RESEARCH

Boosting DeepSeek-R1's Speed with Customized Speculative Decoding

MAY 12, 2025 · BY WAI TONG CHUNG, DAN WATERS, AVNER MAY, BEN ATHIWARATKUN

TLDR: In this blog post, we show that using a custom speculator—trained on your own Deepseek-R1 inference traffic—can yield 1.23-1.45x speedups during decoding (tokens/second), and ~25% reduction in overall cost (same throughput with fewer GPU-hours), relative to Together's state-of-the-art base speculator. This translates to 1.85-2.97x speedup and ~55% cost reductions when compared to conventional next token prediction. Please reach out to our sales team to learn how to get started with a custom speculator on your dedicated endpoint today!

Optimizing a Frontier Open-Weight Model: DeepSeek-R1

DeepSeek-R1 took the world by storm this past January. It is an open-weight large language model (LLM) that performs on par with frontier proprietary models across a wide range of complex tasks, while being trained at a fraction of the cost.

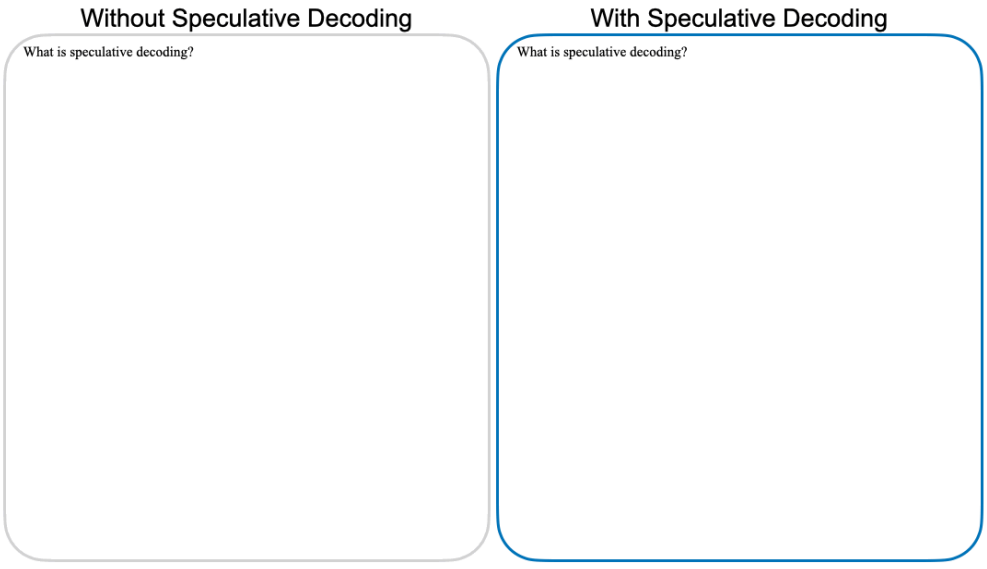
Although training R1 was relatively inexpensive, R1 inference can still be quite costly and slow due to (i) the model's large number of parameters (671B) and (ii) extensive chain-of-thought "thinking" required to generate a well-reasoned response. Thus, optimizing Deepseek-R1's speed and cost is critical for making it an attractive option in many latency or cost-sensitive real-world applications. Examples of these applications include social media or customer support engagement, which require human-response speeds, or batch enterprise applications such as judging résumés that require effective

costs for processing tens of thousands of documents for any given job posting.

One important technique we leverage to speed up inference for DeepSeek-R1 (and the other models on our platform!) is *speculative decoding*—this is a method that optimizes the speed and cost of our serverless and dedicated inference endpoints, and for which we have advanced the state-of-the-art in the field (e.g., Medusa, Sequoia, and SpecExec). A great feature of speculative decoding is its ability to gain higher speedups when the speculator is tailored to a specific domain of interest—for example, by being fine-tuned on data from that domain. In this blog post, we will show the cost and speed benefits of training custom speculators. But first, here is an overview of how speculative decoding works.

A Speculative Decoding Primer

Without any optimizations, large language models (LLMs) are limited to decoding one token at a time during inference, as each token generation requires a full forward pass through the model. This sequential process can be slow, especially for models with billions of parameters, because generating each new token requires the slow operation of moving the entire model from the GPU's storage to its compute cores.



A demonstration of R1 inference in action, without speculative decoding on the left, and with our proprietary speculator on the right. On the left, we show R1's "thinking" tokens in blue and non-"thinking" tokens in black. Additionally, on the right, rejected tokens (corrected by the verifier LLM) in red. As you can see, speculative decoding speeds up inference by 2.3x in this case, as a majority of speculated tokens are accepted.

Speculative decoding can speed up LLM decoding by using a smaller, faster "*speculator" model* to speculate the next few tokens, which are then verified in parallel by the larger "target" model (that is being sped up). For example, an

8B model can quickly generate a sequence of future tokens, which the 671B model can efficiently verify in a single forward pass by comparing the draft model's token probabilities to its own. The sequence of speculated tokens can be fully or partially accepted, or rejected entirely. A strong speculator, such as R1's MTP module, can boost token generation speeds by ~1.5x, making it a powerful optimization technique for generative AI systems.

A great speculator has the following properties:

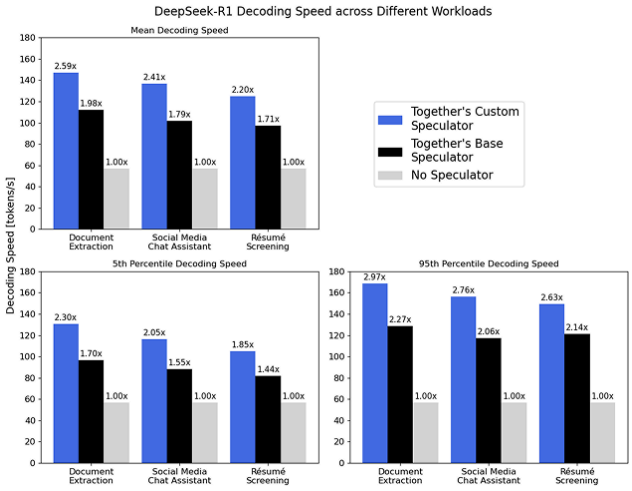
- **Speed**: It is fast, minimizing the overhead of running speculative decoding. This often requires smaller and less expressive language models.
- **Alignment with target model**: It predicts similar outputs to the larger target LLM for the domain of interest. For serverless endpoints, achieving this alignment is particularly challenging due to the vast range of possible workloads, requiring highly robust speculators. In contrast, our dedicated endpoint customers often have a narrower distribution of workloads, allowing us to fine-tune our speculators to closely match the target LLM's outputs on this type of data.

At Together, we optimize our speculators across both of these axes, navigating the trade-offs between the speculator's speed and alignment to attain the best end-to-end speedups. Based on our research team's efforts, we can optimize speculative decoding across the 200+ models hosted on our platform.

Blazing fast R1 speed using custom speculators

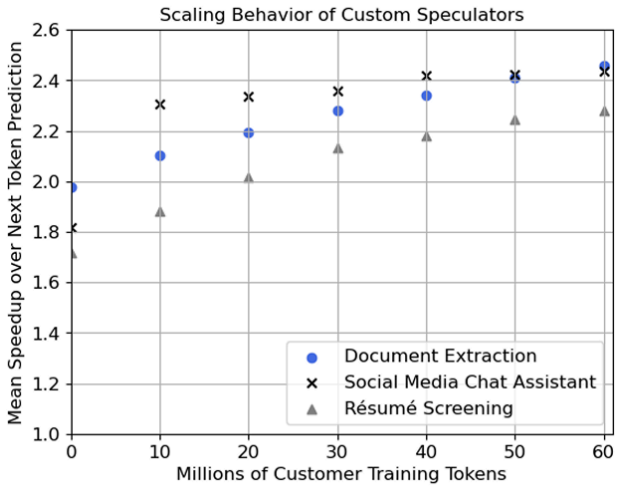
We now show the speedups attained by our state-of-the-art Base and Custom Speculator on three different R1 inference workloads. These workloads—document extraction, a social media chat assistant, and résumé screening—are examples of real-world cost- and latency-sensitive applications that are important to a few of our customers.

As seen below, our Base Speculators allow us to attain approximately 1.44-2.27x over conventional next-token prediction across 3 different R1 customer workloads. We also demonstrate the effectiveness of further customizing speculators for these three workloads: as you can see, Together's Custom Speculators attain speeds of ~100-170 tokens per second, corresponding to speedups of 1.23-1.45x over the Together Base Speculator—with overall speedups of 1.85x-2.97x over next-token predictions. We train these Custom Speculators with our proprietary training pipeline, which finetunes our Base Speculator with data from each workload.



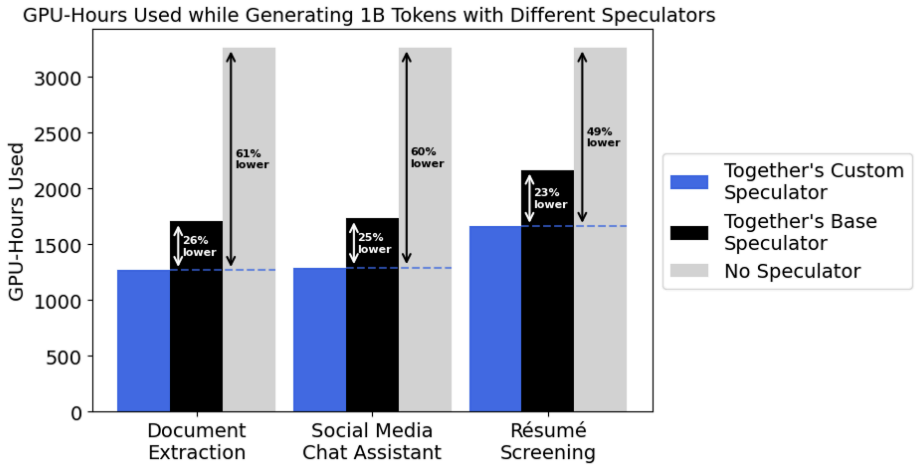
Speedup factors attained by Together's Custom Speculators, fine-tuned for three different R1 customer use cases. We compare their mean, along with 5th and 95th percentile, performance to the Together Base Speculator, and no speculator. Our Custom Speculators yield speedups of 1.23x-1.45x over our Base Speculator. Inference measurements are obtained through hold-out sets from each user's traffic at the low latency regime.

As customers continue to utilize our inference platform, we can leverage the growing volume of inference traffic to enhance our speculator's performance. This, in turn, leads to increased speedups for our customers, particularly those using our dedicated inference products. For instance, with just 20M tokens (approximately 10K prompt-response pairs), our DeepSeek R1 speculator achieves >1.10x speedup over our Base Speculator. Moreover, with 50M tokens, we can achieve >1.20x speedup. As our customers (especially those using our dedicated inference products) continue to use our API, we can tap into this growing dataset to drive further improvements.



Speedup factors attained by Together's Custom Speculators, fine-tuned for three different R1 customer use cases at the low latency regime, as a function of millions of customer training tokens. All Custom Speculators are trained from Together's Base Speculators. 20M tokens is approximately 10K prompt-response pairs at 2048 sequence length.

In addition to these large latency improvements, Custom Speculators can also increase the overall throughput we attain per GPU, thereby lowering overall inference costs by allowing us to serve the same amount of traffic on fewer GPUs. This is particularly important for enterprise customers that rely on AI systems to process huge numbers of requests per day. In the figure below showing mean GPU cost, we show that by using Custom Speculators, we can reduce the number of GPU hours (and thus, overall cost) needed for generating 1B tokens by 23%-26% relative to the Base Speculator, and 49%-61% relative to without speculative decoding.



Mean DeepSeek-R1 inference cost (in GPU hours) estimated from Custom, Base, and without speculators. Costs are estimated through inference measurements on hold-out sets from Together's user traffic at the high throughput regime.

This work is an investigation from our Turbo Research Team, which focuses on inference efficiency, including speculative decoding and model optimizations. Look forward to more efficiency improvements from our team in the future!

Summary

Open-Source Powerhouse: Together AI serves DeepSeek-R1, which matches closed-sourced frontier models in performance.

2-3x Faster Inference: Our speculative decoding techniques dramatically speed up token generation, reducing latency and improving throughput.

Custom Optimization: Fine-tuned Custom Speculators boost speed by an additional 1.23x–1.45x over our state-of-the-art base speculator for your specific workload.

Lower Costs, Higher Efficiency: Reduce GPU costs by 23%-26% vs. standard speculative decoding and 49%-61% vs. no optimization.

Scalable & Adaptive: Performance improves with more data—just 20M tokens (~10K prompts) gives >1.10x speedup.

Dedicated Endpoint Advantage: Get workload-specific optimizations for maximum efficiency on Together AI's dedicated instances.

Reach out to us

If you are interested in exploring custom speculators for your workloads, or would like to learn more about how our world-class inference optimization works, we invite you to get in touch with our Customer Experience team.

together.ai