FROM TOKENS TO LAYERS: REDEFINING STALL-FREE SCHEDULING FOR MOE SERVING WITH LAYERED PREFILL

Gunjun Lee [1] **Jiwon Kim** [1] **Jaiyoung Park** [1] **Younjoo Lee** [1] **Jung Ho Ahn** [1]

ABSTRACT

Large Language Model (LLM) inference in production must meet stringent service-level objectives for both time-to-first-token (TTFT) and time-between-token (TBT) while maximizing throughput under fixed compute, memory, and interconnect budgets. Modern serving systems adopt stall-free scheduling techniques such as chunked prefill, which splits the processing of long prompts along the token dimension and interleaves prefill with ongoing decode iterations. While effective at stabilizing TBT, chunked prefill incurs substantial overhead in Mixture-of-Experts (MoE) models: redundant expert weight loads increase memory traffic by up to **39%** and inflate energy consumption. We propose **layered prefill**, a new scheduling paradigm that treats transformer layer groups as the primary scheduling unit, specifically targeting MoE serving. By vertically partitioning the model into contiguous layer groups and interleaving prefill and decode across the groups, layered prefill sustains stall-free decoding while eliminating chunk-induced MoE weight reloads. It reduces off-chip bandwidth demand, lowering TTFT by up to **70%**, end-to-end latency by **41%** and per-token energy by up to **22%**. Evaluations show that layered prefill consistently improves the TTFT–TBT Pareto frontier over chunked prefill, reducing expert-load traffic and energy cost while maintaining stall-free decoding. Overall, shifting the scheduling axis from tokens to layers unlocks a new operating regime for high-efficiency, energy-aware MoE serving in co-located environments.

1 INTRODUCTION

Transformer-based (Vaswani et al., 2017) large language models (LLMs) have demonstrated remarkable performance across domains once thought to require exclusively human capabilities, including question answering, code generation, and mathematical problem solving (DeepSeek-AI et al., 2025; Yang et al., 2025; OpenAI et al., 2025). Beyond solving discrete tasks, they are now widely used to assist or automate work ranging from complex reasoning to repetitive operations. Rapid adoption of LLMs has led cloud providers to build massive accelerator clusters (AI factories), and has simultaneously sparked intensive systems research on LLM serving (Pope et al., 2022; Zhong et al., 2024; Mitra et al., 2025; Yun et al., 2025; Yu et al., 2022; Agrawal et al., 2024).

LLM performance improves consistently with model size, and **Mixture-of-Experts (MoE)** has emerged as the dominant strategy for scaling. Instead of using a single feed-forward network (FFN) block, an MoE architecture employs multiple experts and a routing mechanism that assigns each token to a small subset of experts. Because only a fraction of experts are activated per token, the number of arithmetic operations per token remains comparable to a standard FFN block, while the total parameter count can grow dramatically. This sparse activation enables models to scale to trillions of parameters, which in turn translates into consistent improvements in downstream performance (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022).

LLM inference is governed by two stages that stress different hardware bottlenecks. The prefill stage processes the entire input sequence at once to create the key-value (KV) cache and generate the first output token. Since LLM weights are reused over all tokens of an input prompt, this stage is typically compute-bound. The decode stage then forwards the previous token, updates the KV cache, and generates the next output token, repeating the process one token at a time. This stage is typically memory-bound as it involves loading the key-value cache and other parameters for each token being generated. Consequently, batching is highly effective for decode, where weight reuse improves throughput, but provides little benefit for prefill, creating an inherent asymmetry in LLM serving.

LLM serving performance is primarily characterized by two latency metrics: time to first token (TTFT) and time between tokens (TBT). TTFT measures the total time from a request's arrival to the generation of its first token, encompassing both the queuing and prefill phases. TBT is the

[1]Seoul National University, Seoul, South Korea. Correspondence to: Gunjun Lee <kevin970401@snu.ac.kr>, Jung Ho Ahn <gajh@snu.ac.kr>.

Proceedings of the 9^{th} MLSys Conference, Bellevue, WA, USA, 2026.

time interval between subsequent output token generations during the decode phase, which reflects how well the system can sustain token generation once the first token has been produced. Service level objectives (SLOs) define the per-request latency budgets that a serving system must meet; for our purposes, the TTFT SLO bounds first-token latency and the TBT SLO bounds each inter-token interval.

The dual goals of maintaining stable TTFT and TBT are often in conflict. If prefill work cannot be sufficiently overlapped with decode, the additional work inflates per-step latency and triggers TBT violations. This constraint forces a trade-off: either tolerate higher TTFT (e.g., by deferring or batching prefill) or sacrifice throughput (e.g., by reducing concurrency or batch size). Thus, the core challenge in LLM serving is to push the Pareto frontier among TTFT, TBT, and throughput under fixed hardware resources.

Early serving systems such as FasterTransformer (NVIDIA, 2019) adopted a straightforward approach: process requests from start to finish in fixed batches. This guarantees stall-free decoding and stable TBT, but new requests must wait until the current batch completes, inflating TTFT. More fine-grained schedulers, such as those in Orca (Yu et al., 2022) and vLLM (Kwon et al., 2023), moved to iteration-level scheduling. Specifically, Orca's continuous batching inserts prefill of new requests immediately after each decode iteration, reducing TTFT while preserving throughput via a steady pool of active decode work.

The shift towards long prompts, however, exposed a new bottleneck: decode batches often stall while waiting for a single, large prefill to finish, triggering TBT SLO violations. Chunked prefill (Agrawal et al., 2024) addresses this by splitting long prompts into smaller chunks and interleaving their prefills with decode. The *chunk size* (number of tokens per chunk) is the key tuning knob: smaller chunks reduce per-iteration prefill work and relax TBT pressure, whereas larger chunks reduce the number of chunks and lower TTFT. By keeping per-iteration prefill work within the available slack, chunked prefill suppresses TBT spikes and restores stall-free execution.

However, this benefit of chunking comes with a structural cost in MoE models. Partitioning the input along the token dimension forces each chunk to traverse the same transformer layers and repeatedly reload expert weights, amplifying expert-load counts; the problem is exacerbated as chunk size shrinks to meet tight TBT SLOs. In contrast, scheduling on layer boundaries aligns with MoE expert load units and eliminates redundant KV-cache scans and weight loads, while still allowing stall-free execution.

This paper proposes **layered prefill**, which treats the layer dimension rather than the prompt (token) dimension as the primary scheduling resource, specifically targeting MoE serving. The decoder stack is divided into N_{lg} contiguous *layer groups* (contiguous subsets of transformer layers); N_{lg}, the *number of layer groups*, is the key tuning knob of layered prefill, playing the same role as chunk size in chunked prefill. In each iteration, prefill is performed alongside decode for exactly one group, while the remaining groups perform decode only. This design ensures that prefill completes in exactly N_{lg} iterations without stalling decode. During prefill, each layer is traversed exactly once, which fundamentally avoids memory-access amplification from repeatedly loading experts.

Our key contributions are as follows:

- We treat layers as a first-class scheduling unit and enforce a one-group-per-iteration rule that preserves stall-free decoding for MoE models.
- We eliminate chunk-amplified MoE weight reloads through layered prefill, systematically reducing redundant parameter traffic and improving the TTFT–TBT trade-off in long-context, small-batch settings.
- We reduce energy consumption during MoE serving by cutting redundant MoE reloads, lowering off-chip parameter movement, and decreasing GPU DRAM activity.

2 Large Language Model (LLM)

2.1 LLM Architecture

Contemporary LLMs such as GPT (Brown et al., 2020; Achiam et al., 2024), Llama (Grattafiori et al., 2024), and Qwen (Yang et al., 2025) typically adopt a decoder-only Transformer architecture (Vaswani et al., 2017). A model maps a sequence of token IDs into continuous vector representations, processes them through a stack of Transformer decoder blocks, and finally projects the resulting hidden states into vocabulary logits to predict the next tokens.

Each decoder block follows the same structure: a self-attention layer followed by a feed-forward network (FFN), with residual connections and normalization applied around both. Self-attention enables each token (represented as a vector) to integrate contextual information from earlier tokens in the sequence. It projects input hidden states to query, key, and value matrices via learned projection weights (W_Q, W_K, W_V), applies a causal mask to ensure that tokens attend only to past and current positions, and derives attention weights through softmax-normalized dot products between queries and keys. These weights guide a weighted sum over the values, producing a context-aware token representation. A rich line of work accelerates the attention operator—for example, FlashAttention and its successors (Dao et al., 2022; Dao, 2024; Shah et al., 2024)—which reduce attention I/O

and improve kernel efficiency on modern AI accelerators, such as GPUs (Choquette, 2023; Tirumala & Wong, 2024) and TPUs (Jouppi et al., 2023; Norrie et al., 2021).

FFN refines each token's hidden state independently. It consists of two or more fully connected (FC) layers with non-linear activations, such as ReLU or SwiGLU (Shazeer, 2020). Conventionally, the first FC layer expands the hidden dimension by a factor of 2–4 to increase representational capacity, while the last layer projects it back to the original size to preserve dimensional consistency across layers.

2.2 LLM Inference Structure: Prefill and Decode

Standard LLM inference—unless using diffusion or similar methods—is autoregressive; it generates output tokens one by one while depending on previously generated tokens. The process consists of multiple passes through a full stack of decoder blocks, referred to as iterations. These iterations are grouped into two distinct stages: *prefill* and *decode*.

The prefill stage corresponds to the first iteration. It processes the entire sequence of prompt tokens in parallel along the sequence dimension, passing all tokens through every decoder block. This stage achieves the high compute utilization of AI accelerators due to its parallelism (Du et al., 2025); however, it also incurs high latency as all prompt tokens must be processed before producing the first output token (TTFT).

The decode stage consists of all subsequent iterations, each generating a new token per request. As the sequence length per iteration is only one token, compute units are underutilized (Park et al., 2024), and its efficiency depends heavily on memory access patterns. To improve utilization, systems typically batch decode iterations from multiple requests.

2.3 Prior LLM Serving Approaches and Challenges

LLM serving systems have evolved rapidly to sustain growing request volumes under strict latency service level objectives (SLOs). A series of recent works (Yu et al., 2022; Kwon et al., 2023; Agrawal et al., 2024; Zhong et al., 2024; Patel et al., 2024; Mitra et al., 2025; Kamath et al., 2025; Wang et al., 2025a) proposed increasingly sophisticated scheduling policies aimed at improving utilization while preserving responsiveness.

Early systems such as FasterTransformer (NVIDIA, 2019) relied on static or per-request batching: each batch was fixed and processed independently, without dynamically merging arriving requests. While straightforward to implement, this design yields poor hardware utilization in the decode stage, where only a small number of vectors are processed concurrently, and decode batches can stall behind long prefills, causing high time-between-token (TBT) latency for concurrent requests.

To mitigate these inefficiencies, systems such as Orca (Yu et al., 2022) introduced continuous batching, which merges prefill and decode requests and schedules them together at the iteration level, substantially improving inference throughput. However, this approach still suffers from stalls when long prefills delay subsequent decode steps.

Sarathi-Serve (Agrawal et al., 2024) addresses this issue through chunked prefill. Instead of processing an entire long input in one pass, the sequence is split into smaller chunks that are interleaved with decode requests to form *hybrid* batches. This restructuring enables stall-free[1] execution, since decode requests are no longer blocked behind lengthy prefills. Further, by setting the chunk size—typically 256 or 512 tokens—the scheduler both satisfies the TBT constraint and distributes prefill work more evenly across iterations, which improves hardware utilization.

As depicted in Figure 1 (left), chunked prefill processes one chunk per iteration, processing all decoder layers of the current chunk before moving to the next one. Each chunk reuses the KV cache from preceding chunks, continuing until the full prompt is processed. To improve efficiency, short requests may be coalesced into a single chunk. The chunk size is capped to meet tail-latency SLOs, limiting per-iteration batch size. Consequently, chunked prefill significantly improves service-level objectives (e.g., TTFT and TBT), and we adopt it as the baseline in our evaluation.

2.4 Recent Trends in LLM Architecture and Usage

Mixture of Experts (MoE): MoE architectures (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022) have emerged as a promising approach to scaling model size without proportionally increasing computational complexity. Unlike dense FFNs, which activate all parameters even with small decode batches, MoE routes tokens to a small subset of independent FFNs, referred to as experts, via routers. This sparsity efficiently scales parameter count by activating only a fraction of the model's parameters per token; however, it also introduces challenges in scheduling and memory access (Zoph et al., 2022; Rajbhandari et al., 2022).

Long Context: Recent LLM workloads involve processing much longer input prompts than in the past. Prior systems typically handled prompts in hundreds of tokens (Brown et al., 2020), whereas modern applications driven by multi-turn conversations, retrieval-augmented generation, and chain-of-thought reasoning require contexts of tens of thousands of tokens. This trend is accelerated by the deployment of long-context LLMs such as Claude 3 (Anthropic, 2024), Gemini 2.5 (Comanici et al., 2025), GPT-OSS (OpenAI et al., 2025), and DeepSeek-R1 (DeepSeek-AI et al., 2025),

[1]We define stall-free scheduling as the state where, for each request, every TBT does not exceed a model-specific threshold.

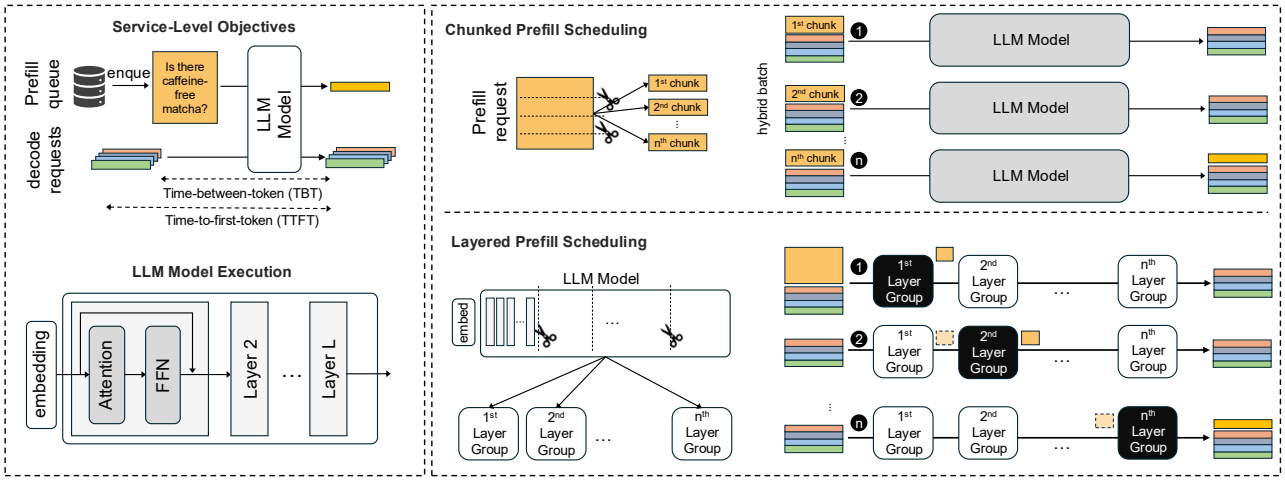

Figure 1. (Upper right) Per iteration, chunked prefill splits an input prompt into multiple chunks, and at each iteration one chunk is processed in order from the beginning with the decode. (Lower right) For layered prefill, exactly one layer group performs both prefill and decode, while the others perform decode only. Prefill advances by one group per iteration, maintaining stall-free decoding.

exceeding a 100k-token context window. Consequently, the prefill stage has become more computationally expensive and memory-intensive, exacerbating the bottlenecks in contemporary, co-located serving systems where both prefill and decode are processed by the same AI accelerators.

These advances in LLM architecture and workloads both increase the computational and memory demands of LLM serving, underscoring the need for new scheduling and parallelism techniques to maintain SLO compliance.

2.5 Energy Consumption in LLM Serving

Power and energy efficiencies are first-order concerns for large-scale LLM serving. Beyond meeting latency-oriented SLOs, service providers must minimize the energy required to serve requests for saving both operating cost and carbon footprint (Stojkovic et al., 2025).

Energy Breakdown and Accounting: The energy cost of LLM serving on AI accelerators can primarily be viewed as the sum of four components: (i) a static baseline that keeps the device active even when idle, (ii) compute energy dominated by matrix–multiply/accumulate engines (e.g., Tensor Cores in GPUs (Navarro et al., 2020; Hanindhito & John, 2024) and MXUs in TPUs (Armoni, 2023; Wang et al., 2020)), (iii) memory energy from moving parameters and KV cache through the memory hierarchy (on-chip SRAM and off-chip DRAM, such as HBM or GDDR), and (iv) communication energy from interconnects that link devices or hosts (e.g., NVLink, PCIe, InfiniBand, and Ethernet).

In practice, data movement accounts for a large share of energy, since each generated token repeatedly rereads substantial portions of the model parameters. This observation can be verified by comparing the power consumption of a 4-bit quantized model (W4A16) with that of a higher-precision counterpart (Shi & Ding, 2025). A practical accounting, thus, is to track the total bytes moved at each level of the memory hierarchy and multiply by the empirically measured energy-per-byte for that level, with the off-chip DRAM term typically setting the overall scale.

MoE Implications: Whether the GEMM is compute-bound or memory-bound can be checked by comparing the batch-implied arithmetic intensity against the accelerator's ridge point. The ridge point is defined as peak arithmetic throughput divided by peak memory bandwidth; it marks the Op-per-Byte (Op/B) at which execution shifts from memory-bound to compute-bound. Operating near the ridge point means that the system utilizes both memory and compute efficiently. Modern AI accelerators typically have ridge points around 100 to 300 Op/B (Yun et al., 2025). This implies a batch size of roughly 200 to 600 for a 2-byte datatype (e.g., bflot16).

With this in mind, although MoE activates only a subset of experts per token, whether this sparsity actually reduces latency and energy in a serving system requires careful analysis. Consider an MoE model with 128 experts and top-k of 8 such as Qwen3-30B-A3B. For an input prompt of 2048 tokens, each expert on average processes about 128 tokens. This falls below the ridge point of contemporary accelerators, so each expert's FC compute remains memory-bound and end-to-end latency and energy are dominated by loading MoE weights from HBM. Making MoE computation compute-bound requires more than 8192 tokens, which would cause a sharp increase in latency (TBT).

Table 1. Expert weight coverage ratio as a function of decode batch size, measured on `Qwen` with the ShareGPT dataset.

Batch Size	1	2	4	8	16	32	64	128	256	≥512
Coverage (%)	6.25	11.7	21.3	29.0	44.5	54.7	69.4	86.3	93.4	≥98

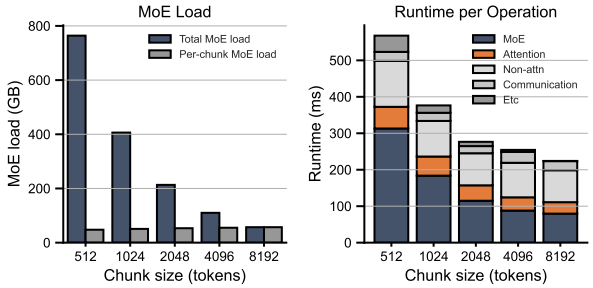

Figure 2. (Left) MoE weight loading vs. chunk size. Per-chunk MoE load remains roughly constant regardless of chunk size, and the total MoE load equals the per-chunk MoE load multiplied by the number of chunks. Therefore, as chunk size increases, the total MoE load tends to decrease. (Right) Runtime of each kernel vs. chunk size, with operations broken down by category (MoE, attention, linear projections, communication, and etc.). The input length is fixed at 8,192 tokens.

3 Limitations of Chunked Prefill on LLMs with MoE

3.1 MoE Weight Load Model

Chunked prefill introduces a conflict for MoE layers. To satisfy tail-latency SLOs, a chunked prefill scheduler limits per-iteration batch sizes (§5.2) whereas efficient MoE weight-loading requires much larger batches to maximize reusing loaded experts. This mismatch makes expert weights broadly activated but poorly reused, leaving execution bottlenecked by memory traffic rather than computation.

Suppose the batch size is 128, top-k is 8, and the total number of experts is 128; on average, only 8 tokens are routed to each expert per iteration. Eight is far below the ridge point of modern AI accelerators. Thus, we formulate this issue as a *sparsity erosion* problem; chunking inflates expert coverage while suppressing reuse, eroding the very sparsity benefits that make MoE efficient. In the following, we quantify expert activation under realistic serving conditions and analyze how chunking disrupts MoE's sparsity advantages.

Measurements on Qwen3-30B-A3B (Yang et al., 2025) (`Qwen`) evaluated on the ShareGPT dataset (ShareGPT, 2023) showcase this issue (see Table 1). Details of the model and dataset are provided in Table 3 and Table 4. At small batch sizes (below 16), fewer than half of the experts are loaded, and even with a moderate batch size of 64, the average expert coverage remains under 70%. As shown in Figure 3, in stable serving regimes (e.g., where SLO attainment exceeds 90%), the average batch size is below 32 on the arXiv dataset (Cohan et al., 2018) and below 128 on the ShareGPT dataset, implying that only a limited fraction of experts is activated per decode batch. In pure decoding, this limited activation is beneficial: it reduces memory traffic relative to dense models.

In contrast, chunked prefill creates hybrid batches that are typically much larger (the batch size often exceeding 256), which activates most experts and erodes the sparsity benefit. At the same time, TBT SLO constraints cap the chunk size; thus, the number of tokens routed to each expert remains insufficient to saturate compute resources, leaving MoE layers memory-bound. Therefore, memory traffic dominates system behavior, amplifying per-iteration latency and energy cost in proportion to prompt length and chunk count.

3.2 Microbenchmark: Runtime vs. Chunk Size

To understand how chunk size shapes MoE efficiency, we isolate its effect on kernel performance. Figure 2 reports MoE weight loading and kernel runtimes as a function of prefill chunk size on `Qwen` (evaluation setup described in §5.1), with the input length fixed at 8,192 tokens to approximate the average prompt length in the arXiv dataset.

When the chunk size is 512, MoE runtime accounts for over 50% of the prefill runtime, and the prefill runtime exceeds 500 ms. As chunk size increases, weight-loading overhead falls sharply, in inverse proportion to chunk size, and runtime per operation decreases correspondingly. Larger chunks route more tokens to each expert, improving reuse and reducing redundant transfers. By 4096–8192 tokens, MoE load drops below 100 GB and prefill runtime stabilizes near 200 ms, with MoE no longer dominating execution. Other kernels modestly depend on chunk size, largely due to lower arithmetic intensity at small problem sizes, but the effect is far less pronounced than in MoE kernels.

These results demonstrate that chunk size strongly governs MoE efficiency: small chunks trigger sparsity erosion and memory-bound slowdowns, whereas larger chunks mitigate the problem by enabling efficient expert reuse.

3.3 Chunk Size Trade-offs in LLM Serving

In a serving scenario, chunk size shapes three key aspects of system behavior: expert reuse, serving capacity, and iteration latency. Larger chunks improve MoE efficiency by increasing reuse and reducing redundant weight load, and they also sustain higher request rates. At the same time, however, larger chunks extend iteration duration, which inflates tail latency and risks violating SLOs.

Table 2 summarizes this trade-off for serving `Qwen` on the arXiv workload. Request rates for each chunk size are ad-

Table 2. Chunk size trade-offs for `Qwen` on the arXiv workload. Larger chunks improve runtime and energy efficiency but inflate tail TBT.

Chunk Size	Req. Rate	TTFT (s) Mean	TTFT (s) p99	TBT (ms) Mean	TBT (ms) p99	Load (GB/req)	Energy (mJ/tok)
512	1.3	2.68	8.05	29.0	48.4	955	60.2
1024	1.7	2.32	5.83	43.6	83.4	631	45.4
2048	2.6	2.56	5.58	73.6	129	304	32.4

justed to keep TTFT approximately 2.5 s. As chunk size increases, prefill runtime decreases. Thus, holding TTFT constant across chunk sizes implies that larger chunks can sustain higher request rates so that the sum of queuing delay and prefill runtime remains 2.5 s. From an efficiency perspective, larger chunks reduce energy consumption per token—defined as total energy divided by the sum of prompt and generated tokens (Wilhelm et al., 2025; Wang et al., 2025b)—from 60 mJ/tok at 512 to 32 mJ/tok at 2048 (-46%). This improvement arises as fewer chunks per request lower the expert weight reloads and increase reuse. Throughput also improves with larger chunks; the system sustains 1.3 req/s at a chunk size of 512 but increases to 2.6 req/s at 2048, consistent with the runtime trends in Figure 2.

However, this conflicts with Sarathi-Serve's original configuration, as large chunk sizes lead to violations of the TBT constraint. As chunk size increases, p99 TBT rises sharply from 48 ms at 512 tokens to 129 ms at 2048 tokens, exceeding the SLO. It is a direct consequence of longer iteration times, which delay the scheduling of concurrent decode requests and inflate the tail of the distribution. Thus, the chunk size that appears optimal from the standpoint of efficiency and throughput is infeasible under strict latency requirements.

4 Layered Prefill Design

4.1 Overview

To address the aforementioned limitations, we propose **layered prefill**. The key idea is to shift the scheduling axis from tokens to layers. Instead of partitioning the input sequence into chunks along the token dimension, the model is vertically partitioned into contiguous *layer groups* (see Figure 1). This group-based layer partitioning naturally aligns with decoder-only Transformers, where layers are homogeneous and executed sequentially.

During execution, layered prefill follows a *one-group-per-iteration* scheduling rule: in each iteration, only one designated layer group performs both decode and prefill for newly admitted requests, while all other groups perform decode only. Therefore, prefill is distributed across iterations, while decoding proceeds continuously without stalls.

This design guarantees that each input prompt traverses the prefill path of each layer exactly once, as opposed to chunked prefill, which forces every layer to process the prompt once per chunk. Consequently, redundant expert-weight reloads across chunks are eliminated, which reduces memory-bandwidth consumption and energy while preserving stall-free decoding under strict TBT SLOs.

4.2 Scheduling Mechanism

At every iteration, up to a single designated layer group jointly processes the ongoing decode batch and the prefill of incoming requests; the remaining groups execute decode-only computation. Thus, prefill progresses group by group across iterations, while decoding remains uninterrupted.

Figure 1 illustrates the process with four layer groups. In the first iteration, prefill for a new request runs within Group 1, whereas Groups 2–4 perform decode only. In the next iteration, prefill advances to Group 2, co-scheduled with decode, and so on. This process repeats until the request has traversed all groups. After N_{lg} (the number of groups) iterations, the prefill of that request completes, while decoding remains stall-free throughout.

Unlike chunked prefill—which induces redundant MoE weight reloads across chunks—the layered prefill schedule ensures that each layer processes the input prompt only once. Thus, decoding stays stall-free while avoiding the amplified memory overheads of token-level partitioning.

Decode is never blocked because every iteration includes decode work across all layer groups, so TBT remains low enough to meet the SLO constraint. *In short, moving from chunked prefill to layered prefill preserves stall-free decoding while substantially reducing redundant MoE weight movement and energy consumption.*

4.3 Generalization with Chunked Prefill

Layered prefill divides the model vertically into N_{lg} contiguous layer groups, whereas chunked prefill partitions the input prompt into multiple chunks along the token axis. Thus, the two methods are orthogonal, meaning that chunked prefill and layered prefill can be applied simultaneously.

Applying both strategies together increases scheduling flexibility and provides two key benefits. First, it enables the efficient processing of very long input sequences. Chunked Pipeline Parallelism, proposed in (Mitra et al., 2025), addresses the challenge of handling extremely long input lengths by partitioning the input into relatively small chunks and pipelining the prefill of each chunk. This method has been shown to be an optimal approach to meeting TTFT constraints for long inputs without requiring complicated parallelism strategies. Since layered prefill is also compatible with chunking, it inherits the advantages of chunked

Table 3. Specifications of the evaluated MoE models, reorganized by metric. GQA stands for grouped-query attention.

Metric	`Qwen`	`GPT`
Parameters	30B	20B
Number of total experts	128	32
Top-k (active experts)	8	4
Experts-to-top-k ratio	16:1	8:1
KV cache size per token	48 KB	<34 KB[2]
Attention	GQA	GQA
Hidden dimension	2048	2880

pipeline parallelism in processing long inputs.

Second, it allows the chunk size to be substantially increased. Increasing the chunk size directly reduces the number of chunks for a given input prompt. Because the MoE weight read overhead grows with the number of chunks, fewer chunks dramatically reduce memory bandwidth consumption and energy cost. Moreover, when the chunk size becomes sufficiently large, the batch size per expert in MoE layers also becomes large enough to saturate compute, shifting the operation from a memory-bound to a compute-bound regime. For example, in a model with 128 experts and top-$k = 8$, a chunk size of 8192 already yields an effective batch size of 512 tokens per expert, which is sufficient to make the MoE layer compute-bound. Once the chunk size becomes large enough to shift the MoE layer into the compute-bound regime, an important advantage emerges: partitioning the input prompt into multiple chunks for MoE computation no longer leads to a surge in latency.

Through this generalization, the system can selectively combine chunked prefill and layered prefill, leveraging the strengths of both approaches while mitigating their individual drawbacks.

4.4 Implementation Details

We adapt the number of layer groups to the input length. Concretely, we choose the number of groups N_{lg} so that the *prefill work per iteration* approximately matches a 512-token chunk in the chunked prefill baseline:

$$N_{lg}(L) \;=\; \max\big(1,\; \lceil L/512 \rceil\big).$$

Hence, for a long prompt of length 8,192 we set $N_{lg} = 16$, whereas for a short prompt of length 512 we set $N_{lg} = 1$. This choice aligns the processing granularity with chunked prefill at a chunk size of 512, making iteration time and admission cadence comparable across schedulers and isolating the effect of switching the scheduling axis (layers vs. tokens). Here, 512 is an arbitrary value and could just as well be 256 or 1024. We use 512 in our experiments to ensure a fair comparison with chunked prefill configured with a chunk size of 512. When combining chunked prefill and layered prefill for very long contexts, we typically use a large chunk size (e.g., 8192) and apply layered prefill with a specific group size (e.g., $N_{lg} = 16$) within each chunk. When multiple small inputs arrive concurrently, we merge them into a single batch to improve efficiency.

We implement layered prefill on top of vLLM (Kwon et al., 2023). Attention leverages FlashAttention-3 (Shah et al., 2024); most computations use custom CUDA kernels, and the remaining layers use `torch.compile`. We adopt CUDA Graphs to accelerate the prefill and decode stages.

[2] `GPT` uses sliding window attention (Beltagy et al., 2020) so that cache size per token varies over the time.

Table 4. Input and output length statistics for evaluation datasets (ShareGPT and arXiv Summarization).

Dataset	Input Length			Output Length		
	Mean	p90	Std	Mean	p90	Std
ShareGPT	2340	5696	2088	438	834	265
arXiv	9194	17152	5754	231	386	104

5 Evaluation

5.1 Experimental Setup

Environment: We conducted all experiments on a server equipped with two NVIDIA H100 GPUs (80 GB each) connected via NVLink. We implemented model parallelism using tensor parallelism.

Models: We evaluated two representative MoE models with similar architectures but different routing configurations. We abbreviate Qwen3-30B-A3B as `Qwen` and GPT-OSS-20B as `GPT` for brevity. `Qwen` comprises 128 experts, and each token activates top-8 experts. `GPT` comprises 32 experts, and each token activates top-4 experts. Detailed specifications are summarized in Table 3. Both employ bfloat16 precision for weights and activations.

Datasets: To analyze the effectiveness of layered prefill, we evaluate on two datasets with distinct characteristics, summarized in Table 4. *ShareGPT* contains multi-turn conversations between users and ChatGPT. ShareGPT exhibits a wide spread in input lengths—from a few hundred tokens to more than 10,000-resulting in a large input—length standard deviation. On average, its input prompts are roughly six times longer than their outputs. *arXiv Summarization* (hereafter, *arXiv*) is a summarization benchmark in which inputs are substantially longer than outputs; on average, the input length is approximately forty times the output length.

Traffic model: We define the request rate as the exogenous arrivals per second (req/s). In our experiments, we generate arrivals with a Poisson process.

Table 5. SLO constraints for the ShareGPT and arXiv datasets.

Model	Dataset	TTFT (s)	TBT (ms)
`Qwen`	ShareGPT	5	125
`Qwen`	arXiv	10	125
`GPT`	ShareGPT	5	100
`GPT`	arXiv	10	100

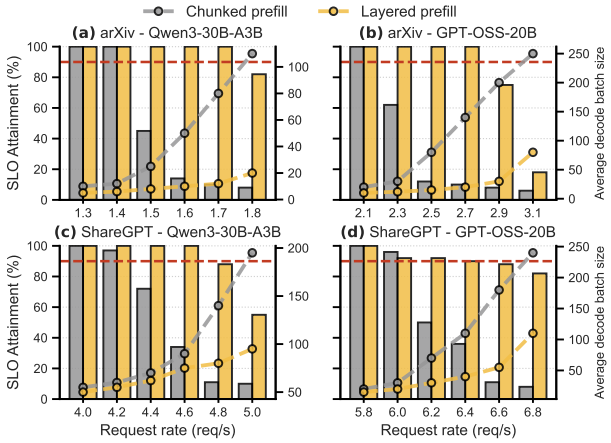

Figure 3. SLO attainment under different request rates. The red horizontal line marks the effective SLO attainment threshold (90%).

SLOs: Following the convention of prior work (Agrawal et al., 2024; Kamath et al., 2025; Feng et al., 2025), the TBT SLO is set to $\sim 5\times$ the time to process 32 decode batches at 4096 tokens. The TTFT SLO is chosen separately for each model-dataset pair to reflect an appropriate operating point. All SLO settings are summarized in Table 5. SLO attainment is evaluated per request: a request attains the SLO if its TTFT meets the TTFT SLO and, thereafter, the TBT of all generated tokens meets the TBT SLO.

Energy: We measured the total energy consumed by the GPU devices during LLM serving using NVML's API (NVIDIA, 2025b; You et al., 2023). Because NVML's API returns the cumulative energy used from the time the driver was loaded to the time of the call, we obtain a task's energy by invoking it at the start and end of the task and taking the difference. Energy per token was measured as the total energy divided by the total number of prompt and generated tokens.

5.2 End-to-end SLO attainment

Layered prefill delivers higher SLO attainment across models and workloads. Figure 3 compares chunked prefill and layered prefill across request rates for two models (`Qwen` and `GPT`) and two workloads (arXiv and ShareGPT).

Table 6. `Qwen` on arXiv (at 1.3 req/s)

Schedule	TTFT (s)		TBT (ms)	
	Mean	p99	Mean	p99
Chunked	2.80	8.65	32.9	51.1
Layered	1.24	4.10	21.5	37.1

`Qwen`: (a) On arXiv, layered prefill sustains $\approx 100\%$ SLO attainment through 1.7 req/s, while chunked prefill collapses at 1.5 req/s. (c) On ShareGPT, layered prefill retains a clear advantage at 4.8 req/s whereas chunked prefill drops at 4.4 req/s.

`GPT`: (b) On arXiv, layered prefill maintains 100% SLO attainment across 2.1–2.7 req/s, whereas chunked prefill degrades quickly beyond 2.3 req/s. (d) On ShareGPT, layered prefill also remains above 90% across 5.8–6.4 req/s, while chunked prefill declines steeply at 6.2 req/s.

The dotted lines denote the average decode batch size. As shown, while maintaining SLO attainment above 90%, the average decode batch size remains at most 32 for arXiv and 128 for ShareGPT. According to Table 1, a decode batch size of 32 activates only about 55% of experts, whereas a batch size of 128 activates roughly 86%. Taken together, these observations indicate that layered prefill improves performance over chunked prefill by approximately 14–45%, depending on the workload. Moreover, comparing within the same model, the performance gap is larger on arXiv—where the prefill-to-decode length ratio is higher—than on ShareGPT. This suggests that as the number of chunks grows relative to the prompt length, chunking-induced degradation becomes more severe, and layered prefill mitigates this effect. These trends are consistent with the detailed TTFT/TBT statistics in Table 6, which show that layered prefill achieves both lower mean latency and tighter tail behavior compared to chunked prefill under the same workload conditions.

Finally, in chunked prefill, increasing the chunk size from 512 to 1024 reduced the number of chunks and lowered TTFT; however, the longer prefill time increased TBT and led to more frequent TBT violations, resulting in lower overall SLO attainment.

5.3 Breakdown of SLO Attainment

Layered prefill maintains higher overall SLO attainment mainly by preserving TTFT under load, while both schedulers already keep TBT near perfect. Figure 4 decomposes end-to-end SLO attainment into its two components—TTFT and TBT—over increasing request rates, comparing chunked prefill to layered prefill. Across models and datasets, layered prefill sustains near-perfect TTFT attainment over a wider operating region than chunked prefill. As the system

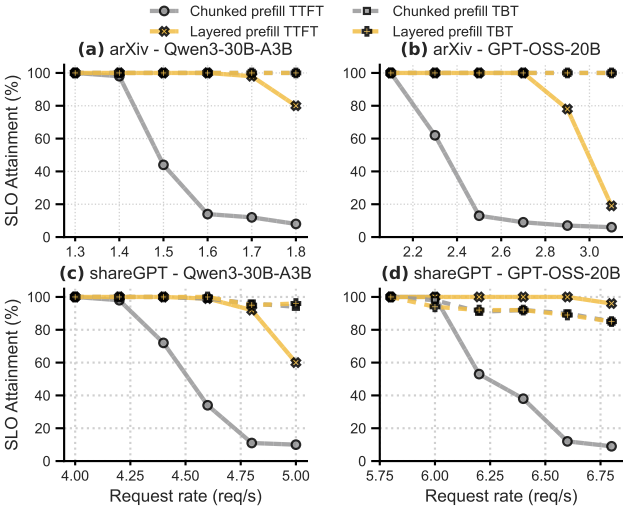

Figure 4. Breakdown of SLO attainment by component across request rates.

approaches saturation, layered prefill exhibits a more gradual decline in TTFT attainment, implying lower queuing delay and shorter prefill runtime relative to chunked prefill.

The TBT curves for both chunked prefill and layered prefill remain near 100% across request rates, indicating that both schedulers effectively provide stall-free scheduling. Even as the request rate increases toward saturation, the TBT attainment exhibits little to no degradation, suggesting that both approaches can maintain stable TBT under severe load conditions. Exceptionally, when evaluating `GPT` on ShareGPT, high request rates inflate decode-batch size and trigger TBT violations in both chunked prefill and layered prefill. However, TTFT violations begin for chunked prefill at 6.2 req/s, whereas layered prefill remains robust with no TTFT violation up to 6.6 req/s.

5.4 MoE Expert Load Traffic

Layered prefill lowers expert-weight load traffic compared to chunked prefill, and the reduction is most significant on long-prompt workloads. We quantified the memory pressure induced by expert activation using a simple counter: the total number of *expert weight load bytes* observed while processing a fixed trace of 100 requests. A load byte is accumulated whenever an MoE expert's parameters are brought into device memory for execution (either during prefill or decode). Table 7 reports this metric for the `Qwen` model on ShareGPT and arXiv under two schedulers: chunked prefill with 512-token chunks and layered prefill.

There are two key observations. First, for ShareGPT, layered prefill reduces expert loads by **12%**, consistent with fewer redundant reloads when prefill is distributed across layer groups and chunk-induced repetition is avoided. Second,

Table 7. Total expert weight loads for 100 requests on `Qwen`.

Dataset	Scheduler	Total Loads	Reduction
ShareGPT	Chunked	28.5 TB	-
	Layered	25.1 TB	**-12.0%**
arXiv	Chunked	35.6 TB	-
	Layered	21.7 TB	**-39.0%**

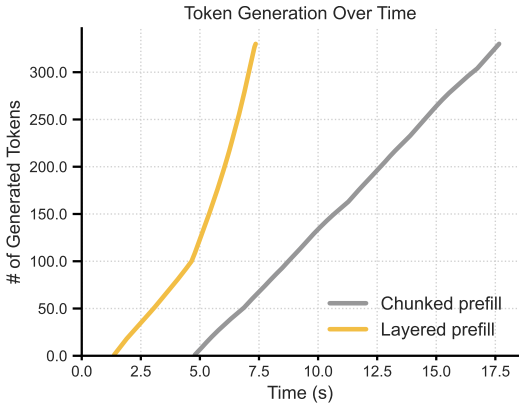

Figure 5. Token generation over time on arXiv with `Qwen`.

the effect is much larger on arXiv (**39%** reduction), where long prompts would otherwise force many chunks, whereas layered prefill with large chunks preserves the sparsity benefit of MoE. These reductions align with the SLO results in Figures 3 and 4: lower expert-load traffic correlates with higher SLO attainment at elevated request rates, particularly on workloads with high prefill-to-decode ratios.

5.5 Token Generation Over Time

Layered prefill emits tokens earlier and maintains a higher generation rate than chunked prefill, yielding more tokens within the same wall-clock window. In Figure 5, we compared cumulative token output for a single request on `Qwen` using the arXiv workload at a request rate of 1.3 req/s under chunked prefill and layered prefill. The steeply rising middle interval reflects the period when layered prefill has quickly finished other requests' prefills and runs in decode-only mode, so token generation accelerates. These factors reduced the average end-to-end latency from **9.4** to **5.5** s (**-41%**). Overall, layered prefill reduces effective TTFT and increases throughput, which lowers end-to-end latency while preserving stall-free decoding.

5.6 Energy per Output Token

Table 8 reports *energy per output token* (mJ/tok) together with latency metrics (mean/p99 TTFT and TBT) at steady-state operating points on arXiv. We selected for each scheduler the highest request rate that still satisfies our latency

Table 8. Normalized energy on arXiv. We report energy per output token (mJ/tok; lower is better) together with mean/p99 TTFT and TBT at steady-state, SLO-compliant operating points. For each scheduler, the listed request rate is the highest that satisfies the latency SLO, so req/s reflects usable capacity.

Model	Method	Req. Rate (req/s)	TTFT (s)		TBT (ms)		mJ / tok
			Mean	p99	Mean	p99	
Qwen	Chunked	1.3	2.80	8.65	32.2	51.3	56.6
	Layered	1.3	1.24	4.10	21.4	37.5	51.7 (-9%)
	Layered	1.6	2.46	8.01	28.1	41.2	44.2 (-22%)
GPT	Chunked	2.1	3.00	8.57	25.1	30.6	37.4
	Layered	2.1	0.87	2.82	18.2	32.3	34.3 (-8%)
	Layered	2.7	2.18	6.86	25.7	33.2	29.8 (-20%)

Table 9. Broader validation across GPUs, models, and workloads.

GPU	Model	Method	TTFT (s)		TBT (ms)	
			Mean	p99	Mean	p99
A100x2	Qwen3-30B-A3B (arXiv, 0.5 req/s)	Chunked (512)	1.32	5.40	22.4	70.4
		Layered ($N_{lg} = 16$)	0.962	3.81	19.3	53.6
H100x2	gpt-oss-120b (fp4) (arXiv, 1.2 req/s)	Chunked (512)	3.11	8.82	34.9	47.2
		Layered ($N_{lg} = 12$)	1.09	3.89	20.5	44.8
H100x8	Qwen3-235B-A22B (arXiv, 1.2 req/s)	Chunked (512)	4.25	11.6	41.7	50.5
		Layered ($N_{lg} = 16$)	2.26	7.42	33.1	45.0
H100x8	Qwen3-235B-A23B (ShareGPT, 3.5 req/s)	Chunked (512)	6.65	16.4	50.2	70.2
		Layered ($N_{lg} = 16$)	3.35	8.48	50.8	88.5
H100x8	gpt-oss-120b (fp4) (arXiv, 3.0 req/s)	Chunked (512)	1.05	3.45	14.4	18.6
		Layered ($N_{lg} = 12$)	0.530	1.75	11.8	19.3
H100x8	gpt-oss-120b (fp4) (ShareGPT, 6.5 req/s)	Chunked (512)	0.212	0.791	12.2	19.6
		Layered ($N_{lg} = 12$)	0.178	0.572	11.3	21.0

SLO targets (TTFT and TBT), so differences in req/s reflect usable capacity under the same QoS envelope.

Qwen: Relative to chunked prefill at 1.3 req/s, layered prefill sustains 1.6 req/s ($+23\%$) while reducing energy per token from **56.6** to **44.2** mJ/tok ($-\mathbf{22}\%$). Latency QoS is maintained or improved (TTFT decreasing; TBT comparable within the SLO). Even at the same request rate, we observed a 9% reduction in energy per token. In this setting, mean TTFT drops by more than **50%** compared to chunked prefill at the same rate.

GPT: Relative to chunked prefill at 2.1 req/s, layered prefill sustains 2.7 req/s ($+29\%$) while reducing energy per token from **37.4** to **29.8** mJ/tok ($-\mathbf{20}\%$). Latency remains within the SLO envelope, with improved TTFT and comparable TBT. When the request rate is the same, energy per token decreases by 8% as well. Under the same request rate, mean TTFT decreases by more than **70%** over chunked prefill.

Takeaways: Across both models, layered prefill delivers **20–22%** lower energy per token at higher sustainable request rates, with equal-or-better TTFT and comparable TBT. These normalized gains are consistent with our microbenchmarks: avoiding repeated expert-weight reloads reduces HBM/NVLink traffic in MoE layers, translating into lower per-token energy without sacrificing latency.

5.7 Broader Validation Across GPUs and Models

To demonstrate the robustness of layered prefill, we evaluated it across different GPUs (A100 and H100), model scales (30B to 235B), and precision regimes (including 4-bit fp4 weights). Table 9 summarizes the results. Across all settings, layered prefill reduces TTFT while maintaining or improving TBT compared to chunked prefill. This confirms that the key benefit of layered prefill is robust across hardware generations, model scales, and precision regimes.

5.8 Comparison to Prefill/Decode Disaggregation

We compared layered prefill against prefill/decode disaggregation on Qwen using an H100x2 server. For disaggrega-

Table 10. Comparison with disaggregated serving.

Method	TTFT (s)		TBT (ms)	
	Mean	p99	Mean	p99
Chunked prefill	3.00	9.15	32.1	49.3
Disaggregated	3.94	11.2	16.4	25.8
Layered prefill	1.24	4.59	19.8	35.9

Table 11. Ablation on group size N_{lg}.

Group size (N_{lg})	TTFT (s)		TBT (ms)	
	Mean	p95	Mean	p99
2	0.480	1.62	20.9	138
4	0.566	2.01	20.6	84.5
8	0.768	2.81	20.8	54.6
16	1.27	3.47	19.7	35.7

tion, one GPU was assigned to prefill and the other to decode, whereas chunked and layered prefill used both GPUs via tensor parallelism. As shown in Table 10, layered prefill achieves significantly better TTFT than disaggregation, while disaggregation achieves lower TBT due to dedicated decode resources. Layered prefill improves both TTFT and TBT over chunked prefill, providing a more balanced trade-off without requiring explicit system-level disaggregation and dynamic capacity balancing.

5.9 Relaxed TBT SLO and Throughput Tradeoff

Under a relaxed TBT SLO, chunked prefill can use larger chunks and layered prefill can use fewer layer groups. As the SLO becomes looser, both saturate and converge, since layered prefill mainly reduces the per-chunk overhead of chunked prefill, which becomes negligible at large chunk sizes. On `Qwen` (H100x2, arXiv, 2.5 req/s), chunked prefill with chunk size 2048 and layered prefill with $N_{lg} = 4$ achieve nearly identical TBT and TTFT (Mean TTFT $\approx$ 2.47s, Mean TBT $\approx$ 78.6ms).

We also compared peak offline throughput on H100x2 (`Qwen`, arXiv). With chunk size 512 for chunked prefill and $N_{lg} = 16$ for layered prefill, layered prefill achieves higher throughput (391.59 vs. 315.36 tokens/sec). This is because chunked prefill can still under-utilize MoE experts despite executing all layers each step, whereas layered prefill improves utilization for the active layers and reduces unnecessary expert weight loading. The throughput gap narrows with larger chunk sizes (and smaller N_{lg}), since redundant MoE loading becomes less significant.

5.10 Experiments on Group Size N_{lg}

Layered prefill introduces the layer group size N_{lg} as a tuning knob. Increasing N_{lg} increases TTFT but improves TBT, while decreasing N_{lg} reduces TTFT at the cost of higher TBT. Table 11 shows this trade-off on H100x2 with `Qwen` (arXiv, 1.4 req/s), demonstrating that N_{lg} provides a practical control knob to match different latency SLOs.

5.11 Long-Context Discussion

As context length grows, attention becomes more expensive, which can reduce the relative impact of MoE weight traffic. However, our profiling shows that MoE remains a substantial fraction of runtime even at long context lengths (e.g., 32K). At chunk size 512, MoE accounts for 57.7% of runtime at 8K input and 44.9% at 32K input. We validated this with $\sim$30K-context ShareGPT requests on `Qwen` (H100x2, 0.1 req/s). Layered prefill ($N_{lg} = 16$) continues to reduce TTFT substantially (1.74s vs. 2.91s) while slightly improving TBT (14.9ms vs. 15.6ms) compared to chunked prefill (chunk=512).

Table 12. Smoothness and fairness metrics.

Method	TTFT (s)		TBT (ms)		TBT std (ms)	Jain Index
	Mean	p99	Mean	p99		
Poisson Arrivals						
Chunked	3.00	9.15	32.1	49.3	10.7	0.835
Layered	1.24	4.59	19.8	35.9	8.51	0.882
Bursty Arrivals (CV=1.83)						
Chunked	5.63	16.7	31.2	48.6	11.2	0.809
Layered	2.55	8.07	22.3	36.6	8.89	0.850

5.12 Responsiveness, Smoothness, and Fairness

Perceived responsiveness is not fully captured by average TTFT/TBT alone. We evaluated token smoothness (TBT std) and a fairness metric based on Jain's index over per-request decode throughput. Table 12 summarizes results under an online workload (H100x2, arXiv, 1.4 req/s). Layered prefill substantially improves TTFT and reduces both mean and tail TBT, while also improving smoothness (lower TBT std). Importantly, Jain's fairness index also increases, suggesting that layered prefill does not introduce additional per-request unfairness in streaming throughput. We also tested bursty arrivals (Coefficient of Variation = 1.83) and observed the same trend, with layered prefill mitigating tail amplification.

5.13 Ablation on a Non-MoE Model

We evaluated a dense (non-MoE) model to test whether layered prefill helps beyond MoE. On Qwen3-8B with a single H100 (arXiv, 1.5 req/s), layered prefill is slower than chunked prefill (Mean TTFT 1.45s vs. 0.955s, Mean TBT 20.3ms vs. 16.2ms). This is expected because chunked prefill already sustains high GPU utilization in dense models,

while layered prefill can reduce utilization without MoE-specific reuse benefits. Thus, layered prefill is most effective for MoE models where expert activation is sparse and strict TBT SLOs require small chunk sizes.

6 Related Work

6.1 LLM Serving Systems

The rapid proliferation of large language models has spurred extensive work on efficient serving infrastructures. Early systems such as FasterTransformer (NVIDIA, 2019) and DeepSpeed-Inference (Rajbhandari et al., 2022) introduced kernel-level optimizations (e.g., fused attention and tensor parallelism) to accelerate single-model inference. Recent work has focused on improving multi-user serving efficiency under resource contention. For example, vLLM (Kwon et al., 2023) proposes PagedAttention to manage the KV cache with fine-grained memory virtualization, enabling high-throughput decoding with large batches. Sarathi-Serve (Agrawal et al., 2024) introduces adaptive batching and request scheduling to balance latency and throughput. Our work builds on this line by revisiting the scheduling of prefill workloads, specifically addressing inefficiencies that arise in MoE models under chunked prefill.

6.2 Energy Efficiency of LLMs

Beyond raw performance, energy efficiency has emerged as a critical concern in large-scale deployment. Prior studies (Wolters et al., 2024; Raha et al., 2025) report that memory traffic dominates the energy footprint of modern accelerators, motivating techniques that reduce parameter movement or KV cache access. Architectural proposals such as MLA (DeepSeek-AI et al., 2024) compress KV caches to lower both memory capacity and power demand. Serving-level approaches, including early-exit strategies and speculative decoding, have been analyzed for their potential to cut energy per token. However, most prior work evaluates dense transformer models; relatively little attention has been given to the unique energy behavior of MoE models, where expert weight loading can significantly amplify off-chip memory energy consumption. Our work complements this line by quantifying the energy overhead of MoE under different prefill scheduling policies and by demonstrating that layered prefill can reduce both latency and energy costs in long-context workloads.

6.3 Scheduling and Disaggregation

FlexGen (Sheng et al., 2023) targets throughput-oriented scenarios with very loose SLOs, focusing on layer-wise offloading and batching strategies (e.g., zig-zag schedule) to maximize efficiency over large batches under memory pressure. In contrast, layered prefill is designed for online serving under strict streaming latency requirements, specifically targeting the MoE expert reuse failure mode and per-chunk overhead induced by chunked prefill under tight TBT constraints. While both approaches share the high-level idea of changing scheduling granularity, they operate under fundamentally different constraints and bottlenecks.

Disaggregated serving systems, such as Dynamo (NVIDIA, 2025a) and DistServe (Zhong et al., 2024), change the placement and pipeline structure across engines or nodes by separating prefill and decode. Layered prefill is orthogonal to these systems because it changes the scheduling unit inside a single engine. These techniques can be combined, and layered prefill can improve expert reuse even within a colocated or partially disaggregated setup.

Layered prefill is compatible with speculative decoding (Leviathan et al., 2023), which restructures the decode process into draft/verify cycles. By treating verification as a regular decode batch, layered prefill can be used together with speculative decoding.

7 Conclusion

Conventional chunked prefill scheduling is inefficient for MoE models, as it amplifies redundant expert weight loads, increasing both latency and energy consumption. This paper introduces *layered prefill*, a new scheduling strategy that partitions a model along the layer dimension rather than the token dimension. This approach fundamentally eliminates the redundant memory traffic caused by chunking. Our evaluation on representative MoE models shows that layered prefill consistently improves SLO attainment, reduces expert-load traffic by up to **39%**, and lowers TTFT by up to **70%**, end-to-end latency by **41%**, and per-token energy by up to **22%** on long-context workloads. These results underscore the importance of co-designing scheduling policies with emerging model architectures.

Future work can extend this approach to complex multi-GPU environments and explore adaptive layer grouping strategies, particularly for models where the layer count is not an even multiple of the group count. Further research could also jointly optimize for throughput, latency, and energy efficiency at a data-center scale.

Acknowledgements

This work was supported by Mobile eXperience (MX) Business, Samsung Electronics Co., Ltd. It was also supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by MSIT (RS-2021-II211343, IITP-2026-RS-2023-00256081). Our source code is available at `https://github.com/scale-snu/layered-prefill`

REFERENCES

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. GPT-4 Technical Report, 2024.

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

Anthropic. The Claude 3 Model Family: Opus, Sonnet, Haiku. https://api.semanticscholar.org/CorpusID:268232499, 2024. Accessed: 2025-10-01.

Armoni, M. Tensor Processing Units (TPU): A Technical Analysis and Their Impact on Artificial Intelligence. *Tech4Future Information Technology Report*, 2023.

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The Long-Document Transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020. ISBN 9781713829546.

Choquette, J. NVIDIA Hopper H100 GPU: Scaling Performance. *IEEE Micro*, 43(3):9–17, 2023.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. *arXiv preprint arXiv:1804.05685*, 2018.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., Marris, L., Petulla, S., Gaffney, C., Aharoni, A., Lintz, N., Pais, T. C., Jacobsson, H., Szpektor, I., Jiang, N.-J., Haridasan, K., Omran, A., Saunshi, N., Bahri, D., Mishra, G., Chu, E., Boyd, T., Hekman, B., Parisi, A., Zhang, C., Kawintiranon, K., Bedrax-Weiss,

T., Wang, O., Xu, Y., Purkiss, O., Mendlovic, U., Deutel, I., Nguyen, N., Langley, A., Korn, F., Rossazza, L., Ramé, A., Waghmare, S., Miller, H., Byrd, N., Sheshan, A., Bhardwaj, R. H. S., Janus, P., Rissa, T., Horgan, D., Silver, S., Wahid, A., Brin, S., Raimond, Y., Kloboves, K., Wang, C., Gundavarapu, N. B., Shumailov, I., Wang, B., Pajarskas, M., Heyward, J., Nikoltchev, M., Kula, M., Zhou, H., Garrett, Z., Kafle, S., Arik, S., Goel, A., Yang, M., Park, J., Kojima, K., Mahmoudieh, P., Kavukcuoglu, K., Chen, G., Fritz, D., Bulyenov, A., Roy, S., Paparas, D., Shemtov, H., Chen, B.-J., Strudel, R., Reitter, D., Roy, A., Vlasov, A., Ryu, C., Leichner, C., Yang, H., Mariet, Z., Vnukov, D., Sohn, T., Stuart, A., Liang, W., Chen, M., Rawlani, P., Koh, C., Co-Reyes, J., Lai, G., Banzal, P., Vytiniotis, D., Mei, J., Cai, M., Badawi, M., Fry, C., Hartman, A., Zheng, D., Jia, E., Keeling, J., Louis, A., Chen, Y., Robles, E., Hung, W.-C., Zhou, H., Saxena, N., Goenka, S., Ma, O., Fisher, Z., Taege, M. H., Graves, E., Steiner, D., Li, Y., Nguyen, S., Sukthankar, R., Stanton, J., Eslami, A., Shen, G., Akin, B., Guseynov, A., Zhou, Y., Alayrac, J.-B., Joulin, A., Farkash, E., Thapliyal, A., Roller, S., Shazeer, N., Davchev, T., Koo, T., Forbes-Pollard, H., Audhkhasi, K., Farquhar, G., Gilady, A. M., Song, M., Aslanides, J., Mendolicchio, P., Parrish, A., Blitzer, J., Gupta, P., Ju, X., Yang, X., Datta, P., Tacchetti, A., Mehta, S. V., Dibb, G., Gupta, S., Piccinini, F., Hadsell, R., Rajayogam, S., Jiang, J., Griffin, P., Sundberg, P., Hayes, J., Frolov, A., Xie, T., Zhang, A., Dasgupta, K., Kalra, U., Shani, L., Macherey, K., Huang, T.-K., MacDermed, L., Duddu, K., Zacchello, P., Yang, Z., Lo, J., Hui, K., Kastelic, M., Gasaway, D., Tan, Q., Yue, S., Barrio, P., Wieting, J., Yang, W., Nystrom, A., Demmessie, S., Levskaya, A., Viola, F., Tekur, C., Billock, G., Necula, G., Joshi, M., Schaeffer, R., Lokhande, S., Sorokin, C., Shenoy, P., Chen, M., Collier, M., Li, H., Bos, T., Wichers, N., Lee, S. J., Pouget, A., Thangaraj, S., Axiotis, K., Crone, P., Sterneck, R., Chinaev, N., Krakovna, V., Ferludin, O., Gemp, I., Winkler, S., Goldberg, D., Korotkov, I., Xiao, K., Mehrotra, M., Mariserla, S., Piratla, V., Thurk, T., Pham, K., Ma, H., Senges, A., Kumar, R., Meyer, C., Talius, E., Pierse, N. W., Sandhu, B., Toma, H., Lin, K., Nath, S., Stone, T., Sadigh, D., Gupta, N., Guez, A., Singh, A., Thomas, M., Duerig, T., Gong, Y., Tanburn, R., Zhang, L. L., Dao, P., Hammad, M., Xie, S., Rijhwani, S., Murdoch, B., Kim, D., Thompson, W., Cheng, H.-T., Sohn, D., Sprechmann, P., Xu, Q., Tadepalli, S., Young, P., Zhang, Y., Srinivasan, H., Aperghis, M., Ayyar, A., Fitoussi, H., Burnell, R., Madras, D., Dusenberry, M., Xiong, X., Oguntebi, T., Albrecht, B., Bornschein, J., Mitrović, J., Dimarco, M., Shamanna, B. K., Shah, P., Sezener, E., Upadhyay, S., Lacey, D., Schiff, C., Baur, S., Ganapathy, S., Schnider, E., Wirth, M., Schenck, C., Simanovsky, A., Tan, Y.-X., Fränken, P., Duan, D., Mankalale, B., Dhawan, N., Sequeira, K., Wei, Z., Goel, S., Unlu, C., Zhu, Y., Sun, H., Balashankar, A., Shuster, K., Umekar, M., Alnahlawi, M., van den Oord, A., Chen, K., Zhai, Y., Dai, Z., Lee, K.-H., Doi, E., Zilka, L., Vallu, R., Shrivastava, D., Lee, J., Husain, H., Zhuang, H., Cohen-Addad, V., Barber, J., Atwood, J., Sadovsky, A., Wellens, Q., Hand, S., Rajendran, A., Turker, A., Carey, C., Xu, Y., Soltau, H., Li, Z., Song, X., Li, C., Kemaev, I., Brown, S., Burns, A., Patraucean, V., Stanczyk, P., Aravamudhan, R., Blondel, M., Noga, H., Blanco, L., Song, W., Isard, M., Sharma, M., Hayes, R., Badawy, D. E., Lamp, A., Laish, I., Kozlova, O., Chan, K., Singla, S., Sunkara, S., Upadhyay, M., Liu, C., Bai, A., Wilkiewicz, J., Zlocha, M., Liu, J., Li, Z., Li, H., Barak, O., Raboshchuk, G., Choi, J., Liu, F., Jue, E., Sharma, M., Marzoca, A., Busa-Fekete, R., Korsun, A., Elisseeff, A., Shen, Z., Carthy, S. M., Lamerigts, K., Hosseini, A., Lin, H., Chen, C., Yang, F., Chauhan, K., Omernick, M., Jia, D., Zainullina, K., Hassabis, D., Vainstein, D., Amid, E., Zhou, X., Votel, R., Vértes, E., Li, X., Zhou, Z., Lazaridou, A., McMahan, B., Narayanan, A., Soyer, H., Basu, S., Lee, K., Perozzi, B., Cao, Q., Berrada, L., Arya, R., Chen, K., Katrina, Xu, Lochbrunner, M., Hofer, A., Sharifzadeh, S., Wu, R., Goldman, S., Awasthi, P., Wang, X., Wu, Y., Sha, C., Zhang, B., Mikuła, M., Graziano, F., Mcloughlin, S., Giannoumis, I., Namiki, Y., Malik, C., Radebaugh, C., Hall, J., Leal-Cavazos, R., Chen, J., Sindhwani, V., Kao, D., Greene, D., Griffith, J., Welty, C., Montgomery, C., Yoshino, T., Yuan, L., Goodman, N., Michaely, A. H., Lee, K., Sawhney, K., Chen, W., Zheng, Z., Shum, M., Savinov, N., Pot, E., Pak, A., Zadimoghaddam, M., Bhatnagar, S., Lewenberg, Y., Kutzman, B., Liu, J., Katzen, L., Selier, J., Djolonga, J., Lepikhin, D., Xu, K., Liang, J., Tan, J., Schillings, B., Ersoy, M., Blois, P., Bandemer, B., Singh, A., Lebedev, S., Joshi, P., Brown, A. R., Palmer, E., Pathak, S., Jalan, K., Zubach, F., Lall, S., Parker, R., Gunjan, A., Rogulenko, S., Sanghai, S., Leng, Z., Egyed, Z., Li, S., Ivanova, M., Andriopoulos, K., Xie, J., Rosenfeld, E., Wright, A., Sharma, A., Geng, X., Wang, Y., Kwei, S., Pan, R., Zhang, Y., Wang, G., Liu, X., Yeung, C., Cole, E., Rosenberg, A., Yang, Z., Chen, P., Polovets, G., Nair, P., Saxena, R., Smith, J., yiin Chang, S., Mahendru, A., Grant, S., Iyer, A., Cai, I., McGiffin, J., Shen, J., Walton, A., Girgis, A., Woodman, O., Ke, R., Kwong, M., Rouillard, L., Rao, J., Li, Z., Xu, Y., Prost, F., Zou, C., Ji, Z., Magni, A., Liechty, T., Calian, D. A., Ramachandran, D., Krivokon, I., Huang, H., Chen, T., Hauth, A., Ilić, A., Xi, W., Lim, H., Ion, V.-D., Moradi, P., Toksoz-Exley, M., Bullard, K., Allamanis, M., Yang, X., Wang, S., Hong, Z., Gergely, A., Li, C., Mittal, B., Kovalev, V., Ungureanu, V., Labanowski, J., Wassenberg, J., Lacasse, N., Cideron, G., Dević, P., Marsden, A., Nguyen, L., Fink, M., Zhong, Y., Kiyono, T., Ivanov, D., Ma, S., Bain, M., Yalasangi, K., She, J., Petrushkina, A., Lunayach, M., Bromberg, C.,

Hodkinson, S., Meshram, V., Vlasic, D., Kyker, A., Xu, S., Stanway, J., Yang, Z., Zhao, K., Tung, M., Odoom, S., Fujii, Y., Gilmer, J., Kim, E., Halim, F., Le, Q., Bohnet, B., El-Sayed, S., Neyshabur, B., Reynolds, M., Reich, D., Xu, Y., Moreira, E., Sharma, A., Liu, Z., Hosseini, M. J., Raisinghani, N., Su, Y., Lao, N., Formoso, D., Gelmi, M., Gueta, A., Dey, T., Gribovskaya, E., Ćevid, D., Mudgal, S., Bingham, G., Wang, J., Kumar, A., Cullum, A., Han, F., Bousmalis, K., Cedillo, D., Chu, G., Magay, V., Michel, P., Hlavnova, E., Calandriello, D., Ariafar, S., Yao, K., Sehwag, V., Vezer, A., Lago, A. D., Zhu, Z., Rubenstein, P. K., Porter, A., Baddepudi, A., Riva, O., Istin, M. D., Yeh, C.-K., Li, Z., Howard, A., Jha, N., Chen, J., de Liedekerke, R., Ahmed, Z., Rodriguez, M., Bhatia, T., Wang, B., Elqursh, A., Klinghoffer, D., Chen, P., Kohli, P., I, T., Zhang, W., Nado, Z., Chen, J., Chen, M., Zhang, G., Singh, A., Hillier, A., Lebron, F., Tao, Y., Liu, T., Dulac-Arnold, G., Zhang, J., Narayan, S., Liu, B., Firat, O., Bhowmick, A., Liu, B., Zhang, H., Zhang, Z., Rotival, G., Howard, N., Sinha, A., Grushetsky, A., Beyret, B., Gopalakrishnan, K., Zhao, J., He, K., Payrits, S., Nabulsi, Z., Zhang, Z., Chen, W., Lee, E., Fallen, N., Gollapudi, S., Zhou, A., Pavetić, F., Köppe, T., Huang, S., Pasumarthi, R., Fernando, N., Fischer, F., Ćurko, D., Gao, Y., Svensson, J., Stone, A., Qureshi, H., Sinha, A., Kulshreshtha, A., Matysiak, M., Mao, J., Saroufim, C., Faust, A., Duan, Q., Fidel, G., Katircioglu, K., Kaufman, R. L., Shah, D., Kong, W., Bapna, A., Weisz, G., Dunleavy, E., Dutta, P., Liu, T., Chaabouni, R., Parada, C., Wu, M., Belias, A., Bissacco, A., Fort, S., Xiao, L., Huot, F., Knutsen, C., Blau, Y., Li, G., Prendki, J., Love, J., Chow, Y., Charoenpanit, P., Shimokawa, H., Coriou, V., Gregor, K., Izo, T., Akula, A., Pinto, M., Hahn, C., Paulus, D., Guo, J., Sharma, N., Hsieh, C.-J., Chukwuka, A., Hashimoto, K., Rauschmayr, N., Wu, L., Angermueller, C., Wang, Y., Gerlach, S., Pliskin, M., Mirylenka, D., Ma, M., Baugher, L., Gale, B., Bijwadia, S., Rakićević, N., Wood, D., Park, J., Chang, C.-C., Seal, B., Tar, C., Krasowiak, K., Song, Y., Stephanov, G., Wang, G., Maggioni, M., Lin, S. X., Wu, F., Paul, S., Jiang, Z., Agrawal, S., Piot, B., Feng, A., Kim, C., Doshi, T., Lai, J., Chuqiao, Xu, Vikram, S., Chelba, C., Krause, S., Zhuang, V., Rae, J., Denk, T., Collister, A., Weerts, L., Luo, X., Lu, Y., Garnes, H., Gupta, N., Spitz, T., Hassidim, A., Liang, L., Shafran, I., Humphreys, P., Vassigh, K., Wallis, P., Shejwalkar, V., Perez-Nieves, N., Hornung, R., Tan, M., Westberg, B., Ly, A., Zhang, R., Farris, B., Park, J., Kosik, A., Cankara, Z., Maksai, A., Xu, Y., Cassirer, A., Caelles, S., Abdolmaleki, A., Chiang, M., Fabrikant, A., Shetty, S., He, L., Giménez, M., Hashemi, H., Panthaplackel, S., Kulizhskaya, Y., Deshmukh, S., Pighin, D., Alazard, R., Jindal, D., Noury, S., S, P. K., Qin, S., Dotiwalla, X., Spencer, S., Babaeizadeh, M., Chen, B. J., Mehta, V., Lees, J., Leach, A., Koanantakool, P., Akolzin, I., Comanescu, R., Ahn, J., Svyatkovskiy, A., Mustafa, B., D'Ambrosio, D., Garlapati, S. M. R., Lamblin, P., Agarwal, A., Song, S., Sessa, P. G., Coquinot, P., Maggs, J., Masoom, H., Pitta, D., Wang, Y., Morris-Suzuki, P., Porter, B., Jia, J., Dudek, J., R, R., Paduraru, C., Ansell, A., Bolukbasi, T., Lu, T., Ganeshan, R., Wang, Z., Griffiths, H., Benenson, R., He, Y., Swirhun, J., Papamakarios, G., Chawla, A., Sengupta, K., Wang, Y., Milutinovic, V., Mordatch, I., Jia, Z., Smith, J., Ng, W., Nigam, S., Young, M., Vušak, E., Hechtman, B., Goenka, S., Zipori, A., Ayoub, K., Popat, A., Acharya, T., Yu, L., Bloxwich, D., Song, H., Roit, P., Li, H., Boag, A., Nayakanti, N., Chandra, B., Ding, T., Mehta, A., Hope, C., Zhang, J., Shtacher, I. H., Badola, K., Nakashima, R., Sozanschi, A., Comşa, I., Žužul, A., Caveness, E., Odell, J., Watson, M., de Cesare, D., Lippe, P., Lockhart, D., Verma, S., Chen, H., Sun, S., Zhuo, L., Shah, A., Gupta, P., Muzio, A., Niu, N., Zait, A., Singh, A., Gaba, M., Ye, F., Ramachandran, P., Saleh, M., Popa, R. A., Dubey, A., Liu, F., Javanmardi, S., Epstein, M., Hemsley, R., Green, R., Ranka, N., Cohen, E., Fu, C. K., Ghemawat, S., Borovik, J., Martens, J., Chen, A., Shyam, P., Pinto, A. S., Yang, M.-H., Ţifrea, A., Du, D., Gong, B., Agarwal, A., Kim, S., Frank, C., Shah, S., Song, X., Deng, Z., Mikhalap, A., Chatziprimou, K., Chung, T., Creswell, T., Zhang, S., Jun, Y., Lebsack, C., Truong, W., Andačić, S., Yona, I., Fornoni, M., Rong, R., Toropov, S., Soudagar, A. S., Audibert, A., Zaiem, S., Abbas, Z., Rusu, A., Potluri, S., Weng, S., Kementsietsidis, A., Tsitsulin, A., Peng, D., Ha, N., Jain, S., Latkar, T., Ivanov, S., McLean, C., GP, A., Venkataraman, R., Liu, C., Krishnan, D., D'sa, J., Yogev, R., Collins, P., Lee, B., Ho, L., Doersch, C., Yona, G., Gao, S., Ferreira, F. T., Ozturel, A., Muckenhirn, H., Zheng, C., Balasubramaniam, G., Bansal, M., van den Driessche, G., Eiger, S., Haykal, S., Misra, V., Goyal, A., Martins, D., Leung, G., Valfridsson, J., Flynn, F., Bishop, W., Pang, C., Halpern, Y., Yu, H., Moore, L., Yuvein, Zhu, Thiagarajan, S., Drori, Y., Xiao, Z., Dery, L., Jagerman, R., Lu, J., Ge, E., Aggarwal, V., Khare, A., Tran, V., Elyada, O., Alet, F., Rubin, J., Chou, I., Tian, D., Bai, L., Chan, L., Lew, L., Misiunas, K., Bilal, T., Ray, A., Raghuram, S., Castro-Ros, A., Carpenter, V., Zheng, C., Kilgore, M., Broder, J., Xue, E., Kallakuri, P., Dua, D., Yuen, N., Chien, S., Schultz, J., Agrawal, S., Tsarfaty, R., Hu, J., Kannan, A., Marcus, D., Kothari, N., Sun, B., Horn, B., Bošnjak, M., Naeem, F., Hirsch, D., Chiang, L., Fang, B., Han, J., Wang, Q., Hora, B., He, A., Lučić, M., Changpinyo, B., Tripathi, A., Youssef, J., Kwak, C., Schlattner, P., Graves, C., Leblond, R., Zeng, W., Andreassen, A., Rasskin, G., Song, Y., Cao, E., Oh, J., Hoffman, M., Skut, W., Zhang, Y., Stritar, J., Cai, X., Khanna, S., Wang, K., Sharma, S., Reisswig, C., Jun, Y., Prasad, A., Sholokhova, T., Singh, P., Rosenthal, A. G., Ruoss, A., Beaufays, F., Kirmani, S., Chen, D., Schalkwyk, J., Herzig, J., Kim, B., Jacob, J., Vincent, D.,

Reyes, A. N., Balazevic, I., Hussenot, L., Schneider, J., Barnes, P., Castro, L., Babbula, S. R., Green, S., Cabi, S., Duduta, N., Driess, D., Galt, R., Velan, N., Wang, J., Jiao, H., Mauger, M., Phan, D., Patel, M., Galić, V., Chang, J., Marcus, E., Harvey, M., Salazar, J., Dabir, E., Sheth, S. S., Mandhane, A., Sedghi, H., Willcock, J., Zandieh, A., Prabhakara, S., Amini, A., Miech, A., Stone, V., Nicosia, M., Niemczyk, P., Xiao, Y., Kim, L., Kwasiborski, S., Verma, V., Oflazer, A. M., Hirnschall, C., Sung, P., Liu, L., Everett, R., Bakker, M., Ágoston Weisz, Wang, Y., Sampathkumar, V., Shaham, U., Xu, B., Altun, Y., Wang, M., Saeki, T., Chen, G., Taropa, E., Vasanth, S., Austin, S., Huang, L., Petrovic, G., Dou, Q., Golovin, D., Rozhdestvenskiy, G., Culp, A., Wu, W., Sano, M., Jain, D., Proskurnia, J., Cevey, S., Ruiz, A. C., Patil, P., Mirzazadeh, M., Ni, E., Snaider, J., Fan, L., Fréchette, A., Pierigiovanni, A., Iqbal, S., Lee, K., Fantacci, C., Xing, J., Wang, L., Irpan, A., Raposo, D., Luan, Y., Chen, Z., Ganapathy, H., Hui, K., Nie, J., Guyon, I., Ge, H., Vij, R., Zheng, H., Lee, D., Castaño, A., Baatarsukh, K., Ibagon, G., Chronopoulou, A., FitzGerald, N., Viswanadha, S., Huda, S., Moroshko, R., Stoyanov, G., Kolhar, P., Vaucher, A., Watts, I., Kuncoro, A., Michalewski, H., Kambala, S., Batsaikhan, B.-O., Andreev, A., Jurenka, I., Le, M., Chen, Q., Jishi, W. A., Chakera, S., Chen, Z., Kini, A., Yadav, V., Siddhant, A., Labzovsky, I., Lakshminarayanan, B., Bostock, C. G., Botadra, P., Anand, A., Bishop, C., Conway-Rahman, S., Agarwal, M., Donchev, Y., Singhal, A., de Chaumont Quitry, F., Ponomareva, N., Agrawal, N., Ni, B., Krishna, K., Samsikova, M., Karro, J., Du, Y., von Glehn, T., Lu, C., Choquette-Choo, C. A., Qin, Z., Zhang, T., Li, S., Tyam, D., Mishra, S., Lowe, W., Ji, C., Wang, W., Faruqui, M., Slone, A., Dalibard, V., Narayanaswamy, A., Lambert, J., Manzagol, P.-A., Karliner, D., Bolt, A., Lobov, I., Kusupati, A., Ye, C., Yang, X., Zen, H., George, N., Bhutani, M., Lacombe, O., Riachi, R., Bansal, G., Soh, R., Gao, Y., Yu, Y., Yu, A., Nottage, E., Rojas-Esponda, T., Noraky, J., Gupta, M., Kotikalapudi, R., Chang, J., Deur, S., Graur, D., Mossin, A., Farnese, E., Figueira, R., Moufarek, A., Huang, A., Zochbauer, P., Ingram, B., Chen, T., Wu, Z., Puigdomènech, A., Rechis, L., Yu, D., Padmanabhan, S. G. S., Zhu, R., ling Ko, C., Banino, A., Daruki, S., Selvan, A., Bhaswar, D., Diaz, D. H., Su, C., Scellato, S., Brennan, J., Han, W., Chung, G., Agrawal, P., Khandelwal, U., Sim, K. C., Lustman, M., Ritter, S., Guu, K., Xia, J., Jain, P., Wang, E., Hill, T., Rossini, M., Kostelac, M., Misiunas, T., Sabne, A., Kim, K., Iscen, A., Wang, C., Leal, J., Sreevatsa, A., Evci, U., Warmuth, M., Joshi, S., Suo, D., Lottes, J., Honke, G., Jou, B., Karp, S., Hu, J., Sahni, H., Taïga, A. A., Kong, W., Ghosh, S., Wang, R., Pavagadhi, J., Axelsson, N., Grigorev, N., Siegler, P., Lin, R., Wang, G., Parisotto, E., Maddineni, S., Subudhi, K., Ben-David, E., Pochernina, E., Keller, O., Avrahami, T., Yuan, Z., Mehta, P., Liu, J., Yang, S., Kan, W., Lee, K., Funkhouser, T., Cheng, D., Shi, H., Sharma, A., Kelley, J., Eyal, M., Malkov, Y., Tallec, C., Bahat, Y., Yan, S., Xintian, Wu, Lindner, D., Wu, C., Caciularu, A., Luo, X., Jenatton, R., Zaman, T., Bi, Y., Kornakov, I., Mallya, G., Ikeda, D., Karo, I., Singh, A., Evans, C., Netrapalli, P., Nallatamby, V., Tian, I., Assael, Y., Raunak, V., Carbune, V., Bica, I., Madmoni, L., Cattle, D., Grover, S., Somandepalli, K., Lall, S., Vázquez-Reina, A., Patana, R., Mu, J., Talluri, P., Tran, M., Aggarwal, R., Skerry-Ryan, R., Xu, J., Burrows, M., Pan, X., Yvinec, E., Lu, D., Zhang, Z., Nguyen, D. D., Mu, H., Barcik, G., Ran, H., Beltrone, L., Choromanski, K., Kharrat, D., Albanie, S., Purser-haskell, S., Bieber, D., Zhang, C., Wang, J., Hudson, T., Zhang, Z., Fu, H., Mauerer, J., Bateni, M. H., Maschinot, A., Wang, B., Zhu, M., Pillai, A., Weyand, T., Liu, S., Akerlund, O., Bertsch, F., Premachandran, V., Jin, A., Roulet, V., de Boursac, P., Mittal, S., Ndebele, N., Karadzhov, G., Ghalebikesabi, S., Liang, R., Wu, A., Cong, Y., Ghelani, N., Singh, S., Fatemi, B., Warren, Chen, Kwong, C., Kolganov, A., Li, S., Song, R., Kuang, C., Miryoosefi, S., Webster, D., Wendt, J., Socala, A., Su, G., Mendonça, A., Gupta, A., Li, X., Tsai, T., Qiong, Hu, Kang, K., Chen, A., Girgin, S., Xian, Y., Lee, A., Ramsden, N., Baker, L., Elish, M. C., Krayvanova, V., Joshi, R., Simsa, J., Yang, Y.-Y., Ambroszczyk, P., Ghosh, D., Kar, A., Shangguan, Y., Yamamori, Y., Akulov, Y., Brock, A., Tang, H., Vashishtha, S., Munoz, R., Steiner, A., Andra, K., Eppens, D., Feng, Q., Kobayashi, H., Goldshtein, S., Mahdy, M. E., Wang, X., Jilei, Wang, Killam, R., Kwiatkowski, T., Kopparapu, K., Zhan, S., Jia, C., Bendebury, A., Luo, S., Recasens, A., Knight, T., Chen, J., Patel, M., Li, Y., Withbroe, B., Weesner, D., Bhatia, K., Ren, J., Eisenbud, D., Songhori, E., Sun, Y., Choma, T., Kementsietsidis, T., Manning, L., Roark, B., Farhan, W., Feng, J., Tatineni, S., Cobon-Kerr, J., Li, Y., Hendricks, L. A., Noble, I., Breaux, C., Kushman, N., Peng, L., Xue, F., Tobin, T., Rogers, J., Lipschultz, J., Alberti, C., Vlaskin, A., Dehghani, M., Sharma, R., Warkentin, T., Lee, C.-Y., Uria, B., Juan, D.-C., Chandorkar, A., Sheftel, H., Liu, R., Davoodi, E., Pigem, B. D. B., Dhamdhere, K., Ross, D., Hoech, J., Mahdieh, M., Liu, L., Li, Q., McCafferty, L., Liu, C., Mircea, M., Song, Y., Savant, O., Saade, A., Cherry, C., Hellendoorn, V., Goyal, S., Pucciarelli, P., Torres, D. V., Yahav, Z., Lee, H., Sjoesund, L. L., Kirov, C., Chang, B., Ghoshal, D., Li, L., Baechler, G., Pereira, S., Sainath, T., Boral, A., Grewe, D., Halumi, A., Phu, N. M., Shen, T., Ribeiro, M. T., Varma, D., Kaskasoli, A., Feinberg, V., Potti, N., Kahn, J., Wisniewski, M., Mohamed, S., Hrafnkelsson, A. M., Shahriari, B., Lespiau, J.-B., Patel, L., Yeung, L., Paine, T., Mei, L., Ramirez, A., Shivanna, R., Zhong, L., Woodward, J., Tubone, G., Khan, S., Chen, H., Nielsen, E., Ionescu, C., Prabhu, U.,

Gao, M., Wang, Q., Augenstein, S., Subramaniam, N., Chang, J., Iliopoulos, F., Luo, J., Khan, M., Kuo, W., Teplyashin, D., Perot, F., Kilpatrick, L., Globerson, A., Yu, H., Siddiqui, A., Sukhanov, N., Kandoor, A., Gupta, U., Andreetto, M., Ambar, M., Kim, D., Wesołowski, P., Perrin, S., Limonchik, B., Fan, W., Stephan, J., Stewart-Binks, I., Kappedal, R., He, T., Cogan, S., Datta, R., Zhou, T., Ye, J., Kieliger, L., Ramalho, A., Kastner, K., Mentzer, F., Ko, W.-J., Suggala, A., Zhou, T., Butt, S., Strejček, H., Belenki, L., Venugopalan, S., Ling, M., Eltyshev, E., Deng, Y., Kovacs, G., Raghavachari, M., Dai, H., Schuster, T., Schwarcz, S., Nguyen, R., Nguyen, A., Buttimore, G., Mallick, S. B., Gandhe, S., Benjamin, S., Jastrzebski, M., Yan, L., Basu, S., Apps, C., Edkins, I., Allingham, J., Odisho, I., Kocisky, T., Zhao, J., Xue, L., Reddy, A., Anastasiou, C., Atias, A., Redmond, S., Milan, K., Heess, N., Schmit, H., Dafoe, A., Andor, D., Gangwani, T., Dragan, A., Zhang, S., Kachra, A., Wu, G., Xue, S., Aydin, K., Liu, S., Zhou, Y., Malihi, M., Wu, A., Gopal, S., Schumann, C., Stys, P., Wang, A., Olšák, M., Liu, D., Schallhart, C., Mao, Y., Brady, D., Xu, H., Mery, T., Sitawarin, C., Velusamy, S., Cobley, T., Zhai, A., Walder, C., Katz, N., Jawahar, G., Kulkarni, C., Yang, A., Paszke, A., Wang, Y., Damoc, B., Borsos, Z., Smith, R., Li, J., Gupta, M., Kapishnikov, A., Prakash, S., Luisier, F., Agarwal, R., Grathwohl, W., Chen, K., Han, K., Mehta, N., Over, A., Azizi, S., Meng, L., Santo, N. D., Zheng, K., Shapiro, J., Petrovski, I., Hui, J., Ghafouri, A., Snoek, J., Qin, J., Jordan, M., Sikora, C., Malmaud, J., Kuang, Y., Świetlik, A., Sang, R., Shi, C., Li, L., Rosenberg, A., Zhao, S., Crawford, A., Peter, J.-T., Lei, Y., Garcia, X., Le, L., Wang, T., Amelot, J., Orr, D., Kacham, P., Alon, D., Tyen, G., Arora, A., Lyon, J., Kurakin, A., Ly, M., Guidroz, T., Yan, Z., Panigrahy, R., Xu, P., Kagohara, T., Cheng, Y., Noland, E., Lee, J., Lee, J., Yip, C., Wang, M., Nehoran, E., Bykovsky, A., Shan, Z., Bhagatwala, A., Yan, C., Tan, J., Garrido, G., Ethier, D., Hurley, N., Vesom, G., Chen, X., Qiao, S., Nayyar, A., Walker, J., Sandhu, P., Rosca, M., Swisher, D., Dektiarev, M., Dillon, J., Muraru, G.-C., Tragut, M., Myaskovsky, A., Reid, D., Velic, M., Xiao, O., George, J., Brand, M., Li, J., Yu, W., Gu, S., Deng, X., Aubet, F.-X., Yeganeh, S. H., Alcober, F., Smith, C., Cohn, T., McKinney, K., Tschannen, M., Sampath, R., Cheon, G., Luo, L., Liu, L., Orbay, J., Peng, H., Botea, G., Zhang, X., Yoon, C., Magalhaes, C., Stradomski, P., Mackinnon, I., Hemingray, S., Venkatesan, K., May, R., Kim, J., Druinsky, A., Ye, J., Xu, Z., Huang, T., Abdallah, J. A., Dostmohamed, A., Fellinger, R., Munkhdalai, T., Maurya, A., Garst, P., Zhang, Y., Krikun, M., Bucher, S., Veerubhotla, A. S., Liu, Y., Li, S., Gupta, N., Adamek, J., Chen, H., Orlando, B., Zaks, A., van Amersfoort, J., Camp, J., Wan, H., Choe, H., Wu, Z., Olszewska, K., Yu, W., Vadali, A., Scholz, M., Freitas, D. D., Lin, J., Hua, A., Liu, X., Ding, F., Zhou, Y., Severson, B., Tsihlas, K., Yang, S., Spalink, T., Yerram, V., Pankov, H., Blevins, R., Vargas, B., Jauhari, S., Miecnikowski, M., Zhang, M., Kumar, S., Farabet, C., Lan, C. L., Flennerhag, S., Bitton, Y., Ma, A., Bražinskas, A., Collins, E., Ahuja, N., Kudugunta, S., Bortsova, A., Giang, M., Zhu, W., Chi, E., Lundberg, S., Stern, A., Puttagunta, S., Xiong, J., Wu, X., Pande, Y., Jhindal, A., Murphy, D., Clark, J., Brockschmidt, M., Deines, M., McKee, K. R., Bahir, D., Shen, J., Truong, M., McDuff, D., Gesmundo, A., Rosseel, E., Liang, B., Caluwaerts, K., Hamrick, J., Kready, J., Cassin, M., Ingale, R., Lao, L., Pollom, S., Ding, Y., He, W., Bellot, L., Iljazi, J., Boppana, R. S., Han, S., Thompson, T., Khalifa, A., Bulanova, A., Mitrevski, B., Pang, B., Cooney, E., Shi, T., Coaguila, R., Yakar, T., Ranzato, M., Momchev, N., Rawles, C., Charles, Z., Maeng, Y., Zhang, Y., Bansal, R., Zhao, X., Albert, B., Yuan, Y., Vijayanarasimhan, S., Hirsch, R., Ramasesh, V., Vodrahalli, K., Wang, X., Gupta, A., Strouse, D., Ni, J., Patel, R., Taubman, G., Huo, Z., Gharibian, D., Monteiro, M., Lam, H., Vasudevan, S., Chaudhary, A., Albuquerque, I., Gupta, K., Riedel, S., Hegde, C., Ruderman, A., György, A., Wainwright, M., Chaugule, A., Ayan, B. K., Levinboim, T., Shleifer, S., Kalley, Y., Mirrokni, V., Rao, A., Radhakrishnan, P., Hartford, J., Wu, J., Zhu, Z., Bertolini, F., Xiong, H., Serrano, N., Tomlinson, H., Ott, M., Chang, Y., Graham, M., Li, J., Liang, M., Long, X., Borgeaud, S., Ahmad, Y., Grills, A., Mincu, D., Izzard, M., Liu, Y., Xie, J., O'Bryan, L., Ponda, S., Tong, S., Liu, M., Malkin, D., Salama, K., Chen, Y., Anil, R., Rao, A., Swavely, R., Bilenko, M., Anderson, N., Tan, T., Xie, J., Wu, X., Yu, L., Vinyals, O., Ryabtsev, A., Dangovski, R., Baumli, K., Keysers, D., Wright, C., Ashwood, Z., Chan, B., Shtefan, A., Guo, Y., Bapna, A., Soricut, R., Pecht, S., Ramos, S., Wang, R., Cai, J., Trinh, T., Barham, P., Friso, L., Stickgold, E., Ding, X., Shakeri, S., Ardila, D., Briakou, E., Culliton, P., Raveret, A., Cui, J., Saxton, D., Roy, S., Azizi, J., Yin, P., Loher, L., Bunner, A., Choi, M., Ahmed, F., Li, E., Li, Y., Dai, S., Elabd, M., Ganapathy, S., Agrawal, S., Hua, Y., Kunkle, P., Rajayogam, S., Ahuja, A., Conmy, A., Vasiloff, A., Beak, P., Yew, C., Mudigonda, J., Wydrowski, B., Blanton, J., Wang, Z., Dauphin, Y., Xu, Z., Polacek, M., Chen, X., Hu, H., Sho, P., Kunesch, M., Manshadi, M. H., Rutherford, E., Li, B., Hsiao, S., Barr, I., Tudor, A., Kecman, M., Nagrani, A., Pchelin, V., Sundermeyer, M., S, A. P., Karmarkar, A., Gao, Y., Chole, G., Bachem, O., Gao, I., BC, A., Dibb, M., Verzetti, M., Hernandez-Campos, F., Lunts, Y., Johnson, M., Trapani, J. D., Koster, R., Brusilovsky, I., Xiong, B., Mohabey, M., Ke, H., Zou, J., Sabolić, T., Campos, V., Palowitch, J., Morris, A., Qiu, L., Ponnuramu, P., Li, F., Sharma, V., Sodhia, K., Tekelioglu, K., Chuklin, A., Yenugula, M., Gemzer, E., Strinopoulos, T., El-Husseini, S., Wang, H., Zhong, Y., Leurent, E., Natsev, P., Wang, W., Mahaarachchi, D., Zhu, T., Peng,

S., Alabed, S., Lee, C.-C., Brohan, A., Szlam, A., Oh, G., Kovsharov, A., Lee, J., Wong, R., Barnes, M., Thornton, G., Gimeno, F., Levy, O., Sevenich, M., Johnson, M., Mallinson, J., Dadashi, R., Wang, Z., Ren, Q., Lahoti, P., Dhar, A., Feldman, J., Zheng, D., Ulrich, T., Panait, L., Blokzijl, M., Baetu, C., Matak, J., Harlalka, J., Shah, M., Marian, T., von Dincklage, D., Du, C., Ley-Wild, R., Brownfield, B., Schumacher, M., Stuken, Y., Noghabi, S., Gupta, S., Ren, X., Malmi, E., Weissenberger, F., Huergo, B., Bauza, M., Lampe, T., Douillard, A., Seyedhosseini, M., Frostig, R., Ghahramani, Z., Nguyen, K., Krishnakumar, K., Ye, C., Gupta, R., Nazari, A., Geirhos, R., Shaw, P., Eleryan, A., Damen, D., Palomaki, J., Xiao, T., Wu, Q., Yuan, Q., Meadowlark, P., Bilotti, M., Lin, R., Sridhar, M., Schroecker, Y., Chung, D.-W., Luo, J., Strohman, T., Liu, T., Zheng, A., Emond, J., Wang, W., Lampinen, A., Fukuzawa, T., Campbell-Ajala, F., Roy, M., Lee-Thorp, J., Wang, L., Naim, I., Tony, ên, N., Bensky, G., Gupta, A., Rogozińska, D., Fu, J., Pillai, T. S., Veličković, P., Drath, S., Neubeck, P., Tulsyan, V., Klimovskiy, A., Metzler, D., Stevens, S., Yeh, A., Yuan, J., Yu, T., Zhang, K., Go, A., Tsang, V., Xu, Y., Wan, A., Galatzer-Levy, I., Sobell, S., Toki, A., Salesky, E., Zhou, W., Antognini, D., Douglas, S., Wu, S., Lelkes, A., Kim, F., Cavallaro, P., Salazar, A., Liu, Y., Besley, J., Refice, T., Jia, Y., Li, Z., Sokolik, M., Kannan, A., Simon, J., Chick, J., Aharon, A., Gandhi, M., Daswani, M., Amiri, K., Birodkar, V., Ittycheriah, A., Grabowski, P., Chang, O., Sutton, C., Zhixin, Lai, Telang, U., Sargsyan, S., Jiang, T., Hoffmann, R., Brichtova, N., Hessel, M., Halcrow, J., Jerome, S., Brown, G., Tomala, A., Buchatskaya, E., Yu, D., Menon, S., Moreno, P., Liao, Y., Zayats, V., Tang, L., Mah, S., Shenoy, A., Siegman, A., Hadian, M., Kwon, O., Tu, T., Khajehnouri, N., Foley, R., Haghani, P., Wu, Z., Keshava, V., Gupta, K., Bruguier, T., Yao, R., Karmon, D., Zintgraf, L., Wang, Z., Piqueras, E., Jung, J., Brennan, J., Machado, D., Giustina, M., Tessler, M., Lee, K., Zhang, Q., Moore, J., Daugaard, K., Frömmgen, A., Beattie, J., Zhang, F., Kasenberg, D., Geri, T., Qin, D., Tomar, G. S., Ouyang, T., Yu, T., Zhou, L., Mathews, R., Davis, A., Li, Y., Gupta, J., Yates, D., Deng, L., Kemp, E., Joung, G.-Y., Vassilvitskii, S., Guo, M., LV, P., Dopson, D., Lachgar, S., McConnaughey, L., Choudhury, H., Dena, D., Cohen, A., Ainslie, J., Levi, S., Gopavarapu, P., Zablotskaia, P., Vallet, H., Bahargam, S., Tang, X., Tomasev, N., Dyer, E., Balle, D., Lee, H., Bono, W., Mendez, J. G., Zubov, V., Yang, S., Rendulic, I., Zheng, Y., Hogue, A., Pundak, G., Leith, R., Bhoopchand, A., Han, M., Žanić, M., Schaul, T., Delakis, M., Iyer, T., Wang, G., Singh, H., Abdelhamed, A., Thomas, T., Brahma, S., Dib, H., Kumar, N., Zhou, W., Bai, L., Mishra, P., Sun, J., Anklin, V., Sukkerd, R., Agubuzu, L., Briukhov, A., Gulati, A., Sieb, M., Pardo, F., Nasso, S., Chen, J., Zhu, K., Sosea, T., Goldin, A., Rush, K., Hombaiah, S. A., Noever, A., Zhou, A., Haves, S., Phuong, M., Ades, J., ting Chen, Y., Yang, L., Pagadora, J., Bileschi, S., Cotruta, V., Saputro, R., Pramanik, A., Ammirati, S., Garrette, D., Villela, K., Blyth, T., Akbulut, C., Jha, N., Rrustemi, A., Wongpanich, A., Nagpal, C., Wu, Y., Rivière, M., Kishchenko, S., Srinivasan, P., Chen, A., Sinha, A., Pham, T., Jia, B., Hennigan, T., Bakalov, A., Attaluri, N., Garmon, D., Rodriguez, D., Wegner, D., Jia, W., Senter, E., Fiedel, N., Petek, D., Liu, Y., Hardin, C., Lehri, H. T., Carreira, J., Smoot, S., Prasetya, M., Akazawa, N., Stefanoiu, A., Ho, C.-H., Angelova, A., Lin, K., Kim, M., Chen, C., Sieniek, M., Li, A., Guo, T., Baltateanu, S., Tafti, P., Wunder, M., Olmert, N., Shukla, D., Shen, J., Kovelamudi, N., Venkatraman, B., Neel, S., Thoppilan, R., Connor, J., Benzing, F., Stjerngren, A., Ghiasi, G., Polozov, A., Howland, J., Weber, T., Chiu, J., Girirajan, G. P., Terzis, A., Wang, P., Li, F., Shalom, Y. B., Tewari, D., Denton, M., Aharoni, R., Kalb, N., Zhao, H., Zhang, J., Filos, A., Rahtz, M., Jain, L., Fan, C., Rodrigues, V., Wang, R., Shin, R., Austin, J., Ring, R., Sanchez-Vargas, M., Hassen, M., Kessler, I., Alon, U., Zhang, G., Chen, W., Ma, Y., Si, X., Hou, L., Mirhoseini, A., Wilson, M., Bacon, G., Roelofs, B., Shu, L., Vasudevan, G., Adler, J., Dwornik, A., Terzi, T., Lawlor, M., Askham, H., Bernico, M., Dong, X., Hidey, C., Kilgour, K., Liu, G., Bhupatiraju, S., Leonhard, L., Zuo, S., Talukdar, P., Wei, Q., Severyn, A., Listík, V., Lee, J., Tripathi, A., Park, S., Matias, Y., Liu, H., Ruiz, A., Jayaram, R., Tolins, J., Marcenac, P., Wang, Y., Seybold, B., Prior, H., Sharma, D., Weber, J., Sirotenko, M., Sung, Y., Du, D., Pavlick, E., Zinke, S., Freitag, M., Dylla, M., Arenas, M. G., Potikha, N., Goldman, O., Tao, C., Chhaparia, R., Voitovich, M., Dogra, P., Ražnatović, A., Tsai, Z., You, C., Johnson, O., Tucker, G., Gu, C., Yoo, J., Majzoubi, M., Gabeur, V., Raad, B., Rhodes, R., Kolipaka, K., Howard, H., Sampemane, G., Li, B., Asawaroengchai, C., Nguyen, D., Zhang, C., Cour, T., Yu, X., Fu, Z., Jiang, J., Huang, P.-S., Surita, G., Iturrate, I., Karov, Y., Collins, M., Baeuml, M., Fuchs, F., Shetty, S., Ramaswamy, S., Ebrahimi, S., Guo, Q., Shar, J., Barth-Maron, G., Addepalli, S., Richter, B., Cheng, C.-Y., Rives, E., Zheng, F., Griesser, J., Dikkala, N., Zeldes, Y., Safarli, I., Das, D., Srivastava, H., Khan, S. M., Li, X., Pandey, A., Markeeva, L., Belov, D., Yan, Q., Rybiński, M., Chen, T., Nawhal, M., Quinn, M., Govindaraj, V., York, S., Roberts, R., Garg, R., Godbole, N., Abernethy, J., Das, A., Thiet, L. N., Tompson, J., Nham, J., Vats, N., Caine, B., Helmholz, W., Pongetti, F., Ko, Y., An, J., Hu, C. H., Ling, Y.-C., Pawar, J., Leland, R., Kinoshita, K., Khawaja, W., Selvi, M., Ie, E., Sinopalnikov, D., Proleev, L., Tripuraneni, N., Bevilacqua, M., Lee, S., Sanford, C., Suh, D., Tran, D., Dean, J., Baumgartner, S., Heitkaemper, J., Gubbi, S., Toutanova, K., Xu, Y., Thekkath, C., Rong, K., Jain, P., Xie, A., Virin, Y., Li, Y., Litchev, L., Powell, R., Bharti, T., Kraft, A., Hua, N., Ikonomidis, M., Hitron, A.,

Kumar, S., Matthey, L., Bridgers, S., Lax, L., Malhi, I., Skopek, O., Gupta, A., Cao, J., Rasquinha, M., Põder, S., Stokowiec, W., Roth, N., Li, G., Sander, M., Kessinger, J., Jain, V., Loper, E., Park, W., Yarom, M., Cheng, L., Guruganesh, G., Rao, K., Li, Y., Barros, C., Sushkov, M., Ferng, C.-S., Shah, R., Aharoni, O., Kumar, R., McConnell, T., Li, P., Wang, C., Pereira, F., Swanson, C., Jamil, F., Xiong, Y., Vijayakumar, A., Shroff, P., Soparkar, K., Gu, J., Soares, L. B., Wang, E., Majmundar, K., Wei, A., Bailey, K., Kassner, N., Kawamoto, C., Žužić, G., Gomes, V., Gupta, A., Guzman, M., Dasgupta, I., Bai, X., Pan, Z., Piccinno, F., Vogel, H. N., Ponce, O., Hutter, A., Chang, P., Jiang, P.-P., Gog, I., Ionescu, V., Manyika, J., Pedregosa, F., Ragan, H., Behrman, Z., Mullins, R., Devin, C., Pyne, A., Gawde, S., Chadwick, M., Gu, Y., Tavakkol, S., Twigg, A., Goyal, N., Elue, N., Goldie, A., Venkatachary, S., Fei, H., Feng, Z., Ritter, M., Leal, I., Dasari, S., Sun, P., Rochman, A. R., O'Donoghue, B., Liu, Y., Sproch, J., Chen, K., Clay, N., Petrov, S., Sidhwani, S., Mihailescu, I., Panagopoulos, A., Piergiovanni, A., Bai, Y., Powell, G., Karkhanis, D., Yacovone, T., Mitrichev, P., Kovac, J., Uthus, D., Yazdanbakhsh, A., Amos, D., Zheng, S., Zhang, B., Miao, J., Ramabhadran, B., Radpour, S., Thakoor, S., Newlan, J., Lang, O., Jankowski, O., Bharadwaj, S., Sarr, J.-M., Ashraf, S., Mondal, S., Yan, J., Rawat, A. S., Velury, S., Kochanski, G., Eccles, T., Och, F., Sharma, A., Mahintorabi, E., Gurney, A., Muir, C., Cohen, V., Thakur, S., Bloniarz, A., Mujika, A., Pritzel, A., Caron, P., Rahman, A., Lang, F., Onoe, Y., Sirkovic, P., Hoover, J., Jian, Y., Duque, P., Narayanan, A., Soergel, D., Haig, A., Maggiore, L., Buch, S., Dean, J., Figotin, I., Karpov, I., Gupta, S., Zhou, D., Huang, M., Vaswani, A., Semturs, C., Shivakumar, K., Watanabe, Y., Rajendran, V. K., Lu, E., Hou, Y., Ye, W., Vashishth, S., Nti, N., Sakenas, V., Ni, D., DeCarlo, D., Bendersky, M., Bagri, S., Cano, N., Peake, E., Tokumine, S., Godbole, V., Guía, C., Lando, T., Selo, V., Ellis, S., Tarlow, D., Gillick, D., Epasto, A., Jonnalagadda, S. R., Wei, M., Xie, M., Taly, A., Paganini, M., Sundararajan, M., Toyama, D., Yu, T., Petrova, D., Pappu, A., Agrawal, R., Buthpitiya, S., Frye, J., Buschmann, T., Crocker, R., Tagliasacchi, M., Wang, M., Huang, D., Perel, S., Wieder, B., Kazawa, H., Wang, W., Cole, J., Gupta, H., Golan, B., Bang, S., Kulkarni, N., Franko, K., Liu, C., Reid, D., Dalmia, S., Whang, J., Cen, K., Sundaram, P., Ferret, J., Isik, B., Ionita, L., Sun, G., Shekhawat, A., Mohammad, M., Pham, P., Huang, R., Raman, K., Zhou, X., Mcilroy, R., Myers, A., Peng, S., Scott, J., Covington, P., Erell, S., Joshi, P., Oliveira, J. G., Noy, N., Nasir, T., Walker, J., Axelrod, V., Dozat, T., Han, P., Chu, C.-T., Weinstein, E., Shukla, A., Chandrakaladharan, S., Poklukar, P., Li, B., Jin, Y., Eruvbetine, P., Hansen, S., Dabush, A., Jacovi, A., Phatale, S., Zhu, C., Baker, S., Shomrat, M., Xiao, Y., Pouget-Abadie, J., Zhang, M., Wei, F., Song, Y., King, H., Huang, Y., Zhu, Y., Sun, R., Franco, J. V., Lin, C.-C., Arora, S., Hui, Li, Xia, V., Vilnis, L., Schain, M., Alarakyia, K., Prince, L., Phillips, A., Habtegebriel, C., Xu, L., Gui, H., Ontanon, S., Aroyo, L., Gill, K., Lu, P., Katariya, Y., Madeka, D., Krishnan, S., Raghvendra, S. S., Freedman, J., Tay, Y., Menghani, G., Choy, P., Shetty, N., Abolafia, D., Kukliansky, D., Chou, E., Lichtarge, J., Burke, K., Coleman, B., Guo, D., Jin, L., Bhattacharya, I., Langston, V., Li, Y., Kotecha, S., Yakubovich, A., Chen, X., Petrov, P., Powell, T., He, Y., Quick, C., Garg, K., Hwang, D., Lu, Y., Bhojanapalli, S., Kjems, K., Mehran, R., Archer, A., van Hasselt, H., Balakrishna, A., Kearns, J., Guo, M., Riesa, J., Sazanovich, M., Gao, X., Sauer, C., Yang, C., Sheng, X., Jimma, T., Gansbeke, W. V., Nikolaev, V., Wei, W., Millican, K., Zhao, R., Snyder, J., Bolelli, L., O'Brien, M., Xu, S., Xia, F., Yuan, W., Neelakantan, A., Barker, D., Yadav, S., Kirkwood, H., Ahmad, F., Wee, J., Grimstad, J., Wang, B., Wiethoff, M., Settle, S., Wang, M., Blundell, C., Chen, J., Duvarney, C., Hu, G., Ronneberger, O., Lee, A., Li, Y., Chakladar, A., Butryna, A., Evangelopoulos, G., Desjardins, G., Kanerva, J., Wang, H., Nowak, A., Li, N., Loo, A., Khurshudov, A., Shafey, L. E., Baddi, N., Lenc, K., Razeghi, Y., Lieber, T., Sinha, A., Ma, X., Su, Y., Huang, J., Ushio, A., Klimczak-Plucińska, H., Mohamed, K., Chen, J., Osindero, S., Ginzburg, S., Lamprou, L., Bashlovkina, V., Tran, D.-H., Khodaei, A., Anand, A., Di, Y., Eskander, R., Vuyyuru, M. R., Liu, J., Kamath, A., Goldenberg, R., Bellaiche, M., Pluto, J., Rosgen, B., Mansoor, H., Wong, W., Ganesh, S., Bailey, E., Baird, S., Deutsch, D., Baek, J., Jia, X., Lee, C., Friesen, A., Braun, N., Lee, K., Panda, A., Hernandez, S. M., Williams, D., Liu, J., Liang, E., Autef, A., Pitler, E., Jain, D., Kirk, P., Bunyan, O., Elias, J. S., Yin, T., Reid, M., Pope, A., Putikhin, N., Samanta, B., Guadarrama, S., Kim, D., Rowe, S., Valentine, M., Yan, G., Salcianu, A., Silver, D., Song, G., Singh, R., Ye, S., DeBalsi, H., Merey, M. A., Ofek, E., Webson, A., Mourad, S., Kakarla, A., Lattanzi, S., Roy, N., Sluzhaev, E., Butterfield, C., Tonioni, A., Waters, N., Kopalle, S., Chase, J., Cohan, J., Rao, G. R., Berry, R., Voznesensky, M., Hu, S., Chiafullo, K., Chikkerur, S., Scrivener, G., Zheng, I., Wiesner, J., Macherey, W., Lillicrap, T., Liu, F., Walker, B., Welling, D., Davies, E., Huang, Y., Ren, L., Shabat, N., Agostini, A., Iinuma, M., Zelle, D., Sathyanarayana, R., D'olimpio, A., Redshaw, M., Ginsberg, M., Murthy, A., Geller, M., Matejovicova, T., Chakrabarti, A., Julian, R., Chan, C., Hu, Q., Jarrett, D., Agarwal, M., Challagundla, J., Li, T., Tata, S., Ding, W., Meng, M., Dai, Z., Vezzani, G., Garg, S., Bulian, J., Jasarevic, M., Cai, H., Rajamani, H., Santoro, A., Hartmann, F., Liang, C., Perz, B., Jindal, A., Bu, F., Seo, S., Poplin, R., Goedeckemeyer, A., Ghazi, B., Khadke, N., Liu, L., Mather, K., Zhang, M., Shah, A., Chen, A., Wei, J., Shivam, K., Cao, Y., Cho, D., Scarpati, A. S., Moffitt, M., Barbu, C., Jurin, I., Chang, M.-W.,

Liu, H., Zheng, H., Dave, S., Kaeser-Chen, C., Yu, X., Abdagic, A., Gonzalez, L., Huang, Y., Zhong, P., Schmid, C., Petrini, B., Wertheim, A., Zhu, J., Nguyen, H., Ji, K., Zhou, Y., Zhou, T., Feng, F., Cohen, R., Rim, D., Phal, S. M., Georgiev, P., Brand, A., Ma, Y., Li, W., Gupta, S., Wang, C., Dubov, P., Tarbouriech, J., Majumder, K., Li, H., Rink, N., Suman, A., Guo, Y., Sun, Y., Nair, A., Xu, X., Elhawaty, M., Cabrera, R., Han, G., Eisenschlos, J., Bai, J., Li, Y., Bansal, Y., Sellam, T., Khan, M., Nguyen, H., Mao-Jones, J., Parotsidis, N., Marcus, J., Fan, C., Zimmermann, R., Kochinski, Y., Graesser, L., Behbahani, F., Caceres, A., Riley, M., Kane, P., Lefdal, S., Willoughby, R., Vicol, P., Wang, L., Zhang, S., Gill, A., Liang, Y., Prasad, G., Mariooryad, S., Kazemi, M., Wang, Z., Muralidharan, K., Voigtlaender, P., Zhao, J., Zhou, H., D'Souza, N., Mavalankar, A., Arnold, S., Young, N., Sarvana, O., Lee, C., Nasr, M., Zou, T., Kim, S., Haas, L., Patel, K., Bulut, N., Parkinson, D., Biles, C., Kalashnikov, D., To, C. M., Kumar, A., Austin, J., Greve, A., Zhang, L., Goel, M., Li, Y., Yaroshenko, S., Chang, M., Jindal, A., Clark, G., Taitelbaum, H., Johnson, D., Roval, O., Ko, J., Mohananey, A., Schuler, C., Dodhia, S., Li, R., Osawa, K., Cui, C., Xu, P., Shah, R., Huang, T., Gruzewska, E., Clement, N., Verma, M., Sercinoglu, O., Qian, H., Shah, V., Yamaguchi, M., Modi, A., Kosakai, T., Strohmann, T., Zeng, J., Gunel, B., Qian, J., Tarango, A., Jastrzebski, K., David, R., Shan, J., Schuh, P., Lad, K., Gierke, W., Madhavan, M., Chen, X., Kurzeja, M., Santamaria-Fernandez, R., Chen, D., Cordell, A., Chervonyi, Y., Garcia, F., Kannen, N., Perot, V., Ding, N., Cohen-Ganor, S., Lavrenko, V., Wu, J., Evans, G., dos Santos, C. N., Sewak, M., Brown, A., Hard, A., Puigcerver, J., Zheng, Z., Liang, Y., Gladchenko, E., Ingle, R., First, U., Sermanet, P., Magister, C., Velimirović, M., Reddi, S., Ricco, S., Agustsson, E., Adam, H., Levine, N., Gaddy, D., Holtmann-Rice, D., Wang, X., Sathe, A., Roy, A. G., Bratanič, B., Carin, A., Mehta, H., Bonacina, S., Cao, N. D., Finkelstein, M., Rieser, V., Wu, X., Altché, F., Scandinaro, D., Li, L., Vieillard, N., Sethi, N., Tanzer, G., Xing, Z., Wang, S., Bhatia, P., Citovsky, G., Anthony, T., Lin, S., Shi, T., Jakobovits, S., Gibson, G., Apte, R., Lee, L., Chen, M., Byravan, A., Maniatis, P., Webster, K., Dai, A., Chen, P.-C., Pan, J., Fadeeva, A., Gleicher, Z., Luong, T., and Bhumihar, N. K. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Dao, T. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mZn2Xyh9Ec.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

DeepSeek-AI, Liu, A., Feng, B., Wang, B., Wang, B., Liu, B., Zhao, C., Dengr, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Xu, H., Yang, H., Zhang, H., Ding, H., Xin, H., Gao, H., Li, H., Qu, H., Cai, J. L., Liang, J., Guo, J., Ni, J., Li, J., Chen, J., Yuan, J., Qiu, J., Song, J., Dong, K., Gao, K., Guan, K., Wang, L., Zhang, L., Xu, L., Xia, L., Zhao, L., Zhang, L., Li, M., Wang, M., Zhang, M., Zhang, M., Tang, M., Li, M., Tian, N., Huang, P., Wang, P., Zhang, P., Zhu, Q., Chen, Q., Du, Q., Chen, R. J., Jin, R. L., Ge, R., Pan, R., Xu, R., Chen, R., Li, S. S., Lu, S., Zhou, S., Chen, S., Wu, S., Ye, S., Ma, S., Wang, S., Zhou, S., Yu, S., Zhou, S., Zheng, S., Wang, T., Pei, T., Yuan, T., Sun, T., Xiao, W. L., Zeng, W., An, W., Liu, W., Liang, W., Gao, W., Zhang, W., Li, X. Q., Jin, X., Wang, X., Bi, X., Liu, X., Wang, X., Shen, X., Chen, X., Chen, X., Nie, X., Sun, X., Wang, X., Liu, X., Xie, X., Yu, X., Song, X., Zhou, X., Yang, X., Lu, X., Su, X., Wu, Y., Li, Y. K., Wei, Y. X., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zhao, Y., Sun, Y., Li, Y., Wang, Y., Zheng, Y., Zhang, Y., Xiong, Y., Zhao, Y., He, Y., Tang, Y., Piao, Y., Dong, Y., Tan, Y., Liu, Y., Wang, Y., Guo, Y., Zhu, Y., Wang, Y., Zou, Y., Zha, Y., Ma, Y., Yan, Y., You, Y., Liu, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Huang, Z., Zhang, Z., Xie, Z., Hao, Z., Shao, Z., Wen, Z., Xu, Z., Zhang, Z., Li, Z., Wang, Z., Gu, Z., Li, Z., and Xie, Z. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *arXiv preprint arXiv:2405.04434*, 2024.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su,

X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. Deepseek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025.

Du, K., Wang, B., Zhang, C., Cheng, Y., Lan, Q., Sang, H., Cheng, Y., Yao, J., Liu, X., Qiao, Y., Stoica, I., and Jiang, J. PrefillOnly: An Inference Engine for Prefill-only Workloads in Large Language Model Applications. *arXiv preprint arXiv:2505.07203*, 2025.

Fedus, W., Zoph, B., and Shazeer, N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Feng, J., Huang, Y., Zhang, R., Liang, S., Yan, M., and Wu, J. WindServe: Efficient Phase-Disaggregated LLM Serving with Stream-based Dynamic Scheduling. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, pp. 1283–1295, 2025.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli,

J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

Hanindhito, B. and John, L. K. Accelerating ML Workloads using GPU Tensor Cores: The Good, the Bad, and the Ugly. In *Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering*, pp. 178–189, 2024.

Jouppi, N. P., Kurian, G., Li, S., Ma, P., Nagarajan, R., Nai, L., Patil, N., Subramanian, S., Swing, A., Towles, B., Young, C., Zhou, X., Zhou, Z., and Patterson, D. TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings. In *Proceedings of the 50th annual international symposium on computer architecture*, pp. 1–14, 2023.

Kamath, A. K., Prabhu, R., Mohan, J., Peter, S., Ramjee, R., and Panwar, A. POD-Attention: Unlocking Full Prefill-Decode Overlap for Faster LLM Inference. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pp. 897–912, 2025.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding . *International Conference on Learning Representations*, 2021.

Leviathan, Y., Kalman, M., and Matias, Y. Fast Inference from Transformers via Speculative Decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Mitra, T., Borkar, R., Bhatia, N., Matas, R., Raj, S., Mudigere, D., Zhao, R., Golub, M., Dutta, A., Madduri, S., Jani, D., Pharris, B., and Rouhani, B. D. Beyond the Buzz: A Pragmatic Take on Inference Disaggregation. *arXiv preprint arXiv:2506.05508*, 2025.

Navarro, C. A., Carrasco, R., Barrientos, R. J., Riquelme, J. A., and Vega, R. GPU Tensor Cores for fast Arithmetic Reductions. *IEEE Transactions on Parallel and Distributed Systems*, 32(1):72–84, 2020.

Norrie, T., Patil, N., Yoon, D. H., Kurian, G., Li, S., Laudon, J., Young, C., Jouppi, N., and Patterson, D. The Design Process for Google's Training Chips: TPUv2 and TPUv3. *IEEE Micro*, 41(2):56–63, 2021.

NVIDIA. FasterTransformer, 2019. URL https://github.com/NVIDIA/FasterTransformer.

NVIDIA. NVIDIA Dynamo. https://docs.nvidia.com/dynamo/latest, 2025a. Accessed: 2025-10-01.

NVIDIA. NVIDIA Management Library (NVML). https://developer.nvidia.com/management-library-nvml, 2025b. Accessed: 2025-10-01.

OpenAI, Agarwal, S., Ahmad, L., Ai, J., Altman, S., Applebaum, A., Arbus, E., Arora, R. K., Bai, Y., Baker, B., Bao, H., Barak, B., Bennett, A., Bertao, T., Brett, N., Brevdo, E., Brockman, G., Bubeck, S., Chang, C., Chen, K., Chen, M., Cheung, E., Clark, A., Cook, D., Dukhan, M., Dvorak, C., Fives, K., Fomenko, V., Garipov, T., Georgiev, K., Glaese, M., Gogineni, T., Goucher, A., Gross, L., Guzman, K. G., Hallman, J., Hehir, J., Heidecke, J., Helyar, A., Hu, H., Huet, R., Huh, J., Jain, S., Johnson, Z., Koch, C., Kofman, I., Kundel, D., Kwon, J., Kyrylov, V., Le, E. Y., Leclerc, G., Lennon, J. P., Lessans, S., Lezcano-Casado, M., Li, Y., Li, Z., Lin, J., Liss, J., Lily, Liu, Liu, J., Lu, K., Lu, C., Martinovic, Z., McCallum, L., McGrath, J., McKinney, S., McLaughlin, A., Mei, S., Mostovoy, S., Mu, T., Myles, G., Neitz, A., Nichol, A., Pachocki, J., Paino, A., Palmie, D., Pantuliano, A., Parascandolo, G., Park, J., Pathak, L., Paz, C., Peran, L., Pimenov, D., Pokrass, M., Proehl, E., Qiu, H., Raila, G., Raso, F., Ren, H., Richardson, K., Robinson, D., Rotsted, B., Salman, H., Sanjeev, S., Schwarzer, M., Sculley, D., Sikchi, H., Simon, K., Singhal, K., Song, Y., Stuckey, D., Sun, Z., Tillet, P., Toizer, S., Tsimpourlas, F., Vyas, N., Wallace, E., Wang, X., Wang, M., Watkins, O., Weil, K., Wendling, A., Whinnery, K., Whitney, C., Wong, H., Yang, L., Yang, Y., Yasunaga, M., Ying, K., Zaremba, W., Zhan, W., Zhang, C., Zhang, B., Zhang, E., and Zhao, S. gpt-oss-120b & gpt-oss-20b Model Card. *arXiv preprint arXiv:2508.10925*, 2025.

Park, J., Choi, J., Kyung, K., Kim, M. J., Kwon, Y., Kim, N. S., and Ahn, J. AttAcc! Unleashing the Power of PIM for Batched Transformer-based Generative Model Inference. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pp. 103–119, 2024.

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

Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Levskaya, A., Heek, J., Xiao, K., Agrawal, S., and Dean, J. Efficiently Scaling Transformer Inference, 2022. URL https://arxiv.org/abs/2211.05102.

Raha, A., Mathaikutty, D. A., Kundu, S., and Ghosh, S. K. FlexNPU: a dataflow-aware flexible deep learning accelerator for energy-efficient edge devices. *Frontiers in High Performance Computing*, 3:1570210, 2025.

Rajbhandari, S., Li, C., Yao, Z., Zhang, M., Aminabadi, R. Y., Awan, A. A., Rasley, J., and He, Y. DeepSpeed-MoE: Advancing Mixture-of-Experts Inference and Training to Power Next-Generation AI Scale. In *International conference on machine learning*, pp. 18332–18346. PMLR, 2022.

Shah, J., Bikshandi, G., Zhang, Y., Thakkar, V., Ramani, P., and Dao, T. FlashAttention-3: Fast and Accurate Attention with Asynchrony and Low-precision. *Advances in Neural Information Processing Systems*, 37:68658–68685, 2024.

ShareGPT. ShareGPT, 2023. URL https://sharegpt.com/.

Shazeer, N. GLU Variants Improve Transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *International Conference on Learning Representations*, 2017.

Sheng, Y., Zheng, L., Yuan, B., Li, Z., Ryabinin, M., Chen, B., Liang, P., Ré, C., Stoica, I., and Zhang, C. FlexGen: High-Throughput Generative Inference of Large Language Models with a Single GPU. In *International Conference on Machine Learning*, pp. 31094–31116. PMLR, 2023.

Shi, T. and Ding, Y. Systematic Characterization of LLM Quantization: A Performance, Energy, and Quality Perspective, 2025. URL https://arxiv.org/abs/2508.16712.

Stojkovic, J., Zhang, C., Goiri, Í., Torrellas, J., and Choukse, E. DynamoLLM: Designing LLM Inference Clusters for Performance and Energy Efficiency. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 1348–1362. IEEE, 2025.

Tirumala, A. and Wong, R. NVIDIA Blackwell Platform: Advancing Generative AI and Accelerated Computing. In *2024 IEEE Hot Chips 36 Symposium (HCS)*, pp. 1–33. IEEE Computer Society, 2024.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is All You Need. *Advances in neural information processing systems*, 30, 2017.

Wang, C., Zuo, P., Chen, Z., Liang, Y., Yu, Z., and Yang, M.-C. Prefill-Decode Aggregation or Disaggregation? Unifying Both for Goodput-Optimized LLM Serving. *arXiv preprint arXiv:2508.01989*, 2025a.

Wang, D., Liu, B., Lu, R., Zhang, Z., and Zhu, S. StoreLLM: Energy Efficient Large Language Model Inference with

Permanently Pre-stored Attention Matrices. In *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*, pp. 398–406, 2025b.

Wang, Y., Wang, Q., Shi, S., He, X., Tang, Z., Zhao, K., and Chu, X. Benchmarking the Performance and Energy Efficiency of AI Accelerators for AI Training. In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, pp. 744–751. IEEE, 2020.

Wilhelm, P., Wittkopp, T., and Kao, O. Beyond Test-Time Compute Strategies: Advocating Energy-per-Token in LLM Inference. In *Proceedings of the 5th Workshop on Machine Learning and Systems*, pp. 208–215, 2025.

Wolters, C., Yang, X., Schlichtmann, U., and Suzumura, T. Memory Is All You Need: An Overview of Compute-in-Memory Architectures for Accelerating Large Language Model Inference. *arXiv preprint arXiv:2406.08413*, 2024.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388.

You, J., Chung, J.-W., and Chowdhury, M. Zeus: Understanding and Optimizing GPU Energy Consumption of DNN Training. In *USENIX NSDI*, 2023.

Yu, G.-I., Jeong, J. S., Kim, G.-W., Kim, S., and Chun, B.-G. Orca: A Distributed Serving System for Transformer-Based Generative Models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pp. 521–538, 2022.

Yun, S., Park, S., Nam, H., Lee, Y., Lee, G., Kyung, K., Kim, S., Kim, N. S., Kim, J., Kim, H., Cho, J., Baek, S., and Ahn, J. The New LLM Bottleneck: A Systems Perspective on Latent Attention and Mixture-of-Experts, 2025. URL https://arxiv.org/abs/2507.15465.

Zhong, Y., Liu, S., Chen, J., Hu, J., Zhu, Y., Liu, X., Jin, X., and Zhang, H. DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 193–210, 2024.

Zoph, B., Bello, I., Kumar, S., Du, N., Huang, Y., Dean, J., Shazeer, N., and Fedus, W. ST-MoE: Designing Stable and Transferable Sparse Expert Models. *arXiv preprint arXiv:2202.08906*, 2022.