Article

A Time-Resolved, SLO-Aware and Bi-Objective Framework to Measure and Minimize LLM Serving's Carbon and Water Footprints

Julian Hoxha [*] , Marsela Thanasi-Boçe , Tarek Khalifa

Posted Date: 13 October 2025

doi: 10.20944/preprints202510.0957.v1

Keywords: LLM inference; carbon-aware routing; water-aware routing; geo-distributed datacenters; SLO; TTFT; TPOT; MILP; CIF; EWIF; PUE; WUE

doi:10.20944/preprints202510.0957.v1

Article

A Time-Resolved, SLO-Aware and Bi-Objective Framework to Measure and Minimize LLM Serving's Carbon and Water Footprints

Julian Hoxha [1,*], Marsela Thanasi-Boce [2] and Tarek Khalifa [1]

[1] College of Engineering and Technology, American University of the Middle East, Kuwait
[2] College of Business Administration, American University of the Middle East,Kuwait
* Correspondence: julian.hoxha@aum.edu.kw

Abstract

Studies of the environmental footprint of large language model (LLM) inference often disagree because they mix incompatible system boundaries, ignore latency and throughput service level objectives (SLOs), and optimize carbon without accounting for water. We present a provider-agnostic framework that unifies scope-transparent measurement with time-resolved, bi-objective orchestration under realistic SLOs. Measurement follows production practice and reports daily medians at a comprehensive serving boundary that includes active accelerators, host CPU/DRAM, provisioned idle, and facility overhead via PUE. Consumptive water is computed as site plus source. Carbon is location-based (LB) by default with a market-based (MB) sensitivity. Optimization is cast as a mixed-integer linear program, solved over 288 five-minute windows per day. For each prompt profile, the solver selects region, batch size, and phase-aware hardware for prefill and decode while enforcing $p95$ Time To First Token/Time Per Output Token (TTFT/TPOT) and capacity constraints. Because grid carbon intensity (CIF) and electricity water intensity (EWIF) are only weakly correlated, the policy is dual-objective by design and balances carbon and water explicitly. Applied to four representative models using public per-prompt energy tables and per-region multipliers, a single SLO-aware policy reduces comprehensive-boundary medians by 57–59% for energy, 59–60% for consumptive water, and 78–80% for LB CO_2, with SLOs met in every window. For a day with 500 M queries on GPT-4o, median-scaled totals drop from 0.344→0.145 GWh, 1.196→0.490 ML, and 121→25 tCO_2 (LB). The framework also reproduces the production-observed accelerator-only versus comprehensive gap (narrow/comprehensive ≈ 0.417), enabling direct translation across studies. Pareto analyses show when routing alone and when joint routing, batching, and token-length controls deliver concurrent reductions in carbon and water at fixed quality of service. The combination of time-resolved control, comprehensive accounting, and dual-objective optimization yields a deployable template for decarbonization and water stewardship in LLM serving.

Keywords: LLM inference; carbon-aware routing; water-aware routing; geo-distributed datacenters; SLO; TTFT; TPOT; MILP; CIF; EWIF; PUE; WUE

1. Introduction

The deployment of large language models on scale has reshaped the sustainability discussion for artificial intelligence [1,2]. Early work focused almost exclusively on the training phase because one-off runs for frontier models were shown to consume on the order of thousands of megawatt-hours, emit hundreds of tons of CO_2 equivalent, and use large volumes of cooling water for a single training job [3–5]. Those numbers rightly drew attention and motivated research on efficient architectures and carbon-aware training [6,7]. The landscape has since changed. Public adoption has turned inference into a continuous and interactive service that runs every minute of the day. The cumulative footprint of serving billions of queries now dominates the life-cycle of many deployments and can exceed training

doi:10.20944/preprints202510.0957.v1

by a wide margin [5,8]. Several analyzes estimate that inference can account for up to 90% of the lifetime energy and that the annual operating energy of the large-scale service can be tens of times higher than the energy used to train the model [5,9,10]. A single short prompt appears small when measured in watt-hours, yet at hundreds of millions of prompts per day, the aggregate demand reaches utility-scale electricity and meaningful volumes of water [5]. It follows that sustainability efforts that target training alone are no longer sufficient.

Accurate accounting for inference is challenging, and the literature has only recently converged on methods that match production practice [11]. Early estimates combined accelerator nameplate power, theoretical FLOPs, and assumed token lengths, producing per-prompt energies that differed by an order of magnitude because each study chose different utilization factors and boundaries [5,12,13]. Empirical instrumentation is now emerging through industry studies [1]. A key result from this line of work is that narrow measurement—accelerator only—misses material contributors such as host CPU and DRAM, provisioned idle, and facility overhead captured by power usage effectiveness. Google's recent production study reports medians at a comprehensive serving boundary and shows that accelerator-only accounting can underestimate per-prompt energy by more than a factor of two for the same workload [1]. This boundary choice explains much of the spread in earlier per-prompt claims and makes clear that scope-transparent reporting is a prerequisite for credible comparison and for effective optimization.

Water has been even less visible in model reporting despite its central role in data center operations [3,5,14]. Cooling systems withdraw and consume water on site, and electricity generation for computing carries a significant upstream water intensity [15]. Model cards and environmental summaries often report scope 2 carbon while omitting water entirely [16,17]. The omission matters because the water intensity of electricity and the carbon intensity of the grid vary by region and time, and they are only weakly correlated. A policy that pursues the lowest carbon intensity without regard to water can inadvertently increase total water consumption, for example by shifting the load to nuclear-heavy or hydro-dominated regions where the liters per kilowatt-hour are higher, while the converse can also occur [3,14]. Recent studies document that carbon and water can move in opposite directions, which means sustainability must be treated as a bi-objective problem rather than a single number to be minimized [14].

Any practical solution must also respect the realities of an interactive service [18]. LLM serving is governed by strict service level objectives for responsiveness. Time to first token and time per output token directly shape perceived latency and throughput [17]. Production systems cannot delay user requests to wait for a greener grid, nor can they indiscriminately route traffic to distant regions if that violates latency budgets. The optimization space is therefore bounded by performance constraints and by the capacity of the serving fleet. A sustainability framework that ignores these constraints does not translate into production.

This paper proposes a deployment-aware framework that unifies measurement and control. The approach is provider-agnostic, scope transparent by construction, and explicitly bi-objective in carbon and water. Measurement follows production practice and reports daily medians at a comprehensive serving boundary that includes active accelerators, host CPU and DRAM, provisioned idle, and PUE uplift [1]. Consumptive water is computed as the site plus the source using $W = \mathrm{PUE} \cdot \mathrm{WUE}_{\mathrm{site}} + \mathrm{EWIF}$ [3]. Carbon is location based by default with a market-based sensitivity when portfolio factors are disclosed. Optimization casts serving orchestration as a mixed-integer linear program solved over 288 five-minute windows per day. For each prompt profile the solver chooses region, batch size, and a phase-aware hardware assignment that separates prefill and decode. The formulation enforces $p95$ constraints on time to first token and time per output token, augmented by a per-region network term. Capacity constraints derive from measured tokens-per-second tables. Token-length directives are modeled as controllable reductions in decode work.

Our contributions are fourfold. First, we provide a full-stack, scope-transparent accounting for inference that reproduces the production gap between accelerator-only and comprehensive boundaries and supplies a defensible translation between the two, with a narrow-to-comprehensive median ratio of about 0.417 for the workloads we study [1]. Second, we formulate a time-resolved, bi-objective optimizer that co-minimizes location-based carbon and consumptive water under explicit $p95$ SLOs and fleet capacity. Third, we demonstrate the framework on four representative models using public per-prompt energy tables and provider PUE, WUE, and carbon factors, and we show that a single SLO-aware policy reduces comprehensive medians by roughly 57–59% for energy, 59–60% for water, and 78–80% for location-based CO_2 while meeting latency and throughput targets in every window. Fourth, we analyze routing-only and joint routing+batch+token frontiers to make the carbon–water geometry explicit, which tells operators when improvements move both metrics together and when trade-offs must be managed.

The remainder of the paper is organized as follows. Section 2 surveys the literature and motivates comprehensive boundaries, water-aware metrics, and SLO-respecting orchestration. Section 3 details the methodology, including the functional unit, scope definitions, impact accounting, time resolution, decision variables, constraints, and parameterization. Section 4 presents the results, reporting comprehensive-boundary medians and daily totals, scope reconciliation between accelerator-only and comprehensive views, and carbon–water movement at fixed quality of service with routing, batch, token, and phase-split analyses. Section 5 discusses implications, deployment guidance, and limitations, and outlines directions for future work. Section 6 concludes the paper.

2. Optimizing the Environmental Footprint of LLM Inference: A Literature Review

2.1. From Training to Inference: Why the Burden Has Shifted

Early work on AI sustainability emphasized training, where single jobs for frontier models consumed thousands of MWh and emitted hundreds of tons of CO_2e while using significant cooling water [3–7]. As generative systems moved into daily use, the continuous nature of serving billions of prompts has become the dominant lifecycle driver for many deployments [5,8–10,19–21]. Per-request impacts that seem small in isolation compound at scale: a single ChatGPT-style prompt has been estimated at several grams CO_2e, far above a web search [22], and daily volumes in the hundreds of millions translate to utility-scale electricity and meaningful water withdrawals [5]. These observations motivate methods that measure inference accurately and optimize it under interactive quality constraints.

2.2. Full Stack per Prompt Accounting: Energy, Carbon, Water, and Embodied Impacts

Production studies recommend a comprehensive serving boundary that includes the active accelerator, host CPU and DRAM, provisioned idle, and facility overhead via PUE[1,23,24]. Under this boundary, “accelerator only” views can undercount by more than a factor of two. The Google study reports a narrow-to-comprehensive median ratio near 0.417 for similar workloads [1]. Carbon per prompt should be reported both location based (grid average) and, when portfolios are disclosed, market based [25], and it should include amortized embodied impacts from hardware manufacturing where possible [26–28]. Water deserves first-class treatment: consumptive site cooling scaled by PUE plus off-site electricity–water intensity (EWIF) yields a scope-consistent measure of liters per kWh that varies widely by region and generation mix [3,29–33]. Because carbon intensity (CIF) and EWIF are only weakly correlated, any realistic framework should measure both and co-optimize them [3,14].

2.3. Measurement Boundaries: Why Scope Transparency Matters

Inconsistent boundaries explain much of the order-of-magnitude spread in earlier per-prompt estimates. Studies that measured only chip power (sometimes at unrealistic utilization) omitted host, idle, and cooling overheads and thus overstated efficiency [5,34,35]. Converging practice now recommends a comprehensive inference boundary under operator control, with explicit exclusions outside that boundary (e.g., end user devices or wide-area network transit) and with scope-transparent reporting so results are comparable across systems [1,24]. The ≈0.417 narrow-to-comprehensive ratio provides a defensible translation when only accelerator-level numbers are available and site factors and mixes are held constant [1]. Our methods section adopts this convention and reports daily medians to handle skewed mixes.

2.4. Real-Time Orchestration: Carbon- and Water-Aware Routing Under SLOs

With consistent measurement in place, the next step is optimization. Carbon-aware request routing has been shown to cut serving emissions substantially without violating latency when traffic is steered to cleaner grids in space and time [10,36–39]. Projections suggest even larger reductions as grids decarbonize and capacity headroom grows [10]. Emerging frameworks treat scheduling as a multi-objective problem, co-optimizing carbon, water, and cost while enforcing latency SLOs and site capacity constraints; practical solutions range from MILPs to learning-based controllers [9,40–43]. Our

2.5. Phase-Aware Hardware Scheduling (Prefill vs. Decode)

Transformer serving has two phases with different bottlenecks. Prefill is parallel and compute-heavy. Decode is sequential and often memory-bound [44,45]. Phase-aware scheduling assigns prefill and decode to different hardware or configurations to improve both responsiveness and efficiency [1,46–48]. Speculative decoding and KV-cache reuse further reduce decode work [49,50]. Our formulation exposes prefill and decode decisions and allows second-life hardware to serve decode when SLOs permit [22,51].

2.6. Semantic-Level Interventions

Beyond systems levers, generation can be made more efficient with semantic directives that reduce unnecessary output length or avoid high-compute behaviors while preserving usefulness. SPROUT exemplifies this approach, reporting large carbon savings with minimal impact on quality by selectively applying concise directives [52]. Related work on eco-adaptive services explores small, user-acceptable adjustments during high-carbon periods to achieve large operational reductions [21]. These methods complement, rather than replace, hardware and orchestration optimizations.

2.7. Lifecycle and Circular Economy Strategies

Holistic sustainability includes upstream manufacturing and end of life. Hardware production for AI accelerators is energy- and material-intensive; embodied emissions can dominate in low-carbon operational settings [53,54]. Scenario analyses warn of rapid growth in AI-related e-waste, and point to device life extension, refurbishment, parts harvesting, and improved recycling as the most effective mitigations [55]. Second-life deployment and modular upgrades align with extended producer responsibility and reduce both embodied and disposal impacts [1,22,55].

2.8. Toward a Unified, Deployment-Aware Framework

Across the literature a consensus is emerging: inference now dominates impacts [5,8–10,19–21]; scope-transparent, full-stack per-prompt accounting is essential [1,23–25]; optimization must be bi-objective in carbon and water and must observe latency SLOs [3,9,10,14,36–39]; and lifecycle thinking is required to avoid shifting costs upstream or downstream [22,53–55]. Persistent gaps remain—especially inconsistent boundaries and limited treatment of water. The present work addresses these gaps by (i) adopting comprehensive, production-style measurement with explicit narrow↔comprehensive translation; (ii) integrating EWIF alongside CIF so water is optimized jointly with carbon; (iii) enforcing $p95$ TTFT/TPOT and capacity constraints so results are deployable; and (iv) exposing phase-aware and semantic levers that reduce watt-hours per prompt while maintaining interactive quality. Together, these elements provide a practical path for operators to measure, compare, and materially lower the environmental footprint of LLM serving.

3. Methodology

The aim of the methodology is to turn non-comparable measurements of LLM inference into one portable, auditable workflow that: (i) reports apples-to-apples per-prompt medians under a comprehensive serving boundary consistent with production practice and, when needed for comparison, under a narrower accelerator-only boundary; (ii) enforces real SLOs through explicit capacity and latency constraints derived from public tokens-per-second and latency quantiles; and (iii) co-optimizes energy, location-based (LB) greenhouse-gas emissions, consumptive water, and amortized embodied impacts through a mixed-integer linear program (MILP). The comprehensive boundary follows the decomposition implemented in a recent first-party production study for Gemini Apps, which measures the median text prompt at 0.24 Wh (comprehensive) with an accelerator-only median of$\approx$ 0.10 Wh when active accelerator, host CPU/DRAM, provisioned idle, and facility overhead (PUE) are aggregated [1]. These medians imply an uplift of $\approx 2.4\times$ due purely to boundary choice (comprehensive vs. accelerator-only). We adopt that full-stack paradigm for comparability and to avoid systematic undercounting identified by the production study [1].

3.1. Functional Unit and System Boundaries

The functional unit is one served prompt. We compute energy per prompt in watt-hours at two nested system boundaries so that our results are comparable to both “narrow” and “comprehensive” studies [1]. Accelerator-only boundary counts only the active accelerator energy directly attributable to the prompt. Comprehensive serving boundary adds host CPU/DRAM, provisioned idle capacity, and facility overhead captured by PUE. We adopt this boundary as our default because it aligns with production telemetry and produces apples-to-apples medians across models and regions [1].

Let E^{active} denote accelerator-only energy per prompt (i.e., the GPU/TPU compute energy for the prefill and decode phases). It excludes host CPU/DRAM energy, provisioned-but-idle energy, and any facility overhead captured by PUE; E^{IT} the IT energy per prompt (accelerators + host CPU/DRAM + provisioned idle); E^{host} and E^{idle} the per-prompt energy used on non-accelerator hardware that is part of serving the request: host CPUs, DRAM, NICs, disks on the inference nodes (and sometimes service sidecars); $\kappa_{\text{host+idle}}$ the ratio of total IT-side energy (accelerator active + host + idle) to the accelerator's active energy:

$$\kappa_{\text{host+idle}} = \frac{E^{\text{IT}}}{E^{\text{active}}} = \frac{E^{\text{active}} + E^{\text{host}} + E^{\text{idle}}}{E^{\text{active}}}. \tag{1}$$

To be numerically consistent with the reported medians ($E^{\text{fac}} \approx 0.24$ Wh and $E^{\text{active}} \approx 0.10$ Wh [1]), we set $E^{\text{IT}} = E^{\text{fac}}/\text{PUE} \approx 0.24/1.09 \approx 0.220$ Wh and thus obtain $\kappa_{\text{host+idle}} \approx E^{\text{IT}}/E^{\text{active}} \approx 0.220/0.10 \approx 2.20$. When host/idle shares are undisclosed, we use this aggregated $\kappa_{\text{host+idle}}$ rather than point estimates for E^{host} and E^{idle}.

We define: $E^{\text{fac}} = \text{PUE} \times E^{\text{IT}}$ the facility energy per prompt (what the whole data center expends to deliver the prompt), with PUE$\approx$1.09 we have $E^{\text{fac}} \approx 0.24$ Wh per text prompt. This matches the comprehensive median found in [1] and implies a $\sim 2.4\times$ uplift over the accelerator-only median ($\sim$0.10 Wh) for the same workload.The overhead energy is exactly the extra above IT:

$$E^{\text{overhead}} = (\text{PUE} - 1) \times E^{\text{IT}} \approx 0.02 \text{ Wh}.$$

3.2. Impact Accounting

We compute operational emissions using the location-based (LB) approach recommended for data-center Scope-2 reporting [1]. The facility energy associated with serving one prompt, E^{fac} (kWh prompt^{-1}), is multiplied with location-based grid carbon intensity for region r, CIF_r^{LB} (kg CO_2kWh^{-1}). This yields the LB, per-prompt operational emissions:

$$g\text{CO}_2^{\text{LB}} = E^{\text{fac}} \times \text{CIF}_r^{\text{LB}} \times 10^3. \tag{2}$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 October 2025 doi:10.20944/preprints202510.0957.v1

LB attributes emissions to the electricity actually drawn from the local grid at the point of consumption and is the basis for statistically comparable across operators and regions, aligning with routing-aware decisions, and avoids year-to-year swings caused by portfolio accounting. For the sensitivity we also compute the provider's prior-year portfolio emission factor EF_r^{MB} ($\mathrm{kg\,CO_2e\,kWh^{-1}}$) [25]. MB, when disclosed, can be reported as a sensitivity and makes results operationally interpretable while retaining comparability. We use LB as the objective default and MB as a sensitivity. Operational emissions are reported in *g* CO_2 *(LB)*; *g* CO_2e is reserved for MB factors and embodied/lifecycle terms.

For water, we follow the scope-consistent "site + source" rule advocated in the AI water-footprint literature [3]. Consumptive site water (cooling, at the facility) is proportional to facility energy via the site Water Usage Effectiveness $\mathrm{WUE}_r^{\mathrm{site}}$ ($\mathrm{L\,kWh^{-1}}$) while source water accounts for the electricity-generation water intensity of the regional grid via the Electricity-Water Intensity Factor $\mathrm{EWIF}_r^{\mathrm{source}}$ ($\mathrm{L\,kWh^{-1}}$), separating site water (cooling; WUE Category 2) from source water associated with electricity generation (EWIF), as recommended in the water-footprint literature [3]. Using $E^{\mathrm{IT}}[kWh/prompt]$ for the electrical work of the IT load, the per-prompt consumptive water is:

$$mL_r = [E^{\mathrm{IT}} \times \left(\mathrm{PUE}_r \times \mathrm{WUE}_r^{\mathrm{site}}\right) + E^{\mathrm{IT}} \times \mathrm{EWIF}_r^{\mathrm{source}}] \times 10^3. \tag{3}$$

This form makes explicit that site water scales with the facility energy (hence the PUE multiplier), while source water scales with the electricity used to power the IT equipment and is governed by the grid mix. Analysis emphasizes both the need to include scope-2 water and the empirical weak correlation between carbon and water intensities [3], motivating dual-metric reporting and optimization. The macro background on facility-level PUE and WUE levels across the U.S. fleet is taken from the 2024 U.S. Data Center Energy Usage Report, which shows hyperscale-weighted averages below ∼1.4 PUE and site WUE around 0.32–0.40 $\mathrm{L\,kWh^{-1}}$ in 2023 (with a likely rise as liquid cooling penetrates), giving realistic bounds for sensitivity tests [56].

We summarize per-prompt footprints (Wh, mL, $\mathrm{gCO_2e}$) as daily medians by profile and season, mirroring the production study's choice of medians as the right statistic for skewed mixes; that study reports comprehensive-boundary medians near 0.24 Wh, 0.03 $\mathrm{gCO_2e}$, and 0.26 mL for the median text prompt and documents the $\sim 2.4\times$ uplift from accelerator-only to comprehensive accounting. Our framework reproduces this uplift when we apply $\kappa_{\mathrm{host+idle}}$ and PUE to chip-only numbers, providing scope-transparent comparability [1].

We include embodied impacts and e-waste on a serving-time basis. For accelerator class h with embodied carbon C_h^{emb} ($\mathrm{kg\,CO_2e\,board^{-1}}$), board mass M_h^{board} (g), and assumed service lifetime L_h (days), we compute daily rates C_h^{emb}/L_h ($\mathrm{kg\,CO_2e\,day^{-1}}$) and M_h^{board}/L_h ($\mathrm{g\,day^{-1}}$). When a class is activated (binary variable in the solver), we allocate its daily embodied rates to the prompts it serves that day, adding an amortized per-prompt embodied term that brings hardware lifecycle into scope. This aligns with the lifecycle scope of the production study [1,22].

3.3. Time Resolution, Traffic Mix and SLOs

We partition a day into $|T|$=288 decision windows of five minutes each with budget Δt=300 s. Demand is described by three prompt profiles $p \in P$=$\{short, medium, long\}$ with token pairs $(T_p^{\mathrm{in}}, T_p^{\mathrm{out}})$ equal to $(100, 300)$, $(1000, 1000)$, and $(10{,}000, 15{,}000)$. Unless noted, arrivals total N_{day}=500 M prompts per day with a 70/25/5% mix. Windowed arrivals $Q_{t,p}$ are either provided or constructed from N_{day}, the mix π_p, and diurnal weights $f(t)$ as $Q_{t,p}$=round$\left(N_{\mathrm{day}}\pi_p f(t)/\sum_{t'} f(t')\right)$.

The router chooses non-negative integer assignments $x_{t,r,h,b,\varphi,p,d}$ by region r, hardware class h, batch size b, phase $\varphi \in \{prefill, decode\}$, and directive $d \in \{default, brief\}$. Prefill and decode are two legs of the same request, so conservation holds per window and profile as in Eq. (6). Capacity in a five-minute window is enforced with quantile throughputs $\mathrm{TPS}_{h,b}(q)$, which bound the total tokens processed per provisioned node at batch b; Eq. (7) applies this limit with the window budget Δt.

Latency SLOs use profile-specific $p95$ targets. We work with the published per-hardware, per-batch quantiles for TTFT and TPOT and the associated TPS rows. The formulation imposes the tail-latency guard of Eq. (8) at p=95 and uses the same p95 throughputs in the capacity constraint so feasibility reflects the stochastic nature of serving. Unless otherwise stated, reported summaries are daily medians by profile and a 70/25/5 mix of those medians, which is consistent with production practice for skewed mixes and utilization patterns [1].

3.4. Decision Variables, objEctive and Constraints

The decision variables are the assignment of prompts $x_{t,r,h,b,\varphi,p,d}$ in each window (non-negative integers), binary activation variables $u_{t,r,h,b} \in \{0,1\}$ for hardware classes at a site, the chosen batch size b from a small discrete set per hardware, binary indicators $z_{r,h} \in \{0,1\}$ that switch on embodied amortization when a hardware class is used *and* $n_{t,r,h,b} \in \mathbb{Z}_{\geq 0}$: the number of replicas provisioned at (r,h,b) in window t. Parameters include the per-site replica cap $U_{r,h,b} \in \mathbb{Z}_{\geq 0}$ and the big-M constants $M_{h,b,\varphi,p,d}$ and $M'_{h,b,p,d}$ used in activation/feasibility linkages.

Each prompt of profile p consists of T_p^{in} input tokens in the prefill phase and T_p^{out} output tokens in the decode phase. Decode tokens are scaled by the directive factor α_d, which can take values such as 0.70, 0.75, or 0.80 depending on whether the output is shortened. The number of tokens served by a routed block is therefore given by:

$$\tau_{r,h,b,\varphi,p,d} = \begin{cases} T_p^{\text{in}}, & \varphi = \textit{prefill}, \\ \alpha_d \, T_p^{\text{out}}, & \varphi = \textit{decode}. \end{cases} \tag{4}$$

Throughput is modeled as $\text{TPS}_{h,b}(q)$, the number of tokens per second at quantile q for hardware class h under batch size b. Latency SLOs are profile-specific constants $L_p^{\star}$, for example 0.9 s for "short" prompts.

We aggregate to daily medians per profile and mix by 70/25/5. The objective is a weighted-sum scalarization with non-negative policy weights:

$$\min_{x,u,z,n} \sum_{t,r,h,b,\varphi,p,d} \left(\alpha \, E^{\text{IT}}_{t,r,h,b,\varphi,p,d} + \beta \, mL_{t,r,h,b,\varphi,p,d} + \lambda \, \text{gCO}_2{}^{\text{LB}}_{t,r,h,b,\varphi,p,d}\right) + \delta \, \text{CO}_2{}^{\text{emb}}_{\text{day}} + \varepsilon \, W^{\text{ewaste}}_{\text{day}}, \tag{5}$$

where minimization is over the decision variables $x_{t,r,h,b,\varphi,p,d}$, $u_{t,r,h,b}$, $z_{r,h}$. The weights $\alpha, \beta, \lambda, \delta, \varepsilon \geq 0$ are chosen by the operator/policy. Equation (5) mirrors multi-impact scoring used in recent infrastructure-aware benchmarking (energy, water, carbon) while adding embodied and e-waste terms for life-cycle completeness [5].

Prefill and decode are two legs of the same request; counts must match and meet demand. The first constraint to satisfy this, a linear one, is:

$$\sum_{r,h,b,d} x_{t,r,h,b,\textit{prefill},p,d} = \sum_{r,h,b,d} x_{t,r,h,b,\textit{decode},p,d} = Q_{t,p}. \tag{6}$$

As we cannot push more tokens through a node in five minutes than the node can handle at the selected batch and the specified throughput quantile, we impose the second constraint, capacity using quantile throughput from the benchmark (per hardware/batch):

$$\sum_{p,d} \left(\frac{x_{t,r,h,b,\textit{prefill},p,d} \, T_p^{\text{in}}}{\text{TPS}_{h,b}(q)} + \frac{x_{t,r,h,b,\textit{decode},p,d} \, \alpha_d \, T_p^{\text{out}}}{\text{TPS}_{h,b}(q)} \right) \leq \Delta t \, n_{t,r,h,b} \qquad \forall \, t,r,h,b, \tag{7}$$

where $\Delta t = 300$ s is the five-minute budget and $q = 95$ unless noted. We enforce $0 \leq n_{t,r,h,b} \leq U_{r,h,b} \, u_{t,r,h,b}$ and $z_{r,h} \geq u_{t,r,h,b}$ for all (t,r,h,b), and gate assignments by $x_{t,r,h,b,\phi,p,d} \leq M_{h,b,\phi,p,d} \, u_{t,r,h,b}$ for all (t,r,h,b,ϕ,p,d).

The third constraint is tail latency controlled by bounding the $p95$ response time per profile to its SLO $L_p^{\star}$:

$$\mathrm{TTFT}_{h,b}(95) + \alpha_d\, T_p^{\mathrm{out}} \times \mathrm{TPOT}_{h,b}(95) \le L_p^{\star}, \qquad \mathrm{TPOT}_{h,b}(95) = \frac{1}{\mathrm{TPS}_{h,b}(95)}. \tag{8}$$

To evaluate the objective, we expand each impact term at the level of a single routing assignment $(t, r, h, b, \varphi, p, d)$. We first compute accelerator-only energy (Wh), after which the solver chooses an assignment count. In each 5-minute window t:

$$E^{\mathrm{acc}}_{t,r,h,b,\varphi,p,d} = x_{t,r,h,b,\varphi,p,d} \times e^{\mathrm{Wh/token}}(h, \varphi, b; q) \times \tau_{r,h,b,\varphi,p,d}, \tag{9}$$

with energy per output token in decode:

$$e^{\mathrm{Wh/token}}(h, decode, b; q) = \bar{P}^{\mathrm{acc}}_{h,b}(decode)\, \frac{1}{\mathrm{TPS}_{h,b}(q)} \times \frac{1}{3600}, \tag{10}$$

where $\bar{P}^{\mathrm{acc}}_{h,b}$ is the accelerator-only average power (W).

Before the first token appears, the system spends $\mathrm{TTFT}_{h,b}(q)$ seconds on prefill; spreading that energy over the T_p^{in} input tokens for profile p makes the prefill term linear and comparable:

$$e^{\mathrm{Wh/token}}(h, prefill, b; q) = \bar{P}^{\mathrm{acc}}_{h,b}(prefill)\, \frac{\mathrm{TTFT}_{h,b}(q)}{T_p^{\mathrm{in}}} \times \frac{1}{3600}. \tag{11}$$

We then lift to IT energy with the production-observed host+idle scaler $\kappa_{\text{host+idle}}$ (cf. Eq. (1)), and to facility (comprehensive) energy via the site's PUE:

$$E^{\mathrm{IT}}_{t,r,h,b,\varphi,p,d} = \kappa_{\text{host+idle}} \times E^{\mathrm{acc}}_{t,r,h,b,\varphi,p,d}, \tag{12}$$

$$E^{\mathrm{fac}}_{t,r,h,b,\varphi,p,d} = \mathrm{PUE}_r \times E^{\mathrm{IT}}_{t,r,h,b,\varphi,p,d}. \tag{13}$$

Consumptive water is obtained by applying the site-level cooling term and source-site electricity water to IT energy:

$$\mathrm{mL}_{t,r,h,b,\varphi,p,d} = E^{\mathrm{IT}}_{t,r,h,b,\varphi,p,d} \times \left(\mathrm{PUE}_r \times \mathrm{WUE}_r^{\mathrm{site}}\right) + E^{\mathrm{IT}}_{t,r,h,b,\varphi,p,d} \times \mathrm{EWIF}_r^{\mathrm{source}}. \tag{14}$$

Operational emissions use facility energy times either the provided market-based factor or the location-based grid factor; with LB as default:

$$g\mathrm{CO}^{\mathrm{LB}}_{2,t,r,h,b,\varphi,p,d} = E^{\mathrm{fac}}_{t,r,h,b,\varphi,p,d} \times \mathrm{CIF}_r \times 10^3. \tag{15}$$

Finally, the embodied CO_2 and e-waste are amortized as daily charges that activate once per hardware class used:

$$\mathrm{CO}^{\mathrm{emb}}_{2,\mathrm{day}} = \sum_{r,h} z_{r,h}\, \frac{C_h^{\mathrm{emb}}}{L_h}, \qquad W^{\mathrm{ewaste}}_{\mathrm{day}} = \sum_{r,h} z_{r,h}\, \frac{M_h^{\mathrm{board}}}{L_h}. \tag{16}$$

Summing Eqs. (9)–(16) over all (r, h, b, p, φ, d) assignments in a window yields the window-level $E_t, mL_t, gCO_{2,t}$ that enter the objective in (5); summing over t and taking daily medians produces the reported statistics, with LB as default and MB as sensitivity [1,3,25,56].

Algorithm 1 Σ-Scale (daily-coupled MILP; p95 SLOs, replicas, daily binaries)

```
Inputs: Sets: regions $R$, hardware classes $H$, batches $B$, phases $\Phi = \{\text{prefill}, \text{decode}\}$, profiles $P$,
    directives $D = \{\text{default}, \text{brief}\}$, windows $T = \{1, \ldots, 288\}$; window budget $\Delta t = 300$ s.
Inputs: Arrivals $Q_{t,p}$ per window and profile, either provided or constructed from $N_{\text{day}}$, mix $\pi_p$, and
    diurnal weights $f(t)$: $Q_{t,p} = \text{round}\big(N_{\text{day}} \pi_p f(t) / \sum_{t'} f(t')\big)$.
Inputs: p95 quantile tables: $\text{TPS}_{h,b}(95)$, $\text{TTFT}_{h,b}(95)$, $\text{TPOT}_{h,b}(95) = 1/\text{TPS}_{h,b}(95)$.
Inputs: Site rows per $r$: $\text{PUE}_r$, $\text{WUE}_{\text{site},r}$, $\text{EWIF}_r^{\text{source}}$, $\text{CIF}_r^{\text{LB}}$ (optionally $EF_r^{\text{MB}}$).
Inputs: Lift and SLOs: $\kappa_{\text{host+idle}} \geq 1$, $L_p^\star$; weights $\alpha, \beta, \lambda, \delta, \varepsilon \geq 0$.
Inputs: Embodiment: $C_h^{\text{emb}}$ (kg CO$_2$e/board), $M_h^{\text{board}}$ (kg), lifetime $L_h$ (days).
Inputs: Accelerator-only per-token models $e_{\text{Wh/token}}(h, \varphi, b)$; tokens $T_p^{\text{in}}$, $T_{p,\text{default}}^{\text{out}}$; directive multipliers $\alpha_d \in (0, 1]$; replica caps $U_{r,h,b} \in \mathbb{Z}_{\geq 0}$.
Outputs: Decision variables:
    $x_{t,r,h,b,\varphi,p,d} \in \mathbb{Z}_{\geq 0}$ (assignments), $n_{t,r,h,b} \in \mathbb{Z}_{\geq 0}$ (replicas), $u_{t,r,h,b} \in \{0,1\}$ (activation),
    $z_{r,h} \in \{0,1\}$ (daily activation).
    Optional: $y_{t,r,h,b,p,d} \in \{0,1\}$ (usage, for gated SLO).
Per-prompt coefficients for each $(r, h, b, \varphi, p, d)$ in Wh/prompt:
    $T_{p,d}^{\varphi} \leftarrow \big(T_p^{\text{in}}$ if $\varphi = \text{prefill}$; $\alpha_d T_{p,\text{default}}^{\text{out}}$ if $\varphi = \text{decode}\big)$
    $E_{h,b,\varphi,p,d}^{\text{acc}} \leftarrow e_{\text{Wh/token}}(h, \varphi, b)\, T_{p,d}^{\varphi}$
    $E_{h,b,\varphi,p,d}^{\text{IT}} \leftarrow \kappa_{\text{host+idle}}\, E_{h,b,\varphi,p,d}^{\text{acc}}$
    $E_{r,h,b,\varphi,p,d}^{\text{fac}} \leftarrow \text{PUE}_r\, E_{h,b,\varphi,p,d}^{\text{IT}}$
    $\text{mL}_{r,h,b,\varphi,p,d} \leftarrow E_{h,b,\varphi,p,d}^{\text{IT}} \big(\text{PUE}_r\, \text{WUE}_{\text{site},r} + \text{EWIF}_r^{\text{source}}\big)$
    $g\text{CO}_{2,r,h,b,\varphi,p,d}^{\text{LB}} \leftarrow E_{r,h,b,\varphi,p,d}^{\text{fac}}\, \text{CIF}_r^{\text{LB}}$
Minimize (global daily sum):
    $\alpha \sum_{t,r,h,b,\varphi,p,d} x_{t,r,h,b,\varphi,p,d} E_{h,b,\varphi,p,d}^{\text{IT}} + \beta \sum_{t,r,h,b,\varphi,p,d} x_{t,r,h,b,\varphi,p,d}\, \text{mL}_{r,h,b,\varphi,p,d} + \lambda \sum_{t,r,h,b,\varphi,p,d} x_{t,r,h,b,\varphi,p,d}\, g\text{CO}_{2,r,h,b,\varphi,p,d}^{\text{LB}}$
        $+ \sum_{r,h} z_{r,h} \Big(\delta \frac{C_h^{\text{emb}}}{L_h} + \varepsilon \frac{M_h^{\text{board}}}{L_h}\Big)$
Subject to
(i) Conservation per $(t, p)$ and phase link:
    $\sum_{r,h,b,d} x_{t,r,h,b,\text{prefill},p,d} = \sum_{r,h,b,d} x_{t,r,h,b,\text{decode},p,d} = Q_{t,p} \quad \forall t, p.$
(ii) Capacity with replicas (RHS scaled by $n$):
    $\sum_{p,d} \Big( \frac{x_{t,r,h,b,\text{prefill},p,d}\, T_p^{\text{in}}}{\text{TPS}_{h,b}(95)} + \frac{x_{t,r,h,b,\text{decode},p,d}\, \alpha_d T_{p,\text{default}}^{\text{out}}}{\text{TPS}_{h,b}(95)} \Big) \leq \Delta t\, n_{t,r,h,b} \quad \forall t, r, h, b.$
(iii) Replica bounds and activation:
    $0 \leq n_{t,r,h,b} \leq U_{r,h,b}\, u_{t,r,h,b} \quad \forall t, r, h, b.$
(iv) Daily coupling (embodiment once/day):
    $z_{r,h} \geq u_{t,r,h,b} \quad \forall t, r, h, b.$
(v) Link assignments to activation (tight big-M):
    $x_{t,r,h,b,\varphi,p,d} \leq M_{h,b,p,d,\varphi}\, u_{t,r,h,b} \quad \forall t, r, h, b, \varphi, p, d.$
(vi) Latency SLO at p95 (choose one form):
    Ungated: $\text{TTFT}_{h,b}(95) + \alpha_d T_{p,\text{default}}^{\text{out}} \text{TPOT}_{h,b}(95) \leq L_p^\star \;\; \forall h, b, p, d.$
    Optional gated: $\text{TTFT}_{h,b}(95) + \alpha_d T_{p,\text{default}}^{\text{out}} \text{TPOT}_{h,b}(95) \leq L_p^\star + M'_{h,b,p,d}\big(1 - y_{t,r,h,b,p,d}\big)$, with
        $x_{t,r,h,b,\varphi,p,d} \leq \widehat{M}_{h,b,p,d,\varphi}\, y_{t,r,h,b,p,d}$ and $y_{t,r,h,b,p,d} \leq u_{t,r,h,b}$.
Solve the MILP once over all $t \in T$; return $x^\star, n^\star, u^\star, z^\star$.
```

Algorithm 2 Aggregation and scope-transparent reporting (post-solve)

```
Inputs: Optimal decisions $x^\star, n^\star, u^\star, z^\star$; coefficients $E^{\mathrm{IT}}, E^{\mathrm{fac}}, \mathrm{mL}, g\mathrm{CO}_2^{\mathrm{LB}}$; arrivals $Q_{t,p}$; profile mix $(0.70, 0.25, 0.05)$.
Outputs: Daily per-prompt medians by profile, then mixed medians; comprehensive default, accelerator-only whiskers; LB default with MB sensitivity.
for all $(t, p)$ do
    $E^{\mathrm{IT}}_{t,p} \leftarrow \sum_{r,h,b,\varphi,d} x^\star_{t,r,h,b,\varphi,p,d}\, E^{\mathrm{IT}}_{h,b,\varphi,p,d}$
    $\mathrm{mL}_{t,p} \leftarrow \sum_{r,h,b,\varphi,d} x^\star_{t,r,h,b,\varphi,p,d}\, \mathrm{mL}_{r,h,b,\varphi,p,d}$
    $g\mathrm{CO}^{\mathrm{LB}}_{2,t,p} \leftarrow \sum_{r,h,b,\varphi,d} x^\star_{t,r,h,b,\varphi,p,d}\, g\mathrm{CO}^{\mathrm{LB}}_{2,r,h,b,\varphi,p,d}$
    Normalize to per-prompt: $E^{\mathrm{IT}}_{t,p} = E^{\mathrm{IT}}_{t,p} / Q_{t,p}$, $w_{t,p} = \mathrm{Water}_{t,p} / Q_{t,p}$, $c^{\mathrm{LB}}_{t,p} = g\mathrm{CO}^{\mathrm{LB}}_{2,t,p} / Q_{t,p}$.
end for
For each profile $p$, take the median over $t = 1{:}288$ of $(E^{\mathrm{IT}}_{t,p}, w_{t,p}, c^{\mathrm{LB}}_{t,p})$; then mix by $0.70/0.25/0.05$ to get per-model medians.
Embodied and e-waste (once/day): $\mathrm{CO2}_{\mathrm{emb,day}} = \sum_{r,h} z^\star_{r,h}\, C^{\mathrm{emb}}_h / L_h$, $\mathrm{Waste}_{\mathrm{day}} = \sum_{r,h} z^\star_{r,h}\, M^{\mathrm{board}}_h / L_h$.
Scopes. Report comprehensive (default) via $E^{\mathrm{fac}} = \mathrm{PUE}_r\, E^{\mathrm{IT}}$; add accelerator-only whiskers using $E^{\mathrm{acc}}$ (or the observed narrow→comprehensive ratio when shares are undisclosed).
Emissions. Default LB via $\mathrm{CIF}^{\mathrm{LB}}_r$; add MB sensitivity by substituting $EF^{\mathrm{MB}}_r$.
```

3.5. Parameterization from Public Sources

We adopt the comprehensive serving boundary by default: active accelerators together with host CPU/DRAM and provisioned idle, lifted to facility energy by the site PUE in Eqs. (12)–(13). When only accelerator-only measurements are disclosed, we translate to the comprehensive boundary with a host+idle lift and PUE, or when shares are undisclosed, by using the production-observed narrow→comprehensive median ratio ≈ 0.417 for comparable workloads [1]. Reporting uses daily medians to mirror production practice and to keep results comparable across models and regions [1].

Baseline per-prompt energy and throughput/latency quantiles by model and profile are taken from the cross-model API benchmark [5]. We ingest watt-hours per prompt together with the p-quantile tokens-per-second and latency rows, for example p95 TTFT, TPS, and TPOT. These quantiles parameterize the capacity and tail-latency constraints in Eqs. (7)–(8). As an illustration, the table for GPT-4o lists a short-profile energy near 0.421 Wh and provides site multipliers; we keep the published provider/region mapping [5].

Per-token coefficients for prefill and decode follow the device-level accelerator-only power models used in Eqs. (10)–(11). The average accelerator power $\bar{P}_{\mathrm{acc},h,b}(\varphi)$ is taken per hardware class and batch. We lift accelerator-only energy to IT with the host+idle scaler $\kappa_{\mathrm{host+idle}}$ in Eq. (12) and then to facility with the site PUE in Eq. (13), consistent with the comprehensive boundary in [1].

Consumptive water is computed with the scope-consistent site+source formulation in Eq. (14). The site component scales with facility energy via $\mathrm{WUE}_{\mathrm{site},r}$. The source component scales with IT energy via the regional electricity–water intensity factor $\mathrm{EWIF}_{\mathrm{source},r}$. Priors and bounds for site WUE come from the LBNL synthesis and AI water-footprint guidance [3,56]. Electricity–water intensity values follow the scope-2 definition in [3] and are applied as annual, location-based factors when hourly mixes are unavailable.

Operational emissions are location-based by default using regional CIF with facility energy in Eq. (15). We also provide a market-based sensitivity by substituting the provider's prior-year portfolio factor when disclosed [1,25]. Units are reported in the native form for each metric.

For the phase-aware variant, we use accelerator-only per-token energy curves that separate compute-bound prefill from memory-bound decode across relevant batches. This enables assignments that reduce Wh per prompt while respecting p95 SLOs [22]. Embodied carbon per board, board mass, and lifetimes are taken from serving and lifecycle assessments [16,53] and are amortized once per day when a hardware class is activated, as in Eq. (16).

Reference ranges for site factors follow [56]. Hyperscale PUE trends to roughly 1.15–1.35. Site WUE is around 0.32–0.40 $\mathrm{L\,kWh^{-1}}$ in 2023 with a likely rise as liquid cooling penetrates. Scenario sweeps therefore use ±0.10 around a regional PUE baseline and ±25% around site WUE. When we construct hydro-like or nuclear-like sites for Pareto illustrations, CIF and electricity–water intensity are kept within documented ranges and flagged as replaceable medians when provider rows are disclosed [3,56].

All numerical inputs—per-model Wh per prompt, p95 TTFT/TPS/TPOT by batch, site rows with PUE, site WUE, electricity–water intensity, and location-based carbon intensity, and embodiment parameters—are taken from peer-reviewed or provider-authored sources [1,3,5,56]. We package them into machine-readable tables so figures and results regenerate one-for-one when providers update disclosures. When deployment caps are known, we set $U_{r,h,b}$ to the maximum provisionable replicas for class h at region r and batch b. Otherwise $U_{r,h,b}$ is a large operator-chosen bound.

Because grid carbon intensity and electricity–water intensity can be weakly correlated at fine time scales, improving one can worsen the other [3]. This motivates the bi-objective treatment and the routing-only and joint routing+batch+token analyses that make the trade-off geometry explicit under p95 SLOs.

4. Results

We simulate a single 24 hour production day split into 288 five-minute decision windows and a fixed demand of 500 million prompts with a 70/25/5 short/medium/long mix. Short, medium, and long profiles use token pairs (100, 300), (1000, 1000), and (10,000, 15,000), respectively. In every window the Σ Scale optimizer chooses, for each profile, the region, the batch size, and the phase-specific hardware (prefill vs. decode) to minimize an equal-weighted sum of energy, consumptive water, and location-based CO_2 subject to explicit $p95$ SLOs and the capacity and conservation constraints. Capacity is enforced with per-hardware, per-batch tokens-per-second quantiles, and latency feasibility uses $p95$ time-to-first-token and time-per-output-token; the short prompt target is ≤ 0.9 s. Window outputs are aggregated to daily medians per profile and then mixed by 70/25/5 to obtain per-model medians. By default we report the comprehensive serving boundary—active accelerator + host CPU/DRAM + provisioned idle—lifted to facility energy by the site's PUE. Consumptive water is computed with the scope-consistent site+source rule, $E^{\mathrm{IT}}(\mathrm{PUE} \times \mathrm{WUE}_{\mathrm{site}} + \mathrm{EWIF})$; CO_2 is location-based using CIF, with a market-based sensitivity when prior-year portfolio factors are available. For reconciliation with chip-only studies we also compute an accelerator-only subset and translate between scopes using the empirically observed narrow/comprehensive median ratio of 0.417 (approximately 2.4× uplift; e.g., 0.10 Wh accelerator-only vs. 0.24 Wh comprehensive at the median) [1]. Regional multipliers (PUE, $\mathrm{WUE}_{\mathrm{site}}$, EWIF, CIF) and the throughput/latency quantiles that bind the SLOs come from the cross-model public benchmark; a phase-aware variant uses device-level per-token energy curves for prefill/decode and includes embodied-carbon amortization when a hardware class is activated during the day [5]. Two policies are evaluated under identical demand and site multipliers. The *baseline* fixes batch$= 8$, uses the home region only, applies no token directive, and does not split phases. The Σ Scale *optimized* policy right-sizes batch over the day, applies a concise token directive when SLO-safe, routes across sites in response to CIF and EWIF, assigns prefill and decode to different hardware when that lowers Wh/prompt, and amortizes embodied impacts; the contrast between these policies isolates the value of orchestration at a fixed model and boundary.

4.1. Comprehensive Boundary Medians and Daily Totals

Table 1 reports per-prompt median-scaled total and associated daily totals for energy, water (consumptive, site+source), and location-based CO_2 for four representative models. The *Optimized* rows correspond to the Σ Scale loop with all levers active (batch right-sizing, semantic token control, geo-routing, phase-aware assignment, and second-life amortization). *Baseline* uses the same comprehensive boundary and medians but with batch$= 8$, no token control, home region only, and no phase split. Across the four evaluated models, the Σ Scale policy delivers large and consistent savings at the comprehensive serving boundary. After aggregating window outputs to daily medians per profile and mixing by 70/25/5, the median energy per prompt falls by 57–59%, the median consumptive water (site+source) by 59–60%, and the median location-based CO_2 by 78–80%, all while meeting $p95$ capacity and $p95$ latency in every five-minute window. For GPT-4o, the comprehensive baseline is 0.6876 Wh/prompt, 2.3915 mL/prompt, and 0.2426 g CO_2/prompt; after optimization the medians become 0.2898 Wh, 0.9803 mL, and 0.0507 g CO_2, respectively. At a volume of 500 M queries per day these medians translate to energy dropping 0.344 → 0.145 GWh, water 1.196 → 0.490 ML, and LB CO_2 121 → 25 t under the same SLA. The other models follow the same pattern, with absolute magnitudes scaling with baseline Wh/prompt (e.g., Claude 3.7 Sonnet shows the largest absolute reductions because its baselines are highest). These gains reflect full-stack serving and scope-consistent water accounting rather than chip-only power, and they arise from the combined effect of batch right-sizing (e.g., 8 → 16 off-peak), token-length directives, carbon- and water-aware geo-routing, phase-aware hardware assignment, and second-life amortization. See Figure 1 for a side-by-side visualization of the comprehensive-boundary medians (baseline vs. optimized) for energy, consumptive water, and LB CO_2, with lower 'scope whiskers' showing accelerator-only values via the observed narrow/comprehensive ≈ 0.417 ratio.

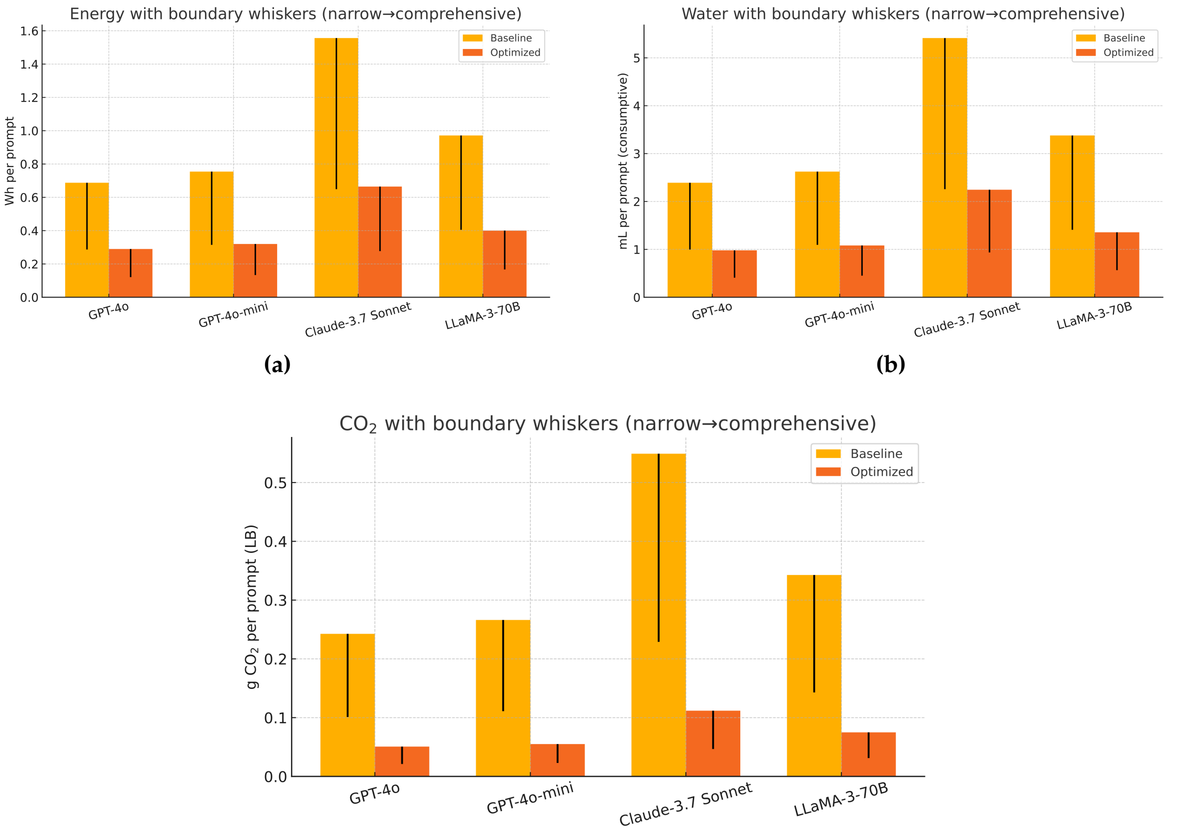
Table 1. Comprehensive boundary medians (weighted by the 70/25/5 mix).

Metric	**GPT 4o**	**GPT 4o mini**	**Claude 3.7 Sonnet**	**LLaMA 3 70B†**
Baseline Wh/prompt	0.6876	0.7545	1.55635	0.97145
Optimized Wh/prompt	0.289824	0.319896	0.664582	0.400321
Δ Energy %	−57.8	−57.6	−57.3	−58.8
Baseline mL/prompt	2.391473	2.624151	5.412985	3.378703
Optimized mL/prompt	0.980349	1.081547	2.245570	1.356734
Δ Water %	−59.0	−58.8	−58.5	−59.8
Baseline g CO_2/prompt (LB)	0.242585	0.266188	0.549080	0.342728
Optimized g CO_2/prompt (LB)	0.050739	0.055035	0.111840	0.074971
Δ CO_2 %	−79.1	−79.3	−79.6	−78.1
Baseline Energy (GWh/d)	0.3438	0.37725	0.778175	0.485725
Optimized Energy (GWh/d)	0.144912	0.159948	0.332291	0.200160
Baseline Water (ML/d)	1.196	1.312	2.706	1.689
Optimized Water (ML/d)	0.490	0.541	1.123	0.678
Baseline CO_2 (t/d, LB)	121.293	133.094	274.540	171.364
Optimized CO_2 (t/d, LB)	25.370	27.517	55.920	37.485

† Long-prompt Wh for LLaMA 3 70B is not reported in the public table; the line aggregates short+medium medians and is flagged accordingly. Units: 1 ML = 10^6 L = 10^9 mL. At 500 M prompts/day, ML/d = 0.5× mL/prompt; GWh/d = 0.5× Wh/prompt; t/d = 500× g/prompt.

4.2. Scope Reconciliation: Accelerator-Only vs. Comprehensive

A persistent source of disagreement in the literature is the accounting boundary used for inference. To make our results directly comparable to chip-only studies, we compute both scopes for the same traffic, mix, and site multipliers and place them side by side. The empirical narrow/comprehensive median energy ratio in our runs is ≈ 0.417, reproducing the ≈ 2.4× uplift observed in production telemetry when host CPU/DRAM, provisioned idle, and facility overheads (PUE) are included [1]. This factor is not universal—it varies with architecture and fleet management—but when underlying shares are unavailable it provides a defensible translation between scopes. Concretely, if a study reports only accelerator-level energy $E_{\text{acc-only}}$ $(\mathrm{Wh\,prompt}^{-1})$, the comprehensive median may be approximated by $E_{\text{comprehensive}} \approx E_{\text{acc-only}}/0.417$, holding PUE, $\mathrm{WUE}_{\text{site}}$, EWIF, and the prompt mix constant.



(a) (b)

(c)

Figure 1. Energy, water, and LB CO_2 medians (baseline vs. optimized). Lower whiskers *(scope whiskers)*: for *energy* and *LB CO_2* we scale the comprehensive medians by the observed accelerator-only/comprehensive ratio (≈ 0.417). For *water*, whiskers indicate a *source-only* value computed as $E^{\mathrm{acc}} \times \mathrm{EWIF}$ (site water set to zero), reflecting the fact that site water scales with facility energy while source water scales with IT/accelerator energy (Eq. 3).

Table 2 provides scope translation to reconcile chip-only studies with comprehensive accounting. The numerical proximity between optimized comprehensive and baseline narrow in our run ($\sim 0.42\times$ of baseline comprehensive in both cases) is coincidental: the former reflects optimization levers and SLOs; the latter reflects boundary accounting shares measured in production. Together with Table 1 (policy effects at the comprehensive boundary), Table 2 furnishes a clear, citable translation layer so that chip-only results can be reconciled with full-stack accounting without ambiguity.

Table 2. Accelerator-only vs. comprehensive (medians, 70/25/5 mix).

Metric	GPT 4o	GPT 4o mini	Claude 3.7 Sonnet	LLaMA 3 70B
Narrow Wh/prompt	0.2865	0.314375	0.648479	0.404771
Comprehensive Wh/prompt	0.6876	0.7545	1.55635	0.97145
Narrow/Comprehensive	0.417	0.417	0.417	0.417
Narrow mL/prompt	0.996447	1.093396	2.255411	1.407793
Comprehensive mL/prompt	2.391473	2.624151	5.412985	3.378703
Narrow g CO_2/prompt (LB)	0.101077	0.110911	0.228783	0.142803
Comprehensive g CO_2/prompt (LB)	0.242585	0.266188	0.549080	0.342728

4.3. Carbon–Water Movement at Fixed QoS

Because grid carbon intensity (CIF) and electricity-water intensity (EWIF) are weakly correlated, carbon-optimal routing can be water-suboptimal. Our Σ Scale policy jointly optimizes both metrics. Figure 2 shows each model's movement from *baseline* to *optimized* in the (g CO_2, mL) plane. For each model we plot the comprehensive-boundary median at the baseline configuration and the corresponding median under the Σ Scale policy; the two points are connected by an arrow. In every case the arrow points down and to the left, showing that the optimized policy reduces both location-based CO_2 and consumptive water while honoring the $p95$ SLOs. This simultaneous reduction is non-trivial because CIF and EWIF are only weakly correlated; a carbon-only router can increase water if it shifts load to a clean but water-intensive region. The Σ Scale objective avoids that pitfall by jointly reducing Wh per prompt—through batch right-sizing, concise token directives, and phase-aware placement—and by routing remaining Wh to regions and hours with favorable CIF and $\text{PUE} \times \text{WUE}_{\text{site}} + \text{EWIF}$.

The geometry is intuitive. Each site i defines a ray through the origin in the $(\text{g CO}_2, \text{mL})$ plane whose slope is W_i/CIF_i, with $W_i = \text{PUE}_i \times \text{WUE}_{\text{site},i} + \text{EWIF}_i$. Moving along a ray changes only energy (Wh) via batching and token control; switching rays changes the water-to-carbon ratio by altering site factors while holding Wh fixed. The optimizer exploits both degrees of freedom: it lowers Wh and, subject to the latency budget, selects rays with more favorable slopes. In a two-region sensitivity where the alternative site is hydro-like with both lower CIF and a lower combined water factor, routing alone becomes a Pareto improvement; the arrow's down-left direction then reflects a pure siting effect. In other configurations, the arrow still points down-left because energy reductions combine with selective routing to offset sites where low carbon coincides with higher water.

To keep scope visible without obscuring directionality, Figure 2 overlays asymmetric boundary whiskers on both endpoints. The barbed end of each whisker marks the accelerator-only value obtained by multiplying the comprehensive coordinate by 0.417 on both axes; there is no upper whisker because comprehensive is our defined upper boundary. The tip-to-tail arrows and their whiskered counterparts are nearly parallel and scaled, illustrating that the direction and magnitude of the optimization gain are robust to scope choice: changing scope shifts absolute values by the familiar ≈ 2.4× but does not alter the qualitative improvement.

This figure is diagnostic rather than a full carbon–water Pareto frontier. Constructing that frontier would require multiple MILP runs with different objective weights. At fixed QoS, the comprehensive, SLO-aware policy achieves concurrent reductions in CO_2 and water for the evaluated workloads. In the next section we quantify the additional reductions available if small latency relaxations or stronger token-length directives are permitted.

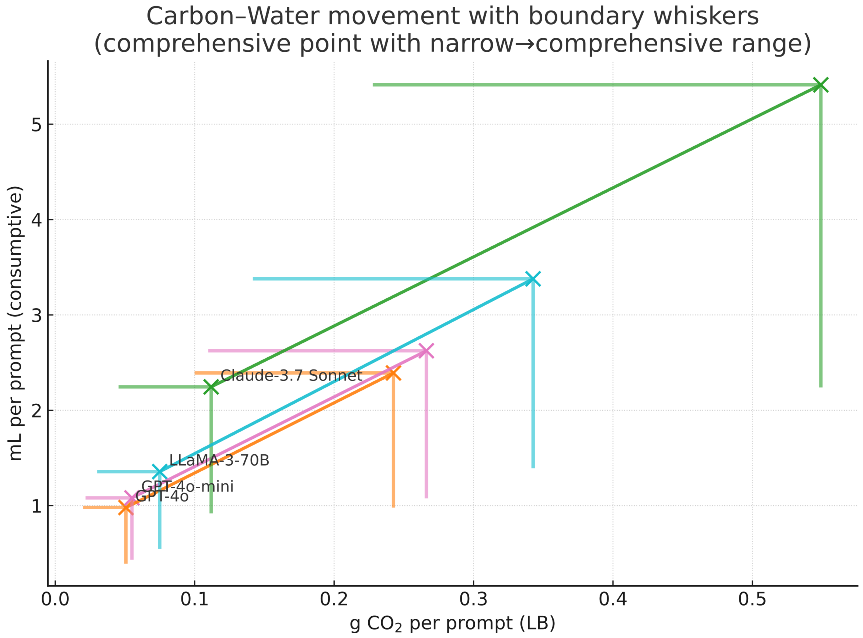


Figure 2. Carbon–water movement at fixed QoS. For each model, the arrow connects the comprehensive-boundary medians from baseline to optimized. Lower whiskers mark accelerator-only values: energy and LB CO_2 via the 0.417 energy ratio; water via source-only $E^{\text{acc}} \times \text{EWIF}$.

4.4. Routing-Only Carbon–Water Pareto Under SLOs

The previous subsection established that the Σ Scale policy moves every model down and to the left in the $(\mathrm{g\,CO_2},\ \mathrm{mL})$ plane: batching and token directives reduce watt-hours per prompt, while geo-routing chooses sites whose environmental multipliers further shrink carbon and water, all without violating the p95 latency SLO. We now isolate the routing lever to understand the geometry of those gains. To avoid conflating routing with operational changes, we hold fixed each model's optimized comprehensive-boundary energy per prompt $E_m^\star$ (Wh prompt^{-1}) and vary only the regional mix subject to the same SLO. This yields an interpretable map of what geo placement alone can deliver once the runtime system has already right-sized batch and applied concise generation.

Let $s = (s_b, s_h, s_n)$ denote the shares routed to a baseline U.S. thermal site b, a hydro-dominated site h, and a nuclear-like thermal site n, with $s_i \geq 0$ and $s_b + s_h + s_n = 1$. Each site i is characterized by a location-based grid carbon intensity CIF_i (kg $\mathrm{CO_2}$ kWh^{-1}) and a consumptive water factor W_i, following a scope-consistent convention (on-site cooling scaled by PUE, plus source-side electricity–water intensity added directly). With $E_m^\star$ fixed, the per-prompt impacts for any routing mix s are linear:

$$\mathrm{g\,CO_2/prompt}(s) = E_m^\star \sum_{i\in\{b,h,n\}} s_i\,\mathrm{CIF}_i, \qquad \mathrm{mL/prompt}(s) = E_m^\star \sum_{i\in\{b,h,n\}} s_i\,W_i. \tag{17}$$

Equation (17) implies that the feasible set in the $(\mathrm{g\,CO_2},\ \mathrm{mL})$ plane is the convex hull of the three single-site vertices $\{\, E_m^\star\mathrm{CIF}_i,\ E_m^\star W_i\,\}$; the Pareto frontier is the lower-left convex envelope of those vertices. Each site also defines a ray through the origin with slope W_i/CIF_i: moving along a ray changes only the energy scale, whereas switching rays changes the water-to-carbon ratio at fixed energy.

Because the hydro site in our parameterization couples very low CIF_h with a high electricity–water intensity (we use 5.50 L kWh^{-1} to stress the conflict), while the nuclear-like site has a lower combined water factor W_n but a higher carbon intensity CIF_n than hydro, the non-dominated set collapses to the straight edge joining hydro and nuclear. The baseline site lies to the right of both in carbon and does not beat nuclear on water; hence it is dominated. Along the efficient edge, a single parameter $\lambda \in [0,1]$ (the nuclear share) describes the trade-off curve:

$$\mathrm{mL/prompt}(\lambda) = E_m^\star\big[(1-\lambda)W_h + \lambda W_n\big], \tag{18}$$

$$\mathrm{g\,CO_2/prompt}(\lambda) = E_m^\star\big[(1-\lambda)\mathrm{CIF}_h + \lambda\mathrm{CIF}_n\big]. \tag{19}$$

The slope of this edge,

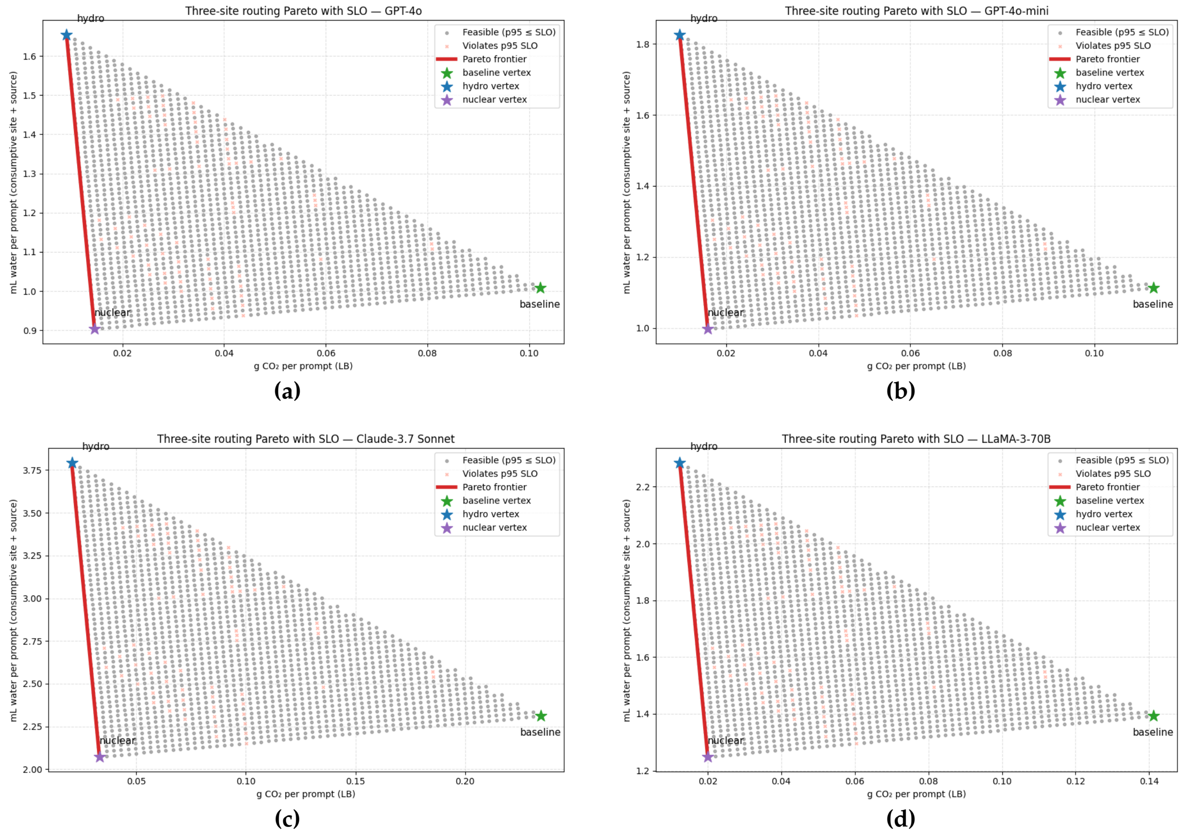
$$\frac{d\,(\mathrm{mL/prompt})}{d\,(\mathrm{g\,CO_2/prompt})} = \frac{W_n - W_h}{\mathrm{CIF}_n - \mathrm{CIF}_h}, \tag{20}$$

depends only on site properties and is therefore independent of $E_m^\star$. This scaling invariance explains why the four model panels share the same shape but span different numerical ranges: increasing $E_m^\star$ stretches both axes uniformly without changing which combinations are efficient.

Latency feasibility is enforced by a conservative p95 proxy:

$$\mathrm{TTFT_{p95}}(s) = \sum_{i\in\{b,h,n\}} s_i\,\mathrm{TTFT}_{i,\mathrm{p95}} \;\leq\; \mathrm{SLO}, \tag{21}$$

so the routing cloud shows exactly those mixes that respect the interactive QoS; infeasible mixes are shaded separately for transparency. With this setup in place, the four panels in Figure 3, for GPT-4o, GPT-4o-mini, Claude-3.7 Sonnet, and LLaMA-3-70B respectively, the feasible routing simplex in $(\mathrm{g\,CO_2},\ \mathrm{mL})$ space, the hydro and nuclear vertices that determine the efficient edge via (18)–(20), and the dominated baseline vertex. Selecting any point on the red edge is equivalent to choosing λ to meet a carbon or water budget at unchanged quality of service.



(a) **(b)**

(c) **(d)**

Figure 3. Three-site routing Pareto with SLO (one panel per model). Each dot is a routing mix $s = (s_b, s_h, s_n)$ across a baseline U.S. thermal site, a hydro-dominated site, and a nuclear-like site, evaluated at the model's optimized energy $E^\star_m$. Coordinates follow (17): g $\mathrm{CO_2} = E^\star_m \sum_i s_i\,\mathrm{CIF}_i$ and mL $= E^\star_m \sum_i s_i\,W_i$ with. Gray points satisfy the p95 TTFT SLO (21); salmon points violate it. The red polyline is the lower-left Pareto frontier. Under hydro low CIF but high EWIF, the efficient set is the hydro↔ nuclear edge; the baseline vertex is dominated (higher carbon and no water advantage).

4.5. Joint Frontiers from Site + Batch + Token Sweeps Under the SLO

The routing-only picture is intentionally conservative because it holds $E^\star_m$ fixed. In practice, the serving stack can reduce energy substantially by right-sizing batch and encouraging concise generations when quality allows. To quantify how these operational levers compound with routing, we extend the enumeration to include a batch multiplier b and a token-length multiplier t (both in $[0,1]$; in our grids $b \in [0.561, 1]$ for batch $8 \to 16$ medians, $t \in [0.7175, 1]$ for default→brief). For a given routing mix s, the effective energy becomes

$$E(b,t) = E^\star_m bt,$$

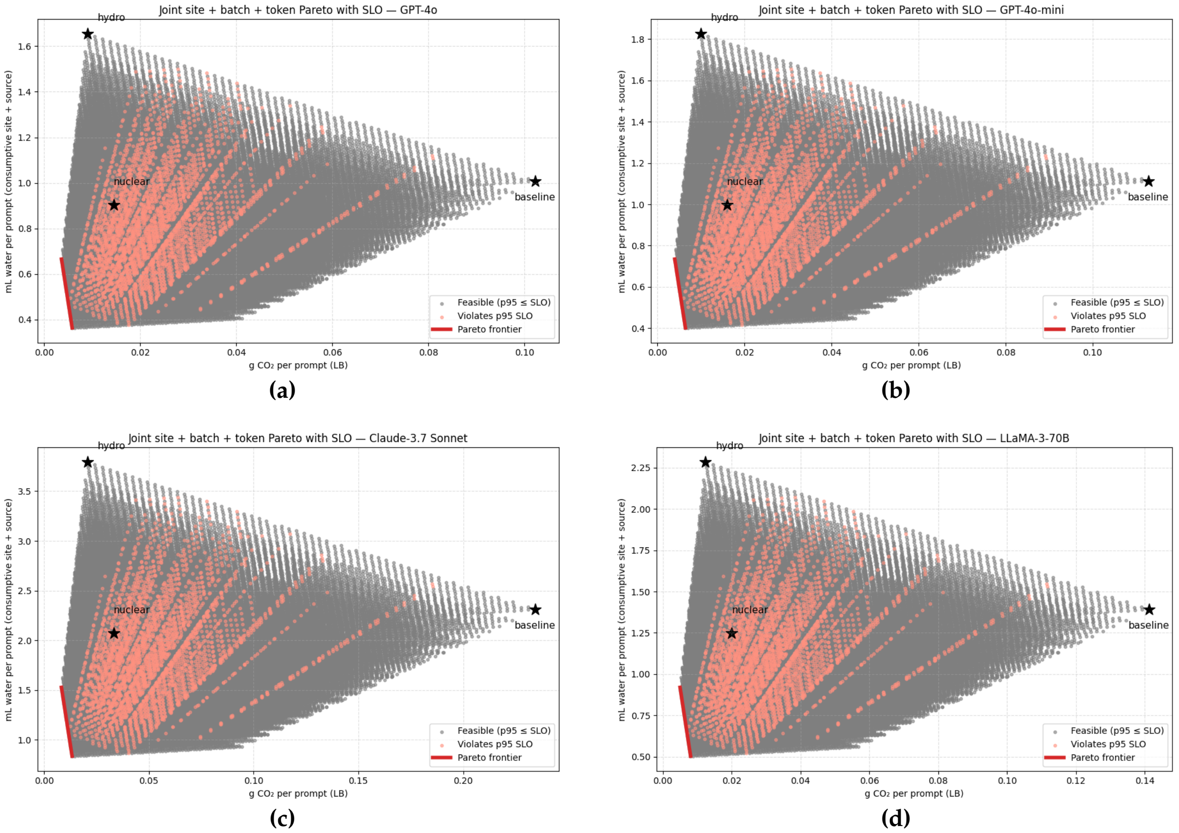
and the per-prompt impacts generalize to:

$$\text{g CO}_2/\text{prompt}(s,b,t) = E^\star_m bt \sum_{i\in\{b,h,n\}} s_i\,\mathrm{CIF}_i, \tag{21a}$$

$$\text{mL}/\text{prompt}(s,b,t) = E^\star_m bt \sum_{i\in\{b,h,n\}} s_i\,W_i. \tag{21b}$$

Because both axes are linear in energy, every point in the routing triangle is translated down and left by the factor bt. As b and t vary over their feasible ranges, the single routing triangle thickens into a wedge of scaled triangles, and the joint Pareto frontier is the lower-left envelope of that wedge. In the feasibility test we retain the conservative p95 proxy of (21), i.e., we mix per-site p95 TTFT linearly by s. When providers supply measured p95 curves as functions of batch and region, those can be dropped into the same mechanism to further tighten feasibility; the frontier construction itself is unchanged.

The four panels in Figure 4 sweep s, b, and t for GPT-4o, GPT-4o-mini, Claude-3.7 Sonnet, and LLaMA-3-70B. Gray points satisfy the p95 SLO; salmon points would violate it. The red curve in each panel is the feasible lower-left envelope. Black stars mark the three single-site anchors at $b = t = 1$; by construction these stars sit inside the feasible cloud and are typically dominated, because there exist mixes with the same s but smaller bt that strictly reduce both axes while remaining inside the SLO. Relative to the routing-only panels, the frontiers shift markedly down and left, quantifying the additional reductions unlocked by batching and concise tokens. For GPT-4o and GPT-4o-mini the feasible wedge is dense and the frontier hugs the extreme lower left. For Claude-3.7 Sonnet, which has the largest $E^{\star}_{m}$, the SLO trims the feasible set more aggressively near the high-batch regime, revealing the practical limit imposed by interactivity. Across all models the direction of the frontier continues to align with the hydro–nuclear edge because hydro remains carbon-optimal while nuclear remains water-optimal. Operationally, routing determines *where* energy is consumed (through CIF and W); batching and tokens determine *how much* energy per prompt is consumed (through $E^{\star}_{m}bt$). The levers compound.



(a)

(b)

(c)

(d)

Figure 4. Joint site + batch + token Pareto with SLO (one panel per model). Each dot corresponds to a triple (s, b, t) of routing shares, batch multiplier, and token-length multiplier. Impacts follow (21): g $\mathrm{CO}_2 = E^{\star}_{m}bt\,\mathrm{CIF}_{\mathrm{mix}}$ and $\mathrm{mL} = E^{\star}_{m}bt\,\mathrm{W}_{\mathrm{mix}}$, where $\mathrm{CIF}_{\mathrm{mix}} = \sum_i s_i\,\mathrm{CIF}_i$ and $W_{\mathrm{mix}} = \sum_i s_i\,W_i$. Gray points satisfy the p95 TTFT SLO, salmon violate it. The red curve is the feasible lower-left Pareto frontier. Stars mark the single-site anchors at $b = t = 1$, which are dominated once batching and concise tokens are allowed. Relative to routing only, the feasible cloud becomes a wedge of scaled triangles and the frontier moves down and left, illustrating how operational levers compound with siting.

4.6. *Ablation: Where the Gains Come from (Fixed p95 SLOs)*

We quantify the marginal contribution of each operational lever at fixed quality of service, using the same data and boundary choices as in Methods and in §4.1. We evaluate feasible point clouds in the $(\mathrm{gCO}_2, \mathrm{mL})$ plane under the p95 latency constraint in (8), capacity in (7), and conservation in (6). We vary exactly one degree of freedom at a time: routing; routing+batch; routing+batch+token; routing+phase split. As in the main results, carbon is location based by default and water is consumptive site+source via (14). When an ablation holds energy fixed we use the per-model optimized energy per prompt $E_m^\star$ from §4.1. Enumeration is deterministic with no RNG. Daily medians are computed by profile and then mixed 70/25/5 for short, medium, and long prompts.

A1 Routing-only. Holding $E_m^\star$ fixed, let $s = (s_b, s_h, s_n)$ denote the shares across the three representative sites with $s_i \geq 0$ and $\sum_i s_i = 1$. Impacts are affine in s by (17), so the feasible set is the routing simplex and the efficient set is its lower-left envelope. For very low CIF$_h$ but higher EWIF$_h$ for hydro, the efficient edge collapses to hydro↔nuclear and the baseline vertex is dominated (see Figure 5). Latency feasibility overlays follow the conservative p95 mix proxy in (21).

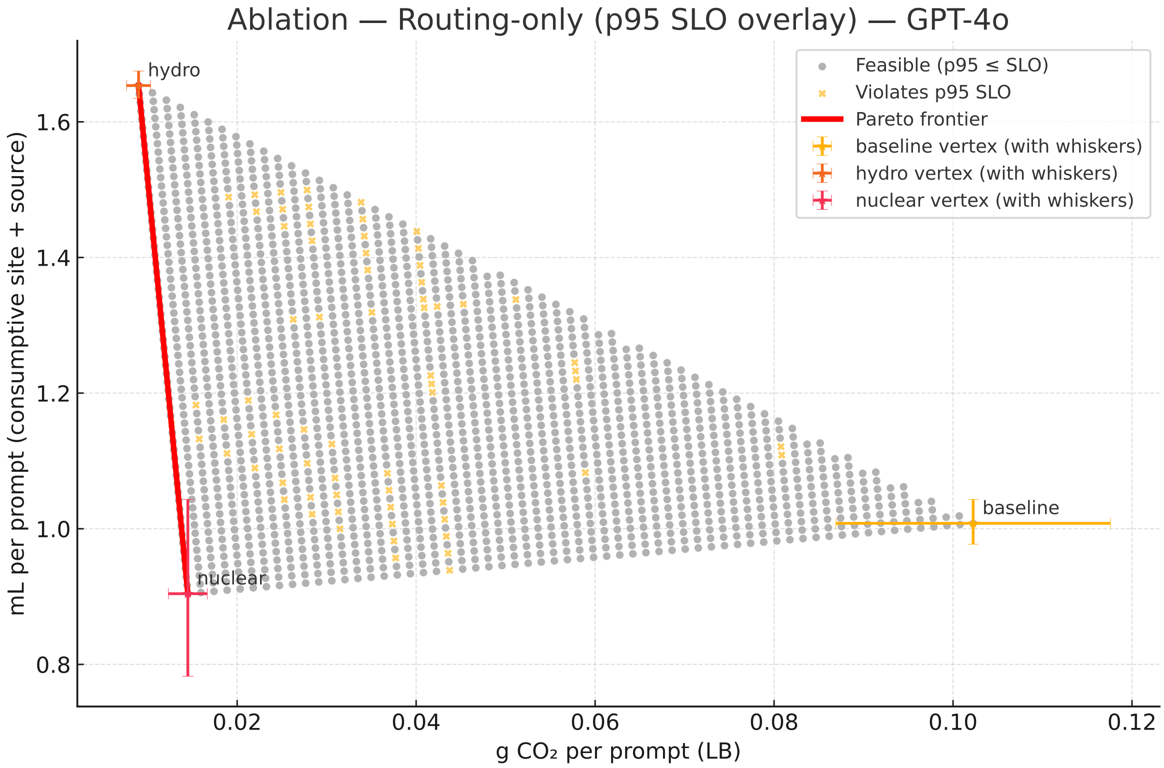


Figure 5. Ablation A1 — Routing-only, SLO overlay (GPT-4o). Feasible routing simplex in the $(\text{g CO}_2, \text{mL})$ plane at fixed optimized energy $E_m^\star$, with site factors (hydro: very low CIF but higher EWIF). Stars mark the three single-site vertices; ○: p95-feasible; ×: p95-violating (mix proxy (21)). The red polyline is the lower-left Pareto frontier; it coincides with the hydro↔nuclear edge. Sensitivity whiskers at vertices: PUE $\pm\,0.10$, WUE$_{\text{site}}$ $\pm\,25\%$ (water), CIF $\pm15\%$ (carbon) or LB↔MB when available.

A2 Routing + Batch. We add a batch multiplier $b \in [b_{\min}, 1]$ that scales energy nearly multiplicatively (while respecting (8)):

$$E_m^\star \;\mapsto\; b\,E_m^\star, \qquad (\text{g}, \text{mL}) \;\mapsto\; b\,(\text{g}, \text{mL}).$$

Each routing triangle is scaled down and left by b, which shifts the Pareto envelope toward the origin (see Figure 6).

A3 Routing + Batch + Token (wider wedge). We also sweep a token-length multiplier $t \in [t_{\min}, 1]$ (*default*→*brief*). With $E_m^\star \mapsto bt\,E_m^\star$, both axes contract by the same bt factor and the impacts follow the joint form already defined in (21a)–(21b). The feasible cloud thickens and the efficient frontier extends down and left relative to A2 (see Figure 7).

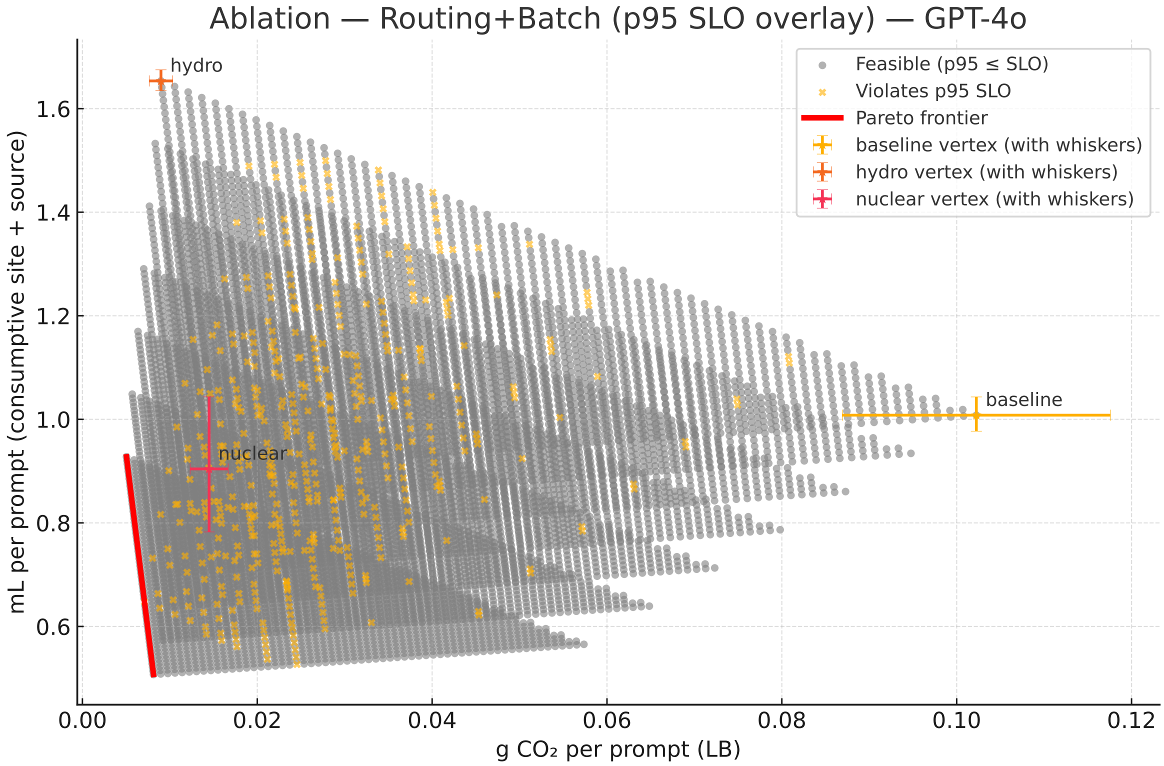


Figure 6. Ablation A2 — Routing + Batch, SLO overlay (GPT-4o). Adds a batch multiplier $b \in [b_{\min}, 1]$ that scales energy and thus both axes by b while meeting p95 SLOs. The routing triangle thickens into a wedge of scaled triangles; the feasible red frontier shifts down-left relative to A1. Vertex whiskers as in Fig. 5.

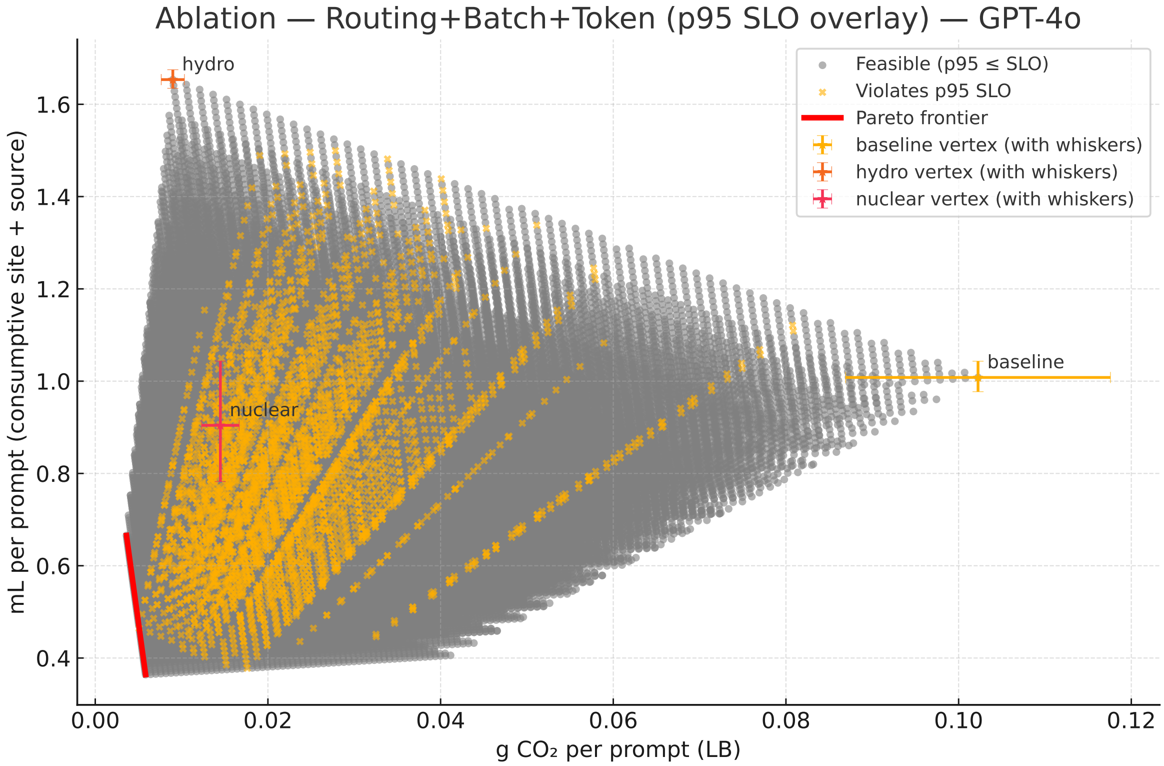


Figure 7. Ablation A3 — Routing + Batch + Token, SLO overlay (GPT-4o). Adds a token-length multiplier $t \in [t_{\min}, 1]$ (default→brief), giving a uniform bt contraction on both axes (cf. (**??**)). The feasible cloud expands and the lower-left frontier lengthens compared to A2; single-site stars at $b = t = 1$ are dominated once batching and concise tokens are allowed. Vertex whiskers as in Fig. 5.

A4 Routing + Phase split (prefill vs. decode). Using the phase-aware per-token energy models $e_{\mathrm{Wh/token}}(h, \varphi, b)$ and the prefill and decode constructions in (10)–(11), we expose the energy-shaping effect of placing prefill and decode on different cohorts. For visualization we sweep phase multipliers $(\eta_{\mathrm{prefill}}, \eta_{\mathrm{decode}}) \in [0.90, 1]^2$ with decode share $\rho \in [0, 1]$ and map $E_m^\star \mapsto E_m^{\star,\mathrm{phase}} = E_m^\star((1-\rho)\eta_{\mathrm{prefill}} + \rho\, \eta_{\mathrm{decode}})$ subject to the same SLO guard. The frontier shifts further toward the origin. The hydro↔nuclear edge remains slope-setting with slope given by (20) (see Figure 8).

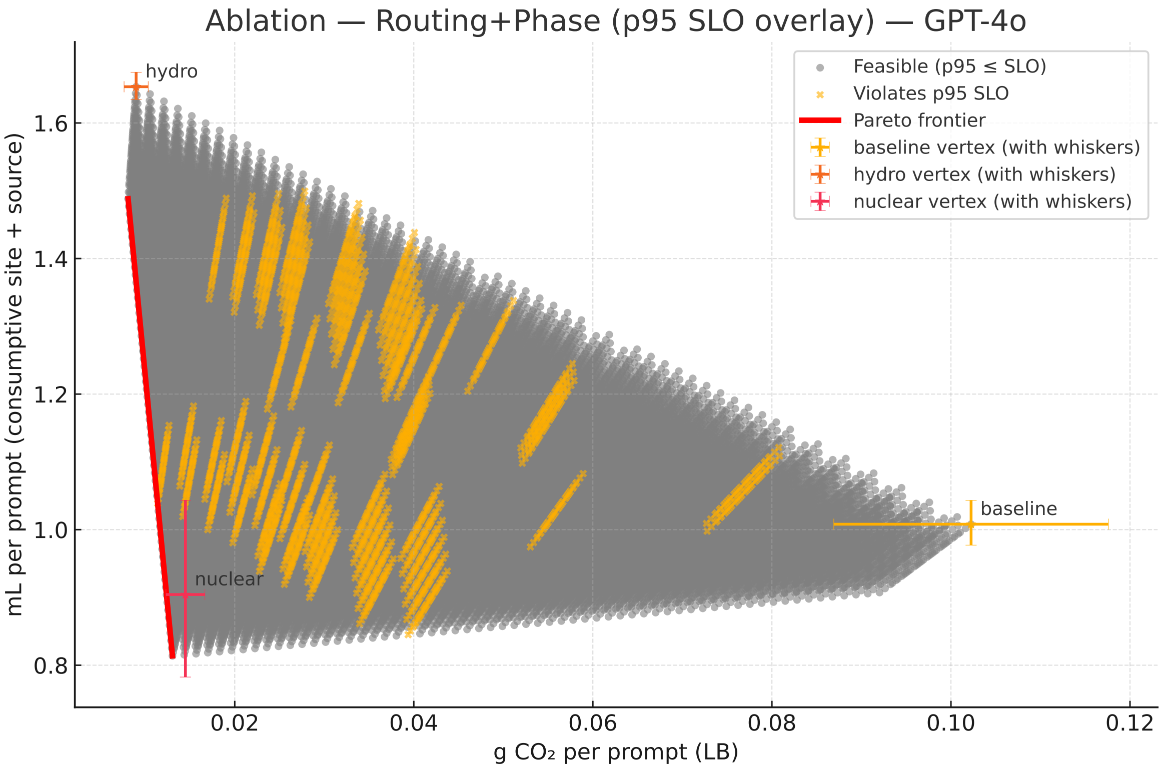


Figure 8. Ablation A4 — Routing + Phase split, SLO overlay (GPT-4o). Prefill (compute-bound) and decode (memory-bound) are allowed to run on different cohorts using phase-aware $e_{\mathrm{Wh/token}}(h, \varphi, b)$; we sweep $(\eta_{\mathrm{prefill}}, \eta_{\mathrm{decode}}) \in [0.90, 1]^2$ with default decode share $\rho \approx 0.7$. The feasible cloud nudges further down-left relative to A1; the hydro↔nuclear edge still sets the trade-off slope ((20)). Whiskers and SLO overlay as before.

In all four panels, ∘ points satisfy p95 SLOs; × points violate. To keep uncertainty visible without clutter, we show *vertex whiskers*: for water, PUE $\in [\mathrm{PUE}_0 \pm 0.10]$ and $\mathrm{WUE}_{\mathrm{site}} \in [0.75,\ 1.25] \times \mathrm{WUE}_{\mathrm{site},0}$ (holding EWIF at the published median); for carbon, a ±15% CIF envelope when only LB is available, otherwise an LB↔MB swap whisker when provider MB factors are disclosed. These match the uncertainty bands and scope practice used elsewhere in the paper.

Across A1 to A4 the feasible set widens and the frontier moves toward the origin. Routing chooses the slope through the site mix. Batch, token, and phase split reduce watt-hours per prompt at fixed QoS, which shrinks both axes. This explains why the optimized medians in Table 1 land down and left of the baseline points without relaxing SLOs and why single-site anchors are dominated once the operational levers are enabled.

5. Discussion

This work introduces a provider-agnostic, time-resolved framework that couples scope-transparent measurement with a mixed-integer orchestration loop (Σ-Scale) to co-minimize the carbon and water footprints of LLM serving under production-grade SLOs. Reporting *daily medians* at a *comprehensive* boundary—active accelerators + host CPU/DRAM + provisioned idle with PUE—resolves a major source of disagreement in prior studies and aligns with operational telemetry. In this boundary, the empirical translation between accelerator-only and comprehensive scopes (narrow/comprehensive

≈ 0.417) enables direct, auditable comparisons across systems and papers; interventions that appear large at the chip boundary can attenuate—or reverse—once host, idle, and facility overheads are included [1].

Operationally, a *single* SLO-aware policy achieves large, consistent reductions without relaxing interactivity. Across four models, median per-prompt impacts fall by roughly 57–59% (energy), 59–60% (consumptive water; site + source), and 78–80% (location-based CO_2), with $p95$ TTFT/TPOT and capacity constraints met in every five-minute window. For a representative 500 M-query day on GPT-4o, totals drop from 0.344 → 0.145 GWh, 1.196 → 0.490 ML, and 121 → 25 t CO_2 (LB). These gains arise from complementary levers—batch right-sizing, concise token directives, carbon- and water-aware geo-routing (via CIF and EWIF), and phase-aware prefill/decode placement—rather than from any single mechanism.

A central conceptual contribution is to treat *water* as a first-class objective, computed as site cooling scaled by PUE plus source-side electricity–water intensity, and to optimize it jointly with carbon [3]. Because CIF and EWIF are only weakly correlated, optimizing one can worsen the other. The framework's dual-objective design and the "ray" geometry in the (g CO_2, mL) plane make these trade-offs explicit: routing alone traces a Pareto edge determined by site factors, while adding batch and token controls translates the entire feasible cloud toward the origin, compounding siting with operational efficiency (Figures 1–2 and 3–4).

Choosing medians over means for skewed mixes matches production practice and provides a stable basis for policy comparison. Table 2 makes scope reconciliation explicit by showing the $\sim 2.4\times$ uplift from accelerator-only to comprehensive across models; pairing comprehensive medians with scope "whiskers" avoids ad hoc conversion factors and helps readers reconcile results across boundary choices [1].

From a deployment standpoint, three near-term steps follow. **(i)** Integrate carbon- and water-aware geo-routing into global load balancers, enforcing $p95$ latency and using realistic tokens-per-second quantiles. **(ii)** Apply concise generation directives to curb unnecessary tokens, especially when combined with higher off-peak batching. **(iii)** Use phase-aware placement—fast, compute-efficient cohorts for prefill and memory-efficient or second-life cohorts for decode—to extend hardware life and bring embodied impacts into scope without sacrificing responsiveness [16,22].

Limitations point directly to future work. Hourly variability in PUE, site WUE, and electricity-mix water intensities suggests moving from annual to time-resolved site and grid factors to sharpen routing signals. Market-based carbon is treated here as a sensitivity; incorporating temporal matching and procurement constraints (e.g., REC/PPA portfolios) would enable joint LB/MB evaluations [25]. Water could be weighted by basin-level scarcity to reflect environmental equity [3]. Finally, richer user-experience models (percentile bands by profile/region) and live-traffic experiments would strengthen external validity; deeper lifecycle modeling (refurbishment yields, end-of-life pathways) would close the loop from serving policy to circularity outcomes [16].

6. Conclusions

This work delivers an operational template for reducing both the carbon and water footprints of LLM serving without compromising interactive quality. By adopting a comprehensive serving boundary, summarizing impacts with daily medians, and co-optimizing carbon and water under explicit $p95$ latency and capacity constraints, the framework turns a fragmented literature into a deployable control policy with repeatable gains.

Across four models, the SLO-aware policy cuts comprehensive-boundary medians by about 58% in energy, about 59% in consumptive water, and about 79% in location-based CO_2. For a representative day with 500 M GPT-4o queries, totals fall from 0.344 to 0.145 GWh, from 1.196 to 0.490 ML, and from 121 to 25 t CO_2 (LB), with $p95$ SLOs satisfied in every five-minute window. These reductions stem from coordinated levers—carbon- and water-aware routing, batch right-sizing, concise token directives, and phase-aware assignment of prefill and decode—rather than from a single intervention.

The scope reconciliation module reproduces the production-observed narrow/comprehensive ≈ 0.417 ratio, which enables apples-to-apples comparison between chip-only and full-stack accounting. Pareto views make the carbon–water geometry explicit and give operators a practical way to navigate trade-offs at fixed service levels. Taken together, these elements support industry adoption, transparent reporting, and continuous improvement as grids, cooling systems, and inference stacks evolve.

Abbreviations

AI	Artificial Intelligence.
API	Application Programming Interface (used when referring to cross-model API benchmarks).
CIF	Carbon Intensity of the Grid (typically kg CO_2 kWh^{-1}); used for location-based emissions.
CO_2	Carbon dioxide; operational emissions are reported in grams per prompt and in tons per day.
CO_2e	Carbon-dioxide equivalent (used when referring to greenhouse-gas accounting).
CPU	Central Processing Unit (host side of serving stack).
DRAM	Dynamic Random-Access Memory (host memory included in comprehensive boundary).
EF_{MB}	Market-Based portfolio emission factor (kg CO_2e kWh^{-1}) used as a sensitivity to LB.
EWIF	Electricity–Water Intensity Factor (L kWh^{-1}) capturing off-site, generation-mix water.
$EWIF_{source}$	"Source" component of water from electricity generation in the site+source accounting.
GHG	Greenhouse Gas.
GPU	Graphics Processing Unit (accelerator).
GWh	Gigawatt-hour (10^9 Wh).
IT	Information Technology load (accelerators + host CPU/DRAM + provisioned idle).
kWh	Kilowatt-hour (10^3 Wh).
KV cache	Key–Value cache (used in decode optimizations).
LB	Location-Based (grid-average, point-of-consumption reporting for emissions; the default in this work).
LBNL	Lawrence Berkeley National Laboratory (source for PUE/WUE context).
LLM	Large Language Model.
MB	Market-Based (portfolio accounting sensitivity for emissions).
MILP	Mixed-Integer Linear Program (optimization formulation).
mL	Milliliter (10^{-3} L).
ML	Megaliter (10^6 L); in results tables, ML day^{-1} is used for daily totals.
s	Second.
PUE	Power Usage Effectiveness (facility/IT energy ratio).
QoS	Quality of Service (used when discussing interactive service constraints).
SLO	Service Level Objective (latency/throughput targets enforced in the optimizer).
Σ-Scale	The time-resolved, SLO-aware bi-objective orchestration loop proposed in the paper.
TPOT	Time Per Output Token (latency metric for decode).
TPS	Tokens Per Second (throughput metric used in capacity constraints).
TPU	Tensor Processing Unit (accelerator).
TTFT	Time To First Token (latency metric for prefill).
Wh	Watt-hour (unit for per-prompt energy).
WUE	Water Usage Effectiveness (L kWh^{-1} at the facility; site cooling).
WUE_{site}	Site-level WUE used in the site+source water formulation.
$p95$	95th-percentile statistic (used for latency and throughput SLO enforcement).

References

1. Elsworth, C.; Huang, K.; Patterson, D.; Schneider, I.; Sedivy, R.; Goodman, S.; Manyika, J. Measuring the environmental impact of delivering AI at Google Scale, 2025, [arXiv:cs.DC/2508.15734].
2. Huang, Y. Advancing industrial sustainability research: A domain-specific large language model perspective. *Clean Technologies and Environmental Policy* **2025**, *27*, 1899–1901.
3. Li, S. Making AI less "thirsty": Uncovering and addressing the secret water footprint of AI models, 2023, [2304.03271].

4. Desislavov, R.; Martínez-Plumed, F.; Hernández-Orallo, J. Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems* **2023**, *38*, 100857.
5. Jegham, N.; Abdelatti, M.; Elmoubarki, L.; Hendawi, A. How hungry is AI? Benchmarking energy, water, and carbon footprint of LLM inference, 2025, [2505.09598].
6. Jagannadharao, A.; Beckage, N.; Nafus, D.; Chamberlin, S. Time shifting strategies for carbon-efficient long-running large language model training. *Innovations in Systems and Software Engineering* **2025**, *21*, 517–531.
7. Naveed, H.; Khan, A.U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Mian, A. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology* **2025**, *16*, 1–72.
8. Husom, E.J.; Goknil, A.; Shar, L.K.; Sen, S. The price of prompting: Profiling energy use in large language models inference, 2024, [2407.16893].
9. Moore, H.; Qi, S.; Hogade, N.; Milojicic, D.; Bash, C.; Pasricha, S. Sustainable Carbon-Aware and Water-Efficient LLM Scheduling in Geo-Distributed Cloud Datacenters, 2025, [2505.23554].
10. Chien, A.A.; Lin, L.; Nguyen, H.; Rao, V.; Sharma, T.; Wijayawardana, R. Reducing the Carbon Impact of Generative AI Inference (today and in 2035). In Proceedings of the Proceedings of the 2nd Workshop on Sustainable Computer Systems, 2023, pp. 1–7.
11. Argerich, M.F.; Patiño-Martínez, M. Measuring and improving the energy efficiency of large language models inference. *IEEE Access* **2024**, *12*, 80194–80207.
12. De Vries, A. The growing energy footprint of artificial intelligence. *Joule* **2023**, *7*, 2191–2194.
13. Luccioni, A.S.; Viguier, S.; Ligozat, A.L. Estimating the carbon footprint of BLOOM, a 176B parameter language model. *Journal of Machine Learning Research* **2023**, *24*, 1–15.
14. Jiang, Y.; Roy, R.B.; Kanakagiri, R.; Tiwari, D. WaterWise: Co-optimizing Carbon-and Water-Footprint Toward Environmentally Sustainable Cloud Computing. In Proceedings of the PPoPP '25: 30th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming, 2025, pp. 297–311.
15. Islam, M.A.; Ren, S.; Quan, G.; Shakir, M.Z.; Vasilakos, A.V. Water-constrained geographic load balancing in data centers. *IEEE Transactions on Cloud Computing* **2015**, *5*, 208–220.
16. Schneider, I.; Xu, H.; Benecke, S.; Patterson, D.; Huang, K.; Ranganathan, P.; Elsworth, C. Life-cycle emissions of AI hardware: A cradle-to-grave approach and generational trends, 2025, [2502.01671].
17. Wu, Y.; Hua, I.; Ding, Y. Unveiling environmental impacts of large language model serving: A functional unit view, 2025, [2502.11256].
18. Cheng, K.; Wang, Z.; Hu, W.; Yang, T.; Li, J.; Zhang, S. SCOOT: SLO-Oriented Performance Tuning for LLM Inference Engines. In Proceedings of the Proceedings of The Web Conference 2025, 2025, pp. 829–839.
19. Wu, C.J.; Raghavendra, R.; Gupta, U.; Acun, B.; Ardalani, N.; Maeng, K.; Hazelwood, K.; et al. Sustainable AI: Environmental implications, challenges and opportunities. In Proceedings of the Proceedings of Machine Learning and Systems, 2022, Vol. 4, pp. 795–813.
20. Samsi, S.; Zhao, D.; McDonald, J.; Li, B.; Michaleas, A.; Jones, M.; Gadepally, V.; et al. From words to watts: Benchmarking the energy costs of large language model inference. In Proceedings of the 2023 IEEE High Performance Extreme Computing Conference (HPEC), 2023, pp. 1–9.
21. Wiesner, P.; Grinwald, D.; Weiß, P.; Wilhelm, P.; Khalili, R.; Kao, O. Carbon-Aware Quality Adaptation for Energy-Intensive Services. In Proceedings of the Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems, 2025, pp. 415–422.
22. Nguyen, S.; Zhou, B.; Ding, Y.; Liu, S. Towards sustainable large language model serving. *ACM SIGENERGY Energy Informatics Review* **2024**, *4*, 134–140.
23. Falk, S.; Ekchajzer, D.; Pirson, T.; Lees-Perasso, E.; Wattiez, A.; Biber-Freudenberger, L.; van Wynsberghe, A. More than Carbon: Cradle-to-Grave environmental impacts of GenAI training on the Nvidia A100 GPU, 2025, [2509.00093].
24. Mistral AI. Our contribution to a global environmental standard for AI. https://mistral.ai/news/ourcontribution-to-a-global-environmental-standard-for-ai, 2025.
25. Soares, I.V.; Yarime, M.; Klemun, M.M. Estimating GHG emissions from cloud computing: sources of inaccuracy, opportunities and challenges in location-based and use-based approaches. *Climate Policy* **2025**, pp. 1–19.
26. Anquetin, T.; Coqueret, G.; Tavin, B.; Welgryn, L. Scopes of carbon emissions and their impact on green portfolios. *Economic Modelling* **2022**, *115*, 105951.
27. Różycki, R.; Solarska, D.A.; Waligóra, G. Energy-Aware Machine Learning Models—A Review of Recent Techniques and Perspectives. *Energies* **2025**, *18*, 2810.

28. Fu, Z.; Chen, F.; Zhou, S.; Li, H.; Jiang, L. LLMCO2: Advancing accurate carbon footprint prediction for LLM inferences. *ACM SIGENERGY Energy Informatics Review* **2025**, *5*, 63–68.
29. Daraghmeh, H.M.; Wang, C.C. A review of current status of free cooling in datacenters. *Applied Thermal Engineering* **2017**, *114*, 1224–1239.
30. Ebrahimi, K.; Jones, G.F.; Fleischer, A.S. A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities. *Renewable and Sustainable Energy Reviews* **2014**, *31*, 622–638.
31. Mytton, D. Data centre water consumption. *npj Clean Water* **2021**, *4*.
32. Sharma, N.; Mahapatra, S.S. A preliminary analysis of increase in water use with carbon capture and storage for Indian coal-fired power plants. *Environmental Technology & Innovation* **2018**, *9*, 51–62.
33. Chlela, S.; Selosse, S. Water use in a sustainable net zero energy system: what are the implications of employing bioenergy with carbon capture and storage? *International Journal of Sustainable Energy Planning and Management* **2024**, *40*, 146–162.
34. Chung, J.W.; Liu, J.; Ma, J.J.; Wu, R.; Kweon, O.J.; Xia, Y.; Chowdhury, M.; et al. The ML.ENERGY Benchmark: Toward Automated Inference Energy Measurement and Optimization, 2025, [2505.06371].
35. Luccioni, S.; Gamazaychikov, B. AI energy score leaderboard. https://huggingface.co/spaces/AIEnergyScore/Leaderboard, 2025.
36. Sarkar, S.; Naug, A.; Luna, R.; Guillen, A.; Gundecha, V.; Ghorbanpour, S.; Babu, A.R. Carbon footprint reduction for sustainable data centers in real-time. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 22322–22330.
37. Mondal, S.; Faruk, F.B.; Rajbongshi, D.; Efaz, M.M.K.; Islam, M.M. GEECO: Green data centers for energy optimization and carbon footprint reduction. *Sustainability* **2023**, *15*, 15249.
38. Riepin, I.; Brown, T.; Zavala, V.M. Spatio-temporal load shifting for truly clean computing. *Advances in Applied Energy* **2025**, *17*, 100202.
39. Rahman, A.; Liu, X.; Kong, F. A survey on geographic load balancing based data center power management in the smart grid environment. *IEEE Communications Surveys & Tutorials* **2013**, *16*, 214–233.
40. Wiesner, P.; Behnke, I.; Scheinert, D.; Gontarska, K.; Thamsen, L. Let's wait awhile: How temporal workload shifting can reduce carbon emissions in the cloud. In Proceedings of the Proceedings of the 22nd International Middleware Conference, 2021, pp. 260–272.
41. Silva, C.A.; Vilaça, R.; Pereira, A.; Bessa, R.J. A review on the decarbonization of high-performance computing centers. *Renewable and Sustainable Energy Reviews* **2024**, *189*, 114019.
42. Radovanović, A.; Koningstein, R.; Schneider, I.; Chen, B.; Duarte, A.; Roy, B.; Cirne, W. Carbon-aware computing for datacenters. *IEEE Transactions on Power Systems* **2022**, *38*, 1270–1280.
43. Faiz, A.; Kaneda, S.; Wang, R.; Osi, R.; Sharma, P.; Chen, F.; Jiang, L. LLMCarbon: Modeling the end-to-end carbon footprint of large language models, 2023, [2309.14393].
44. Patel, P.; Choukse, E.; Zhang, C.; Shah, A.; Goiri, Í.; Maleki, S.; Bianchini, R. Splitwise: Efficient Generative LLM Inference Using Phase Splitting. In Proceedings of the 2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA), 2024, pp. 118–132.
45. Fan, H.; Lin, Y.C.; Prasanna, V. ELLIE: Energy-Efficient LLM Inference at the Edge Via Prefill-Decode Splitting. In Proceedings of the 2025 IEEE 36th International Conference on Application-specific Systems, Architectures and Processors (ASAP), 2025, pp. 139–146.
46. Zhu, K.; Gao, Y.; Zhao, Y.; Zhao, L.; Zuo, G.; Gu, Y.; Kasikci, B. NanoFlow: Towards Optimal Large Language Model Serving Throughput. In Proceedings of the 19th USENIX Symposium on Operating Systems Design and Implementation (OSDI 25), 2025, pp. 749–765.
47. Zhong, Y.; Liu, S.; Chen, J.; Hu, J.; Zhu, Y.; Liu, X.; Zhang, H.; et al. DistServe: Disaggregating prefill and decoding for goodput-optimized large language model serving. In Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24), 2024, pp. 193–210.
48. Feng, J.; Huang, Y.; Zhang, R.; Liang, S.; Yan, M.; Wu, J. WindServe: Efficient Phase-Disaggregated LLM Serving with Stream-based Dynamic Scheduling. In Proceedings of the Proceedings of the 52nd Annual International Symposium on Computer Architecture, 2025, pp. 1283–1295.
49. Svirschevski, R.; May, A.; Chen, Z.; Chen, B.; Jia, Z.; Ryabinin, M. Specexec: Massively parallel speculative decoding for interactive LLM inference on consumer devices. *Advances in Neural Information Processing Systems* **2024**, *37*, 16342–16368.
50. Liu, A.; Liu, J.; Pan, Z.; He, Y.; Haffari, G.; Zhuang, B. MiniCache: KV cache compression in depth dimension for large language models. *Advances in Neural Information Processing Systems* **2024**, *37*, 139997–140031.

doi:10.20944/preprints202510.0957.v1

51. Jiang, Y.; Roy, R.B.; Li, B.; Tiwari, D. Ecolife: Carbon-aware serverless function scheduling for sustainable computing. In Proceedings of the SC24: International Conference for High Performance Computing, Networking, Storage and Analysis, 2024, pp. 1–15.
52. Li, B.; Jiang, Y.; Gadepally, V.; Tiwari, D. SPROUT: Green generative AI with carbon-efficient LLM inference. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 21799–21813.
53. Jiang, P.; Sonne, C.; Li, W.; You, F.; You, S. Preventing the immense increase in the life-cycle energy and carbon footprints of LLM-powered intelligent chatbots. *Engineering* **2024**, *40*, 202–210. https://doi.org/10.1016/j.eng.2024.04.002.
54. Morsy, M.; Znid, F.; Farraj, A. A critical review on improving and moving beyond the 2 nm horizon: Future directions and impacts in next-generation integrated circuit technologies. *Materials Science in Semiconductor Processing* **2025**, *190*, 109376.
55. Wang, P.; Zhang, L.Y.; Tzachor, A.; Chen, W.Q. E-waste challenges of generative artificial intelligence. *Nature Computational Science* **2024**, *4*, 818–823.
56. Shehabi, A.; Smith, S.J.; Hubbard, A.; Newkirk, A.; Lei, N.; Siddik, M.A.B.; Holecek, B.; Koomey, J.G.; Masanet, E.; Sartor, D.A. 2024 United States Data Center Energy Usage Report (LBNL-2001637). Technical report, Lawrence Berkeley National Laboratory, 2024. https://doi.org/10.71468/P1WC7Q.
57. Li, P.; Yang, J.; Wierman, A.; Ren, S. Towards environmentally equitable AI via geographical load balancing. In Proceedings of the Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems, 2024, pp. 291–307.
58. Cao, Z.; Zhou, X.; Hu, H.; Wang, Z.; Wen, Y. Toward a systematic survey for carbon neutral data centers. *IEEE Communications Surveys & Tutorials* **2022**, *24*, 895–936.
59. Islam, M.A.; Mahmud, H.; Ren, S.; Wang, X. A carbon-aware incentive mechanism for greening colocation data centers. *IEEE Transactions on Cloud Computing* **2017**, *8*, 4–16.
60. Kim, H.; Young, S.; Chen, X.; Gupta, U.; Hester, J. Slower is Greener: Acceptance of Eco-feedback Interventions on Carbon Heavy Internet Services. *ACM Journal on Computing and Sustainable Societies* **2025**, *3*, 1–21.
61. Wang, Y.; Chen, K.; Tan, H.; Guo, K. Tabi: An efficient multi-level inference system for large language models. In Proceedings of the Proceedings of the Eighteenth European Conference on Computer Systems, 2023, pp. 233–248.
62. Ahmadpanah, S.H.; Sobhanloo, S.; Afsharfarnia, P. Dynamic token pruning for LLMs: leveraging task-specific attention and adaptive thresholds. *Knowledge and Information Systems* **2025**, pp. 1–20.
63. Belhaouari, S.B.; Kraidia, I. Efficient self-attention with smart pruning for sustainable large language models. *Scientific Reports* **2025**, *15*, 10171.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.