mathcal{D}$
  $\mathcal{L}_{\text{total}} \leftarrow 0$
  for each input $x \in X_{\text{batch}}$ **do**
    $\mathcal{B}_N \leftarrow \text{BeamSearch}(\mathcal{M}_p, x, N)$ ▷ Generate diverse teacher trajectories on-the-fly.
    $\mathcal{L}_{\text{MTD}} \leftarrow 0$
    for each beam $y^{(b)} \in \mathcal{B}_N$ **do** ▷ Aggregate gradient over all beams.
      for step t in $y^{(b)}$ **do**
        $\mathbf{z}_p \leftarrow \mathcal{M}_p(\cdot \mid x, y^{(b)}_{<t})$ ▷ Teacher forward pass.
        $\hat{p}_K \leftarrow \text{TopK\_Renorm}(\mathbf{z}_p, K)$ ▷ Compute sparse target distribution.
        $q_\theta \leftarrow \mathcal{M}_q^\theta(\cdot \mid x, y^{(b)}_{<t})$ ▷ Student forward pass.
        $\mathcal{L}_{\text{MTD}} \leftarrow \mathcal{L}_{\text{MTD}} + \mathcal{D}_{\text{div}}(\hat{p}_K \| q_\theta)$ ▷ Accumulate generalized divergence.
      end for
    end for
    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{total}} + \frac{1}{N}\mathcal{L}_{\text{MTD}}$
  end for
  $\theta \leftarrow \theta - \eta \frac{1}{|X_{\text{batch}}|} \nabla_\theta \mathcal{L}_{\text{total}}$ ▷ Update parameters via backpropagation.
end while
return $\mathcal{M}_q^\theta$

B Implementation Details

B.1 Datasets

We conduct all experiments on the **Flores-200** machine translation benchmark. This dataset was selected due to its wide coverage of linguistic families, scripts, and difficulty levels, providing a robust testbed for generalization.

- **Languages:** We select 20 diverse target languages covering high-resource, mid-resource, and lower-resource/complex-script languages: French, Spanish, German, Portuguese, Italian, Chinese, Japanese, Korean, Arabic, Turkish, Hindi, Bengali, Tamil, Urdu, Telugu, Kannada, Malayalam, Marathi, Gujarati, and Punjabi.
- **Splits:** We utilize the standard `dev` set for training the draft model (distillation) and the `devtest` set for evaluation. Each split contains approximately 1,000 parallel sentences per language.
- **Preprocessing:** Input text is formatted using the standard Llama-3 instruction template: `<|begin_of_text|><|start_header_id|>user<|end_header_id|> Translate to [Language]: [Sentence]...`

B.2 Models

All experiments utilize the Llama-3 family of models, ensuring tokenizer compatibility between draft and target models (vocabulary size $|\mathcal{V}| = 128,256$).

- **Target Model (Teacher):** `Llama-3.1-8B-Instruct`. This model serves as the verification oracle. It utilizes Grouped Query Attention (GQA) and RoPE embeddings.
- **Draft Model (Student):** `Llama-3.2-1B-Instruct`. This model is approximately $8\times$ smaller than the target. It is initialized with pre-trained weights and fine-tuned via distillation.

B.3 Distillation Hyperparameters

The draft model is trained using the **DistillBeam** framework implemented in PyTorch. We utilize 8 $\times$ NVIDIA A100 (80GB) GPUs for data generation and training.

Hyperparameter	**Value**
Optimizer	AdamW
Optimizer Betas	$(0.9, 0.999)$
Optimizer Epsilon	1×10^{-8}
Learning Rate	2×10^{-5}
Weight Decay	0.0
LR Scheduler	Linear Warmup + Cosine Decay
Batch Size	32 sequences
DistillBeam Specifics	
Beam Width (N)	16
Top-K Truncation (K)	50
Temperature (T)	1.0
Loss Function	Reverse KL

Table 1: Hyperparameters used for Multi-Trajectory Distillation.

B.4 Evaluation Metrics

We adopt the efficiency metrics formalized by Zhou et al. (2023) to accurately proxy wall-clock performance.

Sequence-Level Acceptance Rate (α). Because speculative decoding is stochastic, we utilize the **sequence-level acceptance rate** $\alpha(x)$. This metric is defined as the ratio of the expected number of accepted tokens to the expected target sequence length. For a given input x:

$$\alpha(x) := \frac{\mathbb{E}[\text{number of accepted tokens in generating } y]}{\mathbb{E}[\text{number of tokens in } y]} = \frac{\mathbb{E}_{y\sim p(\cdot|x)}\left[\sum_{t=1}^{|y|}\beta(x, y_{<t})\right]}{L_p(x)}, \tag{A.16}$$

where $L_p(x) := \mathbb{E}_{y\sim p(\cdot|x)}[|y|]$ denotes the expected length of the target output, which is invariant to the choice of draft model. The term $\beta(x, y_{<t})$ represents the token-level acceptance probability:

$$\beta(x, y_{<t}) := \mathbb{E}_{y_t\sim q_\theta(y_t)}\left[\min\left(1, \frac{p(y_t)}{q_\theta(y_t)}\right)\right].$$

This formulation is preferred because the expected total number of rejected tokens, given by $(1 - \alpha(x)) \cdot L_p(x)$, directly lower-bounds the expected number of speculative decoding steps.

Block Efficiency (τ). In practice, speculative decoding utilizes a fixed block size γ. A practical measure of efficiency is the **block efficiency** $\tau(x)$, defined as the expected number of accepted tokens per decoding block. Assuming i.i.d. token-level acceptance rates, $\tau(x)$ is derived from the sequence-level acceptance rate α as:

$$\tau(x) = \frac{1 - \alpha(x)^{\gamma+1}}{1 - \alpha(x)}. \tag{A.17}$$

For a fixed γ, the theoretical maximum is $\tau(x) = \gamma + 1$, corresponding to the case where all drafted tokens are accepted and verified by the target model in a single pass.

Wall-Clock Speedup. We estimate the expected wall-clock speedup relative to standard autoregressive decoding as:

$$\text{Speedup} = \frac{\tau(x)}{c\gamma + 1}, \tag{A.18}$$

where c represents the relative latency coefficient, defined as the ratio between the inference latency of the draft model ($\mathcal{M}_q$) and the target model ($\mathcal{M}_p$).

C Additional Results and Analysis

In this section, we provide a detailed decomposition of the performance gains achieved by the **DistillBeam** framework. We analyze the global performance of different distillation objectives using the metrics defined in Appendix B.4, examine the impact of linguistic typology on inference acceleration, and provide a language-specific breakdown of optimal configurations.

C.1 Global Objective Performance

Table 2 presents the comparative performance of the evaluated experimental configurations, averaged across all 20 languages. The results indicate a clear hierarchy in alignment efficiency based on the choice of divergence objective and the density of teacher supervision.

Impact of Divergence Objectives. Consistent with the theoretical analysis regarding mode-seeking behavior, **Reverse KL** consistently yields the highest block efficiency (τ). The configuration **Reverse KL (8 beams)** achieves the highest global performance ($\tau = 3.980$, $\alpha = 0.778$). This suggests that minimizing the reverse KL divergence effectively penalizes the draft model for assigning probability mass to tokens rejected by the teacher, thereby minimizing false positives during rejection sampling.

Multi-Trajectory vs. Single-Trajectory. The results demonstrate a positive correlation between beam width and distillation efficiency. For every divergence objective (Reverse KL, Forward KL, Hellinger), multi-beam variants (4, 8, 16 beams) outperform their single-beam counterparts. For example, standard **Forward KL** improves from $\tau = 3.830$ (1 beam) to $\tau = 3.914$ (16 beams). This confirms that aggregating supervision from multiple high-probability teacher trajectories provides a more robust signal than distilling from a single greedy path.

Rank	Experiment Configuration	Acceptance Rate (α)	Block Efficiency (τ)
DistillBeam (Multi-Trajectory Framework)			
1	**Reverse KL (8 beams)**	0.778	3.980
2	Reverse KL (16 beams)	0.778	3.964
3	Reverse KL (4 beams)	0.776	3.951
4	Hellinger (4 beams)	0.775	3.944
5	Hellinger (16 beams)	0.776	3.939
6	Hellinger (8 beams)	0.774	3.920
7	KL (4 beams)	0.775	3.918
8	KL (16 beams)	0.775	3.914
9	JSD (8 beams)	0.775	3.911
10	Hellinger (2 beams)	0.774	3.900
Standard Distillation (Single-Trajectory Baselines)			
24	**Reverse KL (Top-K, T=1)**	0.768	3.789
30	**KL (EPD, T=1)**	0.766	3.722
32	Hellinger (Top-K, T=1)	0.765	3.718
35	JSD (Top-K, T=1)	0.764	3.703
44	TVD (Top-K, T=1)	0.761	3.665
Standard Supervised Fine-Tuning & No Distillation			
54	SFT (Multinomial Target)	0.758	3.566
55	SFT (Greedy Target)	0.756	3.561
63	SFT (Gold Reference)	0.744	3.316
69	**Baseline (No Distillation)**	0.728	3.044

Table 2: Global performance of distillation strategies sorted by Block Efficiency (τ). The column α denotes the sequence-level acceptance rate defined in Eq. (1).

C.2 Linguistic Categorization and Generalization

To understand the robustness of the framework, we analyzed performance gains across varying linguistic categories. Figure 6 visualizes the improvement over baseline grouped by script, morphological typology, region, and language family.

Language	Baseline τ	Best Configuration	Best τ	% Gain
Marathi (mar_Deva)	1.706	JSD (8 beams)	2.664	+56.1%
Korean (kor_Hang)	1.181	Reverse KL (16 beams)	1.761	+49.2%
Turkish (tur_Latn)	1.320	Reverse KL (8 beams)	1.968	+49.1%
Malayalam (mal_Mlym)	3.624	Reverse KL (8 beams)	5.288	+45.9%
Portuguese (por_Latn)	3.600	Reverse KL (8 beams)	5.102	+41.7%
Arabic (arb_Arab)	1.361	Hellinger (16 beams)	1.876	+37.9%
Italian (ita_Latn)	3.182	TVD (4 beams)	4.363	+37.1%
Punjabi (pan_Guru)	4.044	Reverse KL (8 beams)	5.533	+36.8%
Urdu (urd_Arab)	2.489	KL (16 beams)	3.373	+35.5%
French (fra_Latn)	3.397	Hellinger (1 beam)	4.582	+34.9%
Kannada (kan_Knda)	4.271	TVD (16 beams)	5.738	+34.3%
German (deu_Latn)	2.864	JSD (8 beams)	3.832	+33.8%
Bengali (ben_Beng)	3.943	Hellinger (8 beams)	5.253	+33.2%
Gujarati (guj_Gujr)	4.262	Reverse KL (16 beams)	5.625	+32.0%
Chinese (zho_Hans)	1.770	Reverse KL (16 beams)	2.319	+31.0%
Tamil (tam_Taml)	3.964	KL (2 beams)	5.083	+28.2%
Hindi (hin_Deva)	3.103	Hellinger (16 beams)	3.968	+27.9%
Spanish (spa_Latn)	4.073	Reverse KL (16 beams)	5.190	+27.4%
Japanese (jpn_Jpan)	1.694	JSD (2 beams)	2.157	+27.4%
Telugu (tel_Telu)	5.027	Hellinger (4 beams)	6.386	+27.0%

Table 3: Detailed breakdown of the best-performing distillation configuration for each language. **% Gain** denotes the relative improvement in Block Efficiency (τ) over the Baseline.

Writing Script. We observe significant improvements in non-Latin scripts, particularly **Hangul** (+47.9%) and **Malayalam** (+45.9%). Scripts with higher visual complexity or non-concatenative features (like Devanagari and Gurmukhi) see gains consistently above 35%, whereas Latin-script languages average around 33.2%.

Morphological Typology. A strong distinction is observed based on morphological complexity. **Agglutinative** languages (e.g., Turkish, Korean, Tamil) show the highest average improvement of **35.0%**. In these languages, words are formed by stringing together morphemes, leading to a high perplexity for smaller models. DistillBeam effectively mitigates the "cascading drift" often seen in these languages.

C.3 Per-Language Configuration Analysis

Table 3 details the optimal experimental configuration for each of the 20 evaluated languages. While Reverse KL is the top-performing objective on average, we observe linguistic heterogeneity in the optimal choices.

Highest Improvements. The largest relative gains are observed in languages where the baseline draft model performance is lowest. **Marathi** shows a +56.1% improvement, followed by **Korean** (+49.2%) and **Turkish** (+49.1%). In these cases, multi-trajectory distillation is critical for enabling practical speedups.

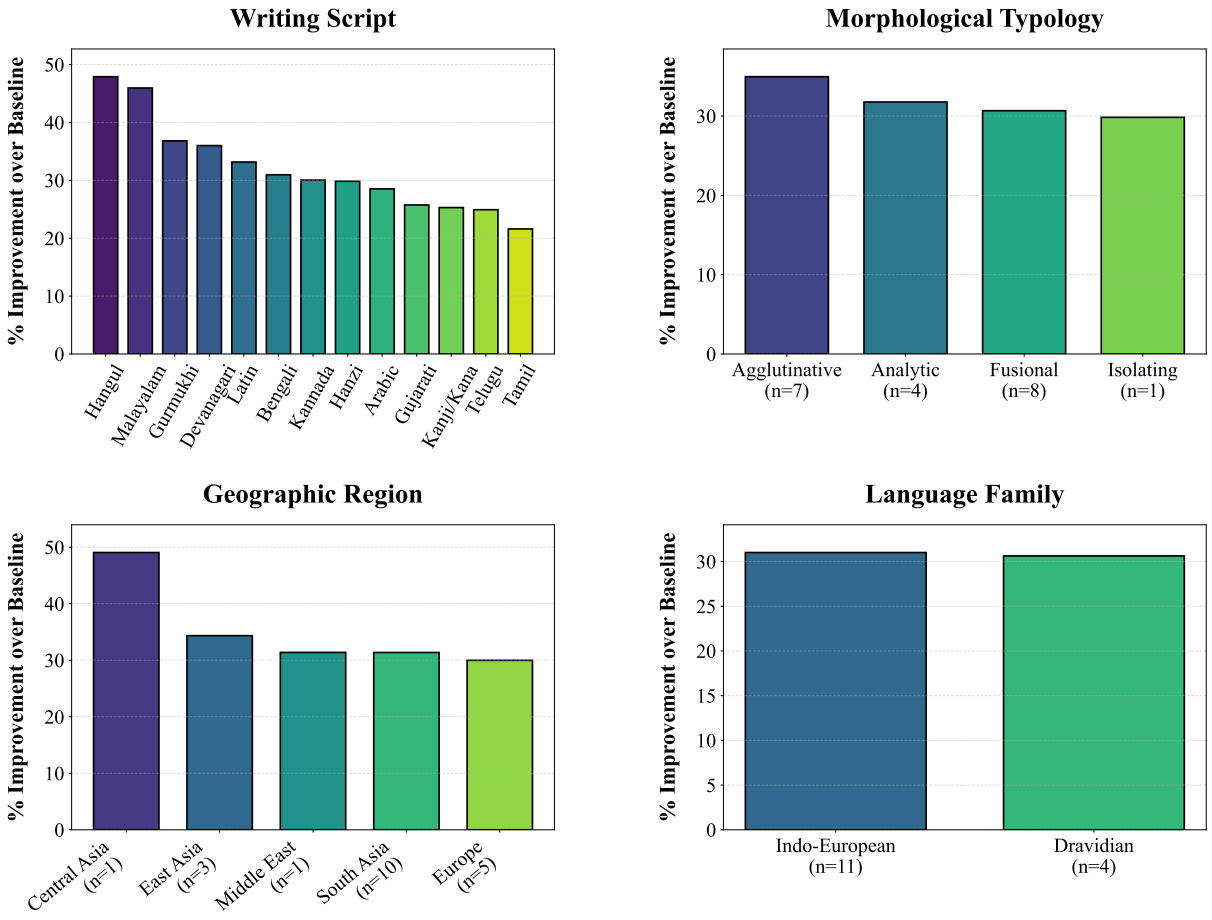


Figure 6: Fine-grained analysis of inference speedup across 20 languages, categorized by (a) Writing Script, (b) Morphological Typology, (c) Geographic Region, and (d) Language Family, demonstrating robust generalization.