Reasoning-Aware SLOs: Why TTFT and TBT Are the Wrong Metrics for o1/R1-Class Models

Mysore Supreeth
Principal AI Architect, Intel Labs
dr.mysores@gmail.com

***Abstract*—Time-To-First-Token (TTFT) and Time-Between-Tokens (TBT) have become the de facto service-level objectives (SLOs) for large language model (LLM) serving. We argue this reflex is miscalibrated for reasoning models—o1 [3] and DeepSeek-R1 [1]—whose user-perceived value is dominated by hidden chain-of-thought, best-of-N search, and verifier loops [5]. A model that emits its first thinking token in 50 ms but then spends 90 seconds exploring branches the user never sees is not well-served by a TTFT target. We propose three replacement metrics: *Time-To-Verified-Answer* (TTVA), *Useful-Token Ratio* (UTR), and *Abandoned-Branch Energy* (ABE). Each captures a workload property invisible to TTFT/TBT, and each implies different scheduler, disaggregation, and accounting decisions in serving stacks [2], [9], [10]. This position paper formalizes the metrics, walks through a worked reasoning trace to show how each is computed, contrasts the new SLOs against classical ones in a side-by-side comparison, sketches a budget-aware admission-control algorithm, and argues that benchmark suites and serving SLAs must be retooled before reasoning-class workloads dominate inference traffic.**

I. Introduction

A decade of LLM-serving research has crystallized around two latency metrics: TTFT, the wall-clock from request arrival to the first response token, and TBT (or inter-token latency / ITL), the steady-state cadence of subsequent tokens. Modern engines such as vLLM [2] and SGLang [9] are tuned against these targets; disaggregated systems like DistServe [10] explicitly co-design prefill and decode to hit them. KV-cache compaction techniques such as StreamingLLM [7] and H2O [8] are similarly evaluated against TTFT/TBT-derived throughput.

These metrics encode a clean assumption: every generated token is delivered to the user, and the user's perceived quality of service is a smooth function of how quickly tokens stream out. Reasoning models break that assumption. o1 [3] and R1 [1] routinely generate *thousands* of internal "thought" tokens that the user neither sees nor wants in real time, and test-time-scaling recipes [5] amplify this with best-of-N, tree search, and self-consistency loops. The token a user actually waits for is not the first one—it is the first *verified* one.

Figure 1 previews the gap. The classic-LLM bar is TTFT-dominated and TBT-smooth; the reasoning bar is TTFT-trivial but TTVA-dominated. Optimizing TTFT here is, at best, a measurement of prefill speed; at worst, a Goodhart's-Law trap that rewards systems for emitting useless tokens fast. We make three claims: (1) current SLOs systematically misrepresent reasoning workloads; (2) three new metrics—TTVA, UTR, ABE—capture what TTFT/TBT miss; (3) adopting them

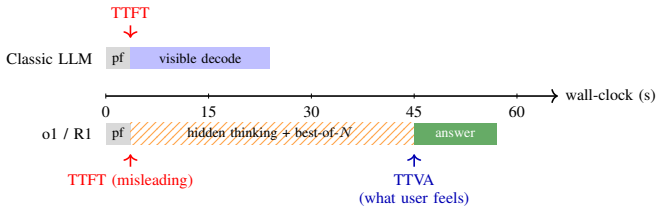

Fig. 1. A reasoning-model request emits tokens long before any of them are useful. TTFT clocks the first thought token; the user only experiences quality at the first *verified-answer* token.

changes scheduling, disaggregation, and energy-accounting choices in non-trivial ways.

a) Scope.: We deliberately do *not* argue that TTFT/TBT should be retired—they remain useful low-level diagnostics for the prefill and decode pipelines they were designed to characterize. We argue they should no longer serve as the *contractual* SLOs against which reasoning-class systems are judged. Concretely, the proposed metrics are intended for inclusion in (i) public serving-benchmark leaderboards, (ii) cloud SLAs for reasoning APIs, and (iii) inference-side carbon disclosures.

II. Why Current SLOs Miss Reasoning Workloads

We identify three concrete failure modes.

a) (a) TTFT measures prefill, not user-visible progress.: TTFT was introduced as a proxy for "time until the user sees the model is working." For a reasoning model with a hidden scratchpad, the first token is structurally invisible: it lives inside a `<thinking>` block that the product surface either hides outright or summarizes post-hoc. Prefill optimizations [10] that shave 100 ms off TTFT have zero effect on the user's experience when the model will spend tens of seconds thinking before any visible output. The scheduling decisions those optimizations imply (e.g., aggressive prefill prioritization, chunked prefill) may even *hurt* reasoning workloads by starving decode-bound thought streams.

b) (b) TBT averages over chain-of-thought tokens.: TBT is reported as a mean or p99 across all generated tokens. In a reasoning trace, the vast majority of those tokens belong to internal chain-of-thought; the handful of tokens the user actually reads (the final answer) are a tiny tail. A system that hits a 30 ms p99 TBT across 4,000 thought tokens but stalls 800 ms on the final-answer commit *passes* its TBT SLO and

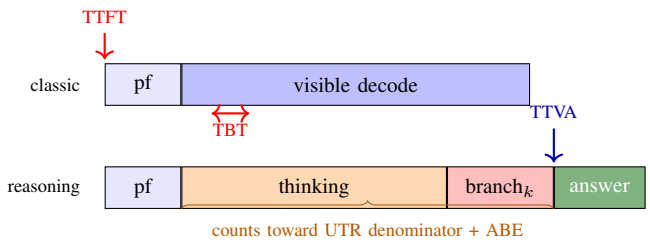

Fig. 2. Where the SLO clock should start and stop. Classic SLOs bracket prefill→decode; reasoning SLOs must bracket request-arrival→verified answer, and must *charge* for the orange/red regions.

TABLE I
CLASSICAL STREAMING-CHATBOT SLOS VS. REASONING-AWARE SLOS. EACH COLUMN LISTS WHAT A CONTRACTUAL METRIC IN THAT FAMILY WOULD ACTUALLY CONSTRAIN.

Axis	**Classical (TTFT, TBT)**	**Reasoning-aware (TTVA, UTR, ABE)**
Clock starts at	request arrival	request arrival
Clock stops at	first emitted token / per-token cadence	first *verified-answer* token
Tokens counted	all emitted tokens	answer tokens vs. thought vs. abandoned
Branch fan-out	not represented	explicit ($\mathcal{B}$, $b^\star$)
Energy on rejected work	not represented	first-class via ABE
Failure mode if optimized in isolation	fast hidden tokens, slow answer commit	branch starvation, premature commit

fails its user. KV-cache eviction heuristics [7], [8] tuned to optimize average TBT may evict the very prefix tokens the verifier needs to validate the final answer.

c) (c) Success metrics ignore abandoned branches.: Best-of-N and tree-of-thought search produce a single accepted trajectory and $N-1$ discarded ones. Standard "tokens/sec" and "requests/sec" throughput numbers count all generated tokens equally—including the discarded ones—so a system can look highly utilized while burning the bulk of its FLOPs on branches that will be thrown away. Conversely, a quality metric like "pass@1" or "verifier-accept rate" tells us nothing about the energy or dollar cost incurred to reach it. The carbon-accounting literature [4], [6] has long argued that aggregate-FLOP metrics obscure work that does not contribute to the deliverable; reasoning models make this concrete.

Figure 2 contrasts the two regimes. The point is not that TTFT/TBT are wrong in absolute terms—they remain useful low-level diagnostics—but that they are the wrong *contractual* metrics for reasoning workloads.

Table I sharpens the contrast across five operational axes that matter for SLA writing: *what is being measured*, *when the clock stops*, *which tokens count*, *whether branch fan-out is captured*, and *whether energy on discarded work is charged*. Classical SLOs are silent on the last three rows.

III. THREE PROPOSED METRICS

We propose TTVA, UTR, and ABE as first-class SLOs for reasoning serving. Table II summarizes them.

TABLE II
REASONING-AWARE SLO METRICS AT A GLANCE.

Metric	**Captures**	**Replaces / augments**
TTVA	Latency to validated output	TTFT, end-to-end latency
UTR	Wasted-token fraction	Raw tokens/sec throughput
ABE	Energy on discarded branches	FLOP-based carbon proxies

a) Time-To-Verified-Answer (TTVA).: Let t_{arr} be request-arrival time and t_{ver} the wall-clock at which a verifier (model-internal, programmatic, or external) emits an `accept` decision on the final answer. Then

$$\text{TTVA} = t_{\text{ver}} - t_{\text{arr}}. \tag{1}$$

TTVA collapses TTFT, TBT, and verifier latency into one number that correlates with user-felt quality. SLAs should be stated as p50/p99 TTVA under a workload mix, not p99 TTFT.

b) Useful-Token Ratio (UTR).: Let T_{ans} be the number of tokens in the final delivered answer and T_{gen} the total tokens generated for the request, including all hidden thought tokens and all tokens emitted on pruned best-of-N branches. Then

$$\text{UTR} = \frac{T_{\text{ans}}}{T_{\text{gen}}}, \qquad 0 < \text{UTR} \leq 1. \tag{2}$$

UTR exposes the "token amplification factor" of a reasoning request and is invariant to engine speed: a $10\times$ faster decoder with the same search policy has the same UTR. It belongs alongside throughput in serving leaderboards.

c) Abandoned-Branch Energy (ABE).: Let $\mathcal{B}$ be the set of branches explored for a request, with $b^\star \in \mathcal{B}$ the accepted branch. Let $E(b)$ be the energy (kWh, measured at the accelerator or rack) consumed generating branch b, including its share of prefill and KV-cache occupancy. Then

$$\text{ABE} = \sum_{b \in \mathcal{B} \setminus \{b^\star\}} E(b). \tag{3}$$

ABE is the reasoning-era analog of the embodied/operational-carbon distinction in [4], [6]: it isolates the energy spent on work the customer never receives. Charge-back models in shared serving stacks should bill ABE separately from $E(b^\star)$.

A. Decompositions and Refinements

The three definitions above are intentionally headline-form. For benchmarking and SLA negotiation, three refinements are useful.

a) TTVA decomposition.: TTVA admits a natural breakdown into four phases that map onto distinct hardware regimes:

$$\text{TTVA} = \underbrace{t_{\text{q}}}_{\text{queue}} + \underbrace{t_{\text{pf}}}_{\text{prefill}} + \underbrace{t_{\text{th}}}_{\text{thinking + branches}} + \underbrace{t_{\text{vf}}}_{\text{verifier}}. \tag{4}$$

Equation 4 makes operationally explicit that classical TTFT optimizations attack only t_{pf}, while the dominant term for

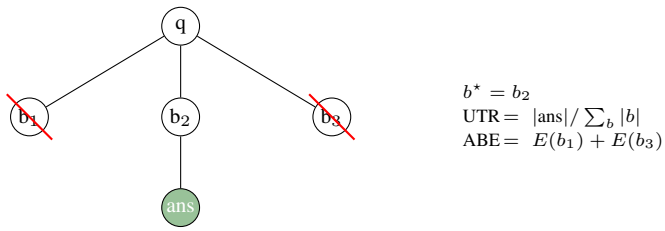

Fig. 3. UTR and ABE applied to a best-of-3 reasoning trace. Crossed branches contribute to the UTR denominator and to ABE but not to the delivered answer.

reasoning workloads is t_{th}. Reporting TTVA as a four-tuple $(t_{\text{q}}, t_{\text{pf}}, t_{\text{th}}, t_{\text{vf}})$ in benchmark output preserves diagnostic information without hiding the end-to-end SLA.

b) Per-request vs. population UTR.: A serving fleet has both per-request and aggregate UTR. Let r index requests in a trace window. Then

$$\overline{\text{UTR}} \;=\; \frac{\sum_r T_{\text{ans}}^{(r)}}{\sum_r T_{\text{gen}}^{(r)}}. \tag{5}$$

Population UTR (Equation 5) is what cloud operators bill against; per-request UTR is what diagnoses pathological queries. The two should be reported together—a fleet can have a healthy population UTR while a long-tail of best-of-N heavy requests dominates ABE.

c) Energy attribution within a branch.: Concretely computing $E(b)$ requires a residency model. A first-order attribution charges a branch for (i) its share of prefill, (ii) the decode energy of its own tokens, and (iii) its KV-cache footprint over its lifetime:

$$E(b) \;\approx\; \alpha\, T_{\text{pf}} + \beta\, T_{\text{dec}}(b) + \gamma \int_0^{\tau(b)} K(b,t)\,\mathrm{d}t, \tag{6}$$

where $K(b,t)$ is the live KV-cache footprint of branch b at time t, $\tau(b)$ is its lifetime, and α, β, γ are hardware-specific coefficients calibratable from rack-level power telemetry. This is the inference analog of the carbon-accounting decomposition in [4], [6]; the KV-residency term explicitly charges branches that hold cache long after their last useful token.

B. Worked Example

To make the metrics concrete, we walk through a synthetic best-of-3 reasoning trace. **All numbers in this subsection are illustrative, not measured.** The point is to show how Equations 1–6 combine on a single request.

a) Setup.: A request arrives at $t_{\text{arr}} = 0\,\text{s}$. The serving stack runs three parallel branches b_1, b_2, b_3 off a shared prefill, then routes the surviving branch through a programmatic verifier (a unit-test runner, regex check, or judge model). Synthetic per-phase durations and token counts:

- Queue: $t_{\text{q}} = 0.05\,\text{s}$.
- Prefill: $t_{\text{pf}} = 0.20\,\text{s}$, producing 1,000 prompt tokens.
- Thinking + branches: each branch generates roughly 1,200 tokens over $t_{\text{th}} = 18.0\,\text{s}$ of wall clock (parallel; the slowest branch dominates). Branch b_2 is selected as $b^\star$ with $T_{\text{ans}} = 200$ tokens of final answer.
- Verifier: $t_{\text{vf}} = 0.30\,\text{s}$.

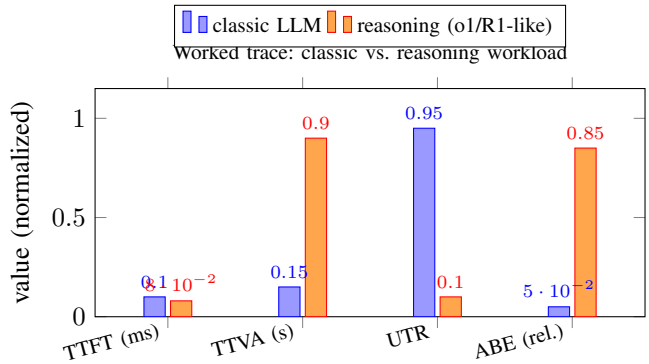

Fig. 4. Worked-trace comparison: a reasoning workload with near-identical TTFT looks dramatically worse on TTVA, UTR, and ABE. The values are normalized from the trace in Section III-B; they are an accounting example, not a claim about a deployed model.

b) TTVA.: By Equation 4, $\text{TTVA} = 0.05 + 0.20 + 18.0 + 0.30 = 18.55\,\text{s}$, with the thinking phase contributing $> 97\%$. Reporting only TTFT ($\approx 0.25\,\text{s}$) would suggest the request was effectively instant, which is the failure mode we are arguing against.

c) UTR.: The total generated-token count includes the answer plus the discarded portions of the two pruned branches plus the non-answer portion of $b^\star$:

$$T_{\text{gen}} \;=\; 3 \times 1200 \;=\; 3600, \qquad T_{\text{ans}} \;=\; 200,$$

giving $\text{UTR} = 200/3600 \approx 0.056$. The same engine running a non-reasoning chat completion on a comparable prompt would post a UTR near 1, which is precisely the contrast Figure 4 illustrates qualitatively.

d) ABE.: If each branch consumes a roughly equal energy share $E(b_i) \approx E_0$ (the simplest first-order attribution from Equation 6 with equal T_{dec} and similar KV-residency), then

$$\text{ABE} \;=\; E(b_1) + E(b_3) \;\approx\; 2E_0, \qquad E(b^\star) \;\approx\; E_0,$$

so two-thirds of the request's accelerator energy was spent on tokens the user never received. A pricing or carbon disclosure that does not separate $E(b^\star)$ from ABE makes this invisible to both the tenant and the operator.

e) Takeaway.: Even on a small best-of-3 trace, the reasoning-aware metrics tell a story (97% of latency is hidden, 94% of generated tokens are wasted, two-thirds of energy is on rejected branches) that TTFT and TBT cannot tell at all.

IV. Implications for System Design

Table III summarizes the design surface affected by adopting TTVA/UTR/ABE. The remainder of this section expands the most load-bearing implications.

TABLE III
SYSTEM-DESIGN IMPLICATIONS OF REASONING-AWARE SLOS ACROSS SERVING-STACK SUBSYSTEMS. QUALITATIVE; NOT A BENCHMARKING CLAIM.

Subsystem	Implication under TTVA / UTR / ABE
Scheduler	Shift from prefill-deadline-aware to thought-budget-aware; track per-request token and energy budgets, not just queue depth.
KV-cache mgmt.	Eviction must preserve verifier-relevant prefix tokens through long thinking phases; cache cost charged into $E(b)$ via the residency term in Eq. 6.
Admission control	Reject or downgrade incoming requests when projected $\sum_b E(b)$ would exceed a fleet-level ABE budget; expose a degraded-quality knob (e.g., best-of-1) instead of OOM.
Disaggregation	Re-cut along thought vs. answer instead of prefill vs. decode; different SKUs can serve latency-tolerant thinking and latency-critical answer commits.
Pricing	Three-tier billing: answer tokens at full rate, thought tokens discounted, ABE surcharged to the requester rather than amortized.
Carbon disclosure	Report $E(b^\star)$ and ABE separately; aggregate-FLOP disclosures structurally hide best-of-N overhead.

Algorithm 1: Budget-aware admission for reasoning requests

Input: request r; per-request budgets B_T (tokens), B_E (energy); fleet-level remaining ABE budget $\hat{A}$; verifier V.
Output: accept / degrade / reject decision.

1. $\hat{T}, \hat{E} \leftarrow$ projected T_{gen}, $\sum_b E(b)$ from request class and historical UTR / ABE distribution.
2. **if** $\hat{E} > \hat{A}$ **then** return REJECT.
3. **if** $\hat{T} > B_T$ **or** $\hat{E} > B_E$ **then**
   downgrade r: cap $|\mathcal{B}|$ to 1 (no best-of-N),
   cap thinking tokens, return ACCEPT-DEGRADED.
4. admit r with branch quota $|\mathcal{B}| = N(r)$ and token budget $B_T(r)$.
5. **while** request is running:
   track live $T_{\text{gen}}^{(r)}$, $\sum_b E(b)$;
   if either exceeds budget, force commit on best so-far branch.
6. run V; update fleet $\hat{A} -= \mathrm{ABE}(r)$;
   record $(\mathrm{TTVA}, \mathrm{UTR}, \mathrm{ABE})$ for this request.

Fig. 5. Budget-aware admission control. Steps 2–3 enforce per-fleet and per-request ABE / token budgets; step 5 prevents a single best-of-N request from dominating ABE; step 6 closes the loop with verifier outcome and metric reporting.

a) Schedulers: deadline-aware vs. budget-aware.: A TTFT-driven scheduler is essentially deadline-aware over prefill. A TTVA-driven scheduler must instead reason about a *thought budget*: how many tokens / how much energy a request is allowed to spend before it must commit. This argues for budget-aware admission control on top of existing continuous-batching engines [2], [9], with explicit speculative-branch quotas rather than implicit token quotas. Figure 5 sketches the control loop in pseudocode.

b) Routing and disaggregation.: DistServe-style prefill/decode disaggregation [10] splits a request along a TTFT/TBT boundary. Reasoning workloads suggest a new boundary: *thought* vs. *answer*. Thought-phase tokens are latency-tolerant (the user is not watching) but throughput- and energy-sensitive; answer-phase tokens are latency-critical. A "think-cluster / answer-cluster" split, with different hardware balance points, is the natural reasoning-era disaggregation pattern.

c) Cost allocation.: Today's serving stacks bill per input/output token. Under UTR/ABE, the honest bill has three line items: answer tokens, thought tokens, and abandoned-branch tokens—each potentially priced differently to align incentives between provider and tenant. A concrete pricing implication: charging full rate for $b^\star$ tokens, a discount for thought tokens (latency-tolerant, can be batched aggressively), and an explicit carbon surcharge on ABE makes the cost of best-of-N visible to the requester rather than amortized into a single token price. This is the inference analog of the energy-accounting argument in [4].

V. THREATS TO VALIDITY AND LIMITATIONS

We are deliberately stating new SLOs without yet measuring them at scale. We list threats to validity in decreasing order of how much they could weaken the position.

a) T1: TTVA presupposes a verifier.: TTVA is well-defined when an `accept` signal exists: a unit-test runner, a regex/JSON-schema check, a math grader, or a judge model. For open-ended reasoning tasks (creative writing, open Q&A), there is no clean verifier boundary, and TTVA must be approximated by a proxy (e.g., end-of-final-message). UTR and ABE remain well-defined regardless. We do not claim TTVA is the right SLO when the task itself has no verifiable success criterion.

b) T2: UTR's denominator depends on workload mix.: A fleet's population UTR (Equation 5) is a function of how many reasoning vs. non-reasoning requests it sees. A 0.5 UTR is excellent for a math-tutor product and alarming for a chat assistant. UTR must be reported per workload class, not as a single fleet number, or it becomes gameable (e.g., by routing easy traffic through the same engine to inflate the denominator).

c) T3: ABE requires per-branch energy attribution.: Today's GPU telemetry reports rack- or device-level power, not per-branch energy. Equation 6 is a first-order attribution; in production, α, β, γ must be calibrated against measured rack power and will drift with hardware generation, batch shape, and KV-cache pressure. Initial deployments should treat ABE as a relative ranking metric across requests on the same fleet, not an absolute kWh figure.

d) T4: Metrics interact and can be gamed jointly.: A system can trivially improve UTR by killing branches early, at the cost of TTVA and accuracy; it can improve TTVA by skipping the verifier; it can improve ABE by refusing best-of-N. The metrics must be reported as a joint $(\mathrm{TTVA}, \mathrm{UTR}, \mathrm{ABE}, \mathrm{accuracy})$ tuple, ideally Pareto-style across operating points, before any single number is treated as a leaderboard column.

e) T5: Position-paper status.: This paper proposes definitions and sketches mechanisms; the illustrative figure (Figure 4) and the worked example (Section III-B) are explicitly synthetic. An empirical instantiation on an open-weights R1-class [1] model under mixed traces is the natural follow-on and is out of scope for a position paper.

VI. Related Work

a) Test-time compute and reasoning models.: Test-time scaling laws [5] establish that compute spent at inference can substitute for parameters; o1 [3] and DeepSeek-R1 [1] are the operational manifestation, and they are precisely the workloads that motivate this paper.

b) Existing serving SLO frameworks.: vLLM's PagedAttention [2] and SGLang [9] are evaluated against TTFT, TBT, and request-throughput; they do not distinguish answer tokens from thought tokens or charge for branch fan-out. DistServe [10] disaggregates prefill from decode to optimize *goodput* under TTFT/TBT, which is a strictly finer slicing of the same metric family. Our position is orthogonal: the metric family itself, not just its scheduling, needs to change for reasoning workloads. Reasoning-aware SLOs would compose with all three engines: TTVA budgets become per-request deadlines that the continuous batcher honors, UTR tags throughput accounting, and ABE attaches to the energy attribution already collected for capacity planning.

c) KV-cache management.: Cache compaction and eviction work [7], [8] optimizes the substrate that reasoning traces stress hardest. Equation 6's KV-residency term is the place where these techniques would interact with ABE: more aggressive eviction reduces $\gamma \int K(b,t)\,\mathrm{d}t$ but risks evicting verifier-relevant prefix tokens, hurting TTVA via t_{vf} in Equation 4.

d) Sustainability accounting.: Carbon accounting in ML [4], [6] motivates ABE's framing: aggregate-FLOP metrics conceal work that does not ship. ABE is the inference-side analog of separating training-phase from deployment-phase carbon: it isolates the fraction of inference energy that funded discarded branches.

e) Position relative to prior work.: To our knowledge, no prior position paper argues that the SLO definitions themselves—not just the engines that hit them—need to change for reasoning models. The closest adjacent work is the goodput reframing of [10], which keeps TTFT/TBT and adjusts what counts toward useful capacity; this paper goes further and proposes that what counts toward *user-observed quality* must also change.

VII. Conclusion and Call to Action

TTFT and TBT served the streaming-chatbot era well. The reasoning era is different in kind, not degree: hidden chains of thought, best-of-N search, and verifier loops mean that tokens are no longer fungible and "first token" is no longer meaningful. We have proposed three metrics —TTVA, UTR, ABE—that recover what TTFT/TBT lose, formalized them with a TTVA decomposition, a population-UTR aggregator, and an energy-residency attribution model, walked them through a synthetic best-of-3 trace, and sketched a budget-aware admission-control loop. We urge the serving-systems community to (i) report TTVA alongside TTFT in benchmark results, (ii) include UTR in throughput leaderboards, (iii) add ABE to inference carbon disclosures, and (iv) report the four-tuple jointly to prevent the gaming patterns enumerated in T4. ShareGPT-style trace replay is no longer a sufficient workload; reasoning-aware traces with branch- and verifier-level annotations are needed before MLArchSys-class systems can be honestly compared.

References

[1] DeepSeek-AI, "DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[2] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with PagedAttention," in *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023, arXiv:2309.06180.

[3] OpenAI, "Learning to reason with LLMs," OpenAI technical report and system card, 2024.

[4] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," *arXiv preprint arXiv:2104.10350*, 2021.

[5] C. Snell, J. Lee, K. Xu, and A. Kumar, "Scaling LLM test-time compute optimally can be more effective than scaling model parameters," *arXiv preprint arXiv:2408.03314*, 2024.

[6] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. A. Behram, J. Huang, C. Bai, M. Gschwind, A. Gupta, M. Ott, A. Melnikov, S. Candido, D. Brooks, G. Chauhan, B. Lee, H.-H. S. Lee, B. Akyildiz, M. Balandat, J. Spisak, R. Jain, M. Rabbat, and K. Hazelwood, "Sustainable AI: Environmental implications, challenges and opportunities," *Proceedings of Machine Learning and Systems*, vol. 4, 2022, arXiv:2111.00364.

[7] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis, "Efficient streaming language models with attention sinks," *arXiv preprint arXiv:2309.17453*, 2023.

[8] Z. Zhang, Y. Sheng, T. Zhou, T. Chen, L. Zheng, R. Cai, Z. Song, Y. Tian, C. Re, C. Barrett, Z. Wang, and B. Chen, "H2O: Heavy-hitter oracle for efficient generative inference of large language models," *arXiv preprint arXiv:2306.14048*, 2023.

[9] L. Zheng, L. Yin, Z. Xie, C. Sun, J. Huang, C. H. Yu, S. Cao, C. Kozyrakis, I. Stoica, J. E. Gonzalez, C. Barrett, and Y. Sheng, "SGLang: Efficient execution of structured language model programs," arXiv preprint arXiv:2312.07104, 2024.

[10] Y. Zhong, S. Liu, J. Chen, J. Hu, Y. Zhu, X. Liu, X. Jin, and H. Zhang, "DistServe: Disaggregating prefill and decoding for goodput-optimized large language model serving," in *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, 2024, arXiv:2401.09670.