BIRDS: Characterizing and Understanding Biodiversity Impact of Large Language Model Serving

Anonymous ACL submission

Abstract

Large language model (LLM) serving creates environmental impacts beyond carbon and water, including ecosystem damage through biodiversity-related pathways. We present BIRDS, a framework for Biodiversity Impact of Request-Driven LLM Serving. BIRDS defines request-level functional units, quantifies operational and embodied biodiversity impact, and introduces Quality-Normalized Biodiversity Impact (QNBI) to jointly analyze ecological impact and response quality. Across diverse workloads, models, GPUs, and regions, BIRDS reveals that biodiversity impact accumulates at scale and exposes actionable quality-aware serving tradeoffs. The code is available at `https://anonymous.4open.science/r/BIRDS-F44D`.

1 Introduction

Large language models (LLMs) are rapidly becoming critical computing infrastructure, but their growing deployment has also raised concerns about environmental impact (Ding and Shi, 2024). Recent work has begun to characterize the carbon emissions (Strubell et al., 2019; Li et al., 2024; Wu et al., 2025b) and water consumption (Li et al., 2025; Ren et al., 2024) of LLMs. However, environmental impact extends beyond carbon or water alone. Electricity generation and semiconductor manufacturing can also release air pollutants and chemical waste that contribute to acidification, eutrophication, and ecotoxicity (Blum, 2024; UNEP, 2024). The ecological stakes are substantial: around 1 million animal and plant species are threatened with extinction and industrial facilities discharge 300–400 million tons of heavy metals, solvents, toxic sludge, and other wastes into waters each year (IPBES, 2019). Such stressors can damage ecosystems by degrading soil quality, disrupting habitats, and reducing species survival over time (Falk et al., 2025; Shi et al., 2025).

To provide a more comprehensive understanding of the environmental impact of LLM serving, this paper focuses on biodiversity impact (BI) characterization. BI measures ecosystem damage induced by human activities through multiple environmental pathways (Cardinale et al., 2012). Importantly, optimizing LLM serving for carbon or water alone does not necessarily minimize BI. A region with low carbon intensity may still induce high ecological damage through air-pollution-related pathways (Shi et al., 2025). As a result, biodiversity-aware serving decisions can differ from carbon-aware or water-aware optimization. Although the BI of a single request is small, modern LLM systems process trillions of tokens per day (Pichai, 2026; OpenRouter, 2026) causing ecological impacts to accumulate rapidly at deployment scale.

While recent work has modeled BI for computing (Falk et al., 2025; Shi et al., 2025), existing approaches cannot directly apply to LLM serving for two reasons. ❶ They lack a request-driven modeling framework tailored to LLM serving workloads. LLM serving fundamentally operates on requests with serving conditions and latency constraints, yet no serving-oriented functional unit (definition in §2.1) has been defined to attribute BI at the request level. ❷ Existing approaches lack mechanisms to reason about the tradeoffs between BI and LLM output quality. Smaller models may appear environmentally preferable while producing lower-quality outputs that require retries or corrections.

To address these limitations, we present BIRDS, a modeling framework for **B**iodiversity **I**mpact of **R**equest-**D**riven LLM **S**erving. The key insight behind BIRDS is that biodiversity accounting for LLM serving should move beyond coarse infrastructure-level estimation into request-driven analysis that explicitly accounts for LLM output quality. In particular, environmentally efficient LLM serving should consider not only the ecosystem damage from generating responses, but also

Figure 1: Conceptual overview of how LLM serving activities contribute to biodiversity impact.

whether those responses successfully accomplish the target task. BIRDS consists of three main steps. First, BIRDS defines a request-driven functional unit (FU) that models LLM serving under explicit workloads, serving configurations, and latency constraints. Second, BIRDS quantifies FU-level BI by jointly modeling operational and embodied lifecycle impacts through midpoint-to-endpoint biodiversity accounting. Third, BIRDS enables quality-aware biodiversity analysis through a new metric, Quality-Normalized Biodiversity Impact (QNBI), which jointly evaluates BI and response quality across models and serving configurations.

We evaluate BIRDS across diverse LLM serving workloads, model families, and GPU platforms. Our results show that per-request BI is numerically small, but can accumulate rapidly under trillion-token-scale inference traffic; operational impact dominates total BI of LLM serving, making serving efficiency a primary lever for mitigation; BI-aware deployment can select different regions than carbon- or water-aware deployment; and quality-aware analysis reveals that the lowest-impact model is not always the most ecologically efficient choice. In particular, intermediate-scale dense models and sparse MoE models often achieve better QNBI than very small or very large models, though extremely large MoE deployments can lose this advantage due to multi-GPU overhead. Long-output workloads, reasoning modes and older GPUs can substantially increase BI when their quality or throughput gains do not offset additional serving overhead.

We make the following contributions:

- Demonstrate biodiversity impact as a complementary ecological metric beyond carbon and water for LLM serving.
- Present BIRDS, a quality-aware, request-driven framework for modeling BI of LLM serving.
- Define QNBI to jointly compare BI and response quality across models and serving configurations.
- Characterize BI across diverse workloads, model families, GPUs, and deployment regions, revealing actionable serving tradeoffs.

2 Background and Related Work

2.1 Life Cycle Assessment

Life cycle assessment (LCA) is a widely used methodology for quantifying the environmental impact induced by products and systems throughout their lifecycle (Finkbeiner et al., 2006). LCA evaluates environmental stressors through **midpoint indicators**, such as acidification, eutrophication, and ecotoxicity, which can be further translated into **endpoint** ecosystem damage metrics. This endpoint damage is expressed in **species·yr**, which estimates ecosystem degradation in terms of species loss integrated over time. As illustrated in Figure 1, LLM serving contributes to biodiversity impact indirectly through multiple environmental pathways.

LCA studies commonly distinguish between operational impact and embodied impact (Menzies et al., 2007). **Operational impact** refers to the impact induced during system operation, such as electricity consumption during LLM serving. **Embodied impact** captures the lifecycle impact associated with hardware manufacturing, transportation, and end-of-life processes. To enable fair comparison across systems and operating conditions, LCA analyses further rely on a **functional unit** (Klöpffer and Grahl, 2014), which defines the normalized unit of analysis for comparing environmental impacts across systems or operating conditions.

2.2 Environmental Impact of LLM Serving

Recent work has begun to characterize the environmental impact of LLMs focusing on carbon emissions (Strubell et al., 2019; Wu et al., 2025b; Li et al., 2024; Faiz et al., 2024) and water consumption (Li et al., 2025; Ren et al., 2024; Wu et al., 2025a). However, these analyses do not capture broader biodiversity-related ecological impacts.

Limited recent work has modeled BI for computing systems (Falk et al., 2025; Shi et al., 2025), focusing on device- and infrastructure-level analyses. However, these approaches cannot directly apply to LLM serving for two reasons. ❶ They lack request-driven functional units needed to characterize serving workloads under serving constraints. ❷ They lack quality-aware metrics needed to analyze tradeoffs between BI and LLM output quality. In

contrast, BIRDS moves BI accounting from coarse infrastructure-level estimation into request-driven, quality-aware LLM serving analysis.

3 The BIRDS Framework

This section presents BIRDS, a modeling framework for **B**iodiversity **I**mpact of **R**equest-**D**riven LLM **S**erving. The key insight behind BIRDS is that BI accounting for LLM serving should move beyond coarse infrastructure-level estimation into request-driven and quality-aware analysis. As illustrated in Figure 2, BIRDS consists of three steps. First, BIRDS establishes a request-driven functional unit (FU) for LLM serving workloads under serving operating points and latency constraints. Second, BIRDS performs BI accounting by modeling both operational and embodied impacts associated with each served FU. Third, BIRDS enables quality-aware biodiversity analysis across LLM serving configurations and model accuracy.

3.1 Step 1: Establish a Functional Unit

Because LLM serving is request-based and constrained by throughput and latency targets, BIRDS models BI at the request level to enable fair comparison across serving operating points.

Functional Unit Definition. Assume we model the BI of an LLM serving workload type w executed by a serving instance i in deployment region r. A workload type denotes a class of LLM tasks (e.g., summarization or code completion) with similar prompt/response length distribution and service level objective (SLO) with respect to latency. A serving instance denotes the complete serving configuration used to host the model, including the GPU set and parallelism strategy.

Following LCA practice, BIRDS defines the request-driven functional unit (FU) as one completed request of workload type w, served by instance i at achieved throughput λ while satisfying latency constraints in terms of Time-to-First-Token (TTFT) and Time-per-Output-Token (TPOT). We denote a serving operating point as $\theta = (w, i, \lambda)$. Here is an example of request-driven FU definition:

> One completed summarization request served by an instance at $\lambda = 12$ requests/s, while satisfying TTFT of 2s and TPOT of 150 ms.

3.2 Step 2: Quantify FU-Level BI

Following LCA practices (§2.1), BIRDS models BI per FU as the sum of operational BI (OBI) and embodied BI (EBI). Let $\theta = (w, i, \lambda)$ denote an SLO-satisfying serving operating point:

$$\mathrm{BI}_{\mathrm{fu}}(\theta, r) = \mathrm{OBI}_{\mathrm{fu}}(\theta, r) + \mathrm{EBI}_{\mathrm{fu}}(\theta). \quad (1)$$

$\mathrm{OBI}_{\mathrm{fu}}$ captures BI from serving electricity consumption, while $\mathrm{EBI}_{\mathrm{fu}}$ captures the amortized lifecycle BI of serving hardware.

BIRDS adopts midpoint-to-endpoint BI accounting (Huijbregts et al., 2016). Midpoint indicators capture ecological stressors induced by activities. Endpoint indicator translates these stressors into ecosystem damage using ecological characterization factors. This formulation enables BIRDS to model BI through multiple pathways instead of relying on a single ecological indicator:

$$\mathrm{BI} = \sum_{c \in \mathcal{C}} M_c \Phi_c(T), \quad M_c = \sum_{q \in \mathcal{Q}} X_q I_{q,c}(r), \quad (2)$$

where M_c denotes the midpoint impact for category c, and $\Phi_c(T)$ converts midpoint impacts into endpoint ecosystem damage by integrating on future time horizon T; q denotes an activity or flow, X_q is the attributed amount of that activity, and $I_{q,c}(r)$ is its midpoint intensity for category c. For example, for the global warming (GW) midpoint of operational electricity, X_q is the energy and $I_{q,\mathrm{GW}}(r)$ is the regional carbon intensity in kg CO_2eq/kWh.

OBI and EBI Instantiation. For OBI, BIRDS measures energy consumption during steady-state serving and attributes it to completed FUs:

$$E_{\mathrm{fu}}(\theta) = \frac{E(\theta)}{N(\theta)} = \frac{\bar{P}(\theta)}{\lambda}, \quad (3)$$

where $E(\theta)$ is the measured energy, $N(\theta)$ is the number of completed FUs, $\bar{P}(\theta)$ is the average power and λ is the achieved throughput. BIRDS then converts serving energy into OBI using region-specific grid ecological intensity factors $I_{\mathrm{grid},c}(r)$:

$$\mathrm{OBI}_{\mathrm{fu}}(\theta, r) = E_{\mathrm{fu}}(\theta) \cdot \sum_{c \in \mathcal{C}} I_{\mathrm{grid},c}(r) \Phi_c(T). \quad (4)$$

For EBI, BIRDS amortizes the lifecycle EBI of the serving instance over the hardware lifetime:

$$\mathrm{EBI}_{\mathrm{fu}}(\theta) = \frac{\mathrm{EBI}_{\mathrm{sec}}(i)}{\lambda}, \quad \mathrm{EBI}_{\mathrm{sec}}(i) = \frac{\mathrm{EBI}(i)}{\mathrm{LT}}. \quad (5)$$

Here, $\mathrm{EBI}_{\mathrm{sec}}(i)$ and $\mathrm{EBI}(i)$ are the per-second and lifetime EBI of the hardware, respectively. LT is the assumed lifetime. More details of hardware-specific EBI modeling are in §B.2.

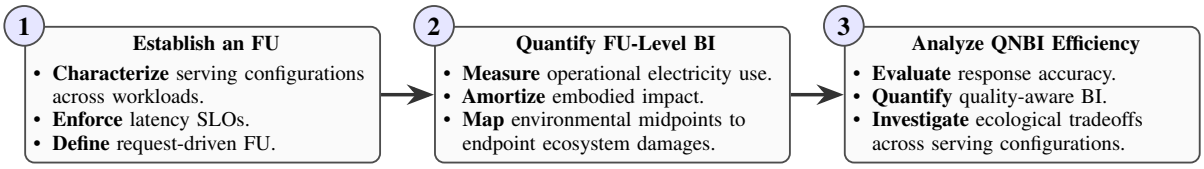

Figure 2: Overview of the three-step modeling procedure of BIRDS.

3.3 Step 3: Analyze Quality-Aware Efficiency

While $\mathrm{BI}_{\mathrm{fu}}$ quantifies the BI of serving one FU, it does not indicate whether the generated response successfully accomplishes the target task. In particular, smaller models may appear environmentally preferable per FU but could produce lower-quality outputs that require retries or human correction.

For quality-aware serving analysis, BIRDS normalizes $\mathrm{BI}_{\mathrm{fu}}$ by output quality to provide a fair basis of evaluation across models. Based on this intuition, we introduce Quality-Normalized Biodiversity Impact (QNBI). Let $Q(\theta) \in [0, 1]$ be the average response quality under serving operating point θ, measured using task-specific evaluation metrics such as exact-match accuracy, pass rate, execution success, or LLM-judge score. We have

$$\mathrm{QNBI}(\theta, r) = \frac{\mathrm{BI}_{\mathrm{fu}}(\theta, r)}{\max(Q(\theta), \epsilon)}, \qquad (6)$$

where $\epsilon > 0$ is a small constant used to avoid numerical instability when $Q(\theta)$ approaches zero. A lower QNBI indicates that less BI is expected to be required to obtain a successful response.

4 Evaluation

We evaluate BIRDS through a comprehensive study of BI across diverse LLM serving workloads, models, and serving hardware. We focus on four research questions: **1** What is the overall BI of serving LLM inference requests? **2** Where do BIs originate across operational and embodied lifecycle stages? **3** How does BI relate to or differ from existing environmental metrics such as carbon and water footprints? **4** What tradeoffs exist between LLM serving configurations, output quality, and BI? We first describe the evaluation methodology and then present results for each research question.

4.1 Evaluation Methodology

Workloads. We select diverse non-agentic serving workloads where requests are self-contained and do not invoke external tools during inference. The evaluated datasets cover everyday text conversation ((ShareGPT) and WildChat (Zhao et al., 2024)), knowledge-intensive reasoning (MMLU-Pro (Wang et al., 2024) and SuperGPQA-Hard (Team et al., 2025)), IDE-style code completion (CrossCodeEval (Ding et al., 2023) and RepoBench (Liu et al., 2024)), and long-context NLP tasks (LongBench (Bai et al., 2023)). This workload selection enables us to study how different prompt and response length distributions affect latency, energy consumption, and BI. Detailed dataset descriptions are in Table 4 in §C.

Models. To study BI tradeoffs across model capability and scale, we evaluate representative open LLM families, including Llama-2 and 3 (Touvron et al., 2023; Grattafiori et al., 2024), GPT-OSS (OpenAI, 2025), Qwen3 (Qwen Team, 2025), and Gemma-4 (Google DeepMind, 2026). These models span 0.6B–235B parameters and include both dense and mixture-of-experts (MoE) architectures. This selection enables direct comparison between smaller lower-impact models and larger higher-capability models. Full model variants and compatibility filters are listed in Table 5 in §C.

GPUs. We run inference on three datacenter NVIDIA GPUs: L40 (NVIDIA, 2022b), A100 (NVIDIA, 2020), and H100 (NVIDIA, 2022a). These GPUs differ in memory capacity, bandwidth, and power characteristics, enabling us to study how hardware generation changes throughput, energy consumption, and BI. Detailed GPU specifications are provided in Table 6 in §C.

Metrics. We measure achieved throughput, TTFT, TPOT, and GPU power for each workload–model–GPU configuration. We then compute energy consumption and $\mathrm{BI}_{\mathrm{fu}}$ using the modeling in §3.2 with the latest available annual average grid midpoint intensities of the U.S., unless otherwise specified. We compute QNBI where response quality is evaluated using task-specific protocols. Open-ended chat workloads are evaluated by an LLM judge against responses from a reference model, selected as Llama-3.1-8B to provide a balanced comparison basis of model size and capability. Objective benchmarks use their native accuracy metrics. Detailed prompts, scoring rules, and quality

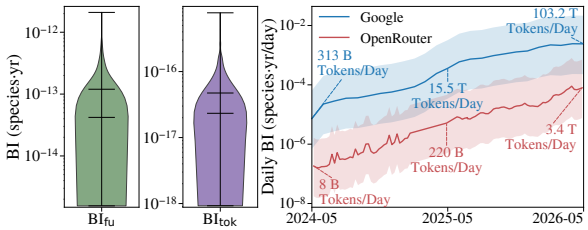

Figure 3: Distribution of FU- and token-level BIs (BI_{fu} and BI_{tok}) across evaluated LLM serving configurations, and daily estimated aggregate BI under publicly reported token traffic from Google and OpenRouter between May 2024 and May 2026. Shaded regions denote 95% confidence intervals.

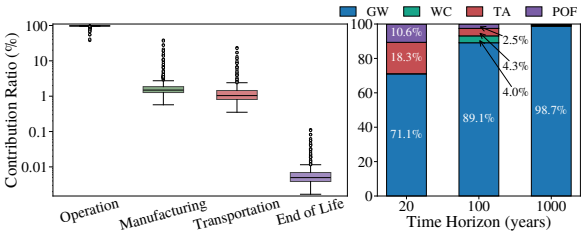

Figure 4: BI_{fu} by lifecycle stage integrated over 100 years and by midpoint-to-endpoint contribution ratios under different time horizons. Global warming (GW), water consumption (WC), terrestrial acidification (TA), and photochemical ozone formation (POF) are the top-4 midpoint pathways, with the other midpoints contributing less than 0.01%.

scores are provided in §E.

Profiling. To obtain realistic measurements, we use vLLM 0.19.1 (Kwon et al., 2023) as the serving engine and collect GPU power traces through NVML (NVIDIA Corporation, 2025). For each serving configuration, we sweep incoming request rates and retain only stable operating points that satisfy the SLO constraints in §C. Unless otherwise specified, we report results at *maximum sustainable throughput (MST)*, defined as the highest stable SLO-satisfying throughput. MST amortizes serving overheads across the largest number of completed requests, making it the most energy-efficient satisfactory operating point. Detailed energy profiling methodology is provided in §D.2.

4.2 Overall BI Characterization

We first characterize the overall BI of LLM serving by analyzing both per-request impact and its accumulation under large-scale inference traffic.

FU- and token-level BIs remain low in magnitude. Figure 3 (left) shows the distribution of FU- and token-level BIs across profiled serving configurations. At the scale of a single request or token, these values are numerically small, typically corresponding to very small amounts of ecosystem damage. However, modern LLM systems process massive token volumes, causing these impacts to accumulate rapidly at deployment scale.

Aggregate BI grows with token traffic. To estimate deployment-scale impact magnitude, we multiply the token-level BI distribution by publicly reported token traffic from Google (Pichai, 2026) and OpenRouter (OpenRouter, 2026). Although these traces do not represent the entire LLM ecosystem, they provide observable lower bounds on aggregate inference demand. As shown in Figure 3 (right), reported traffic increased from 313B to 103.2T tokens/day for the larger trace and from 8B to 3.4T tokens/day for the smaller trace between May 2024 and May 2026. By May 2026, the resulting median BI estimates reach 2.41×10^{-3} and 7.95×10^{-5} species·yr/day, respectively. Extrapolating 100T tokens/day over one year yields approximately 0.85 species·yr annually, demonstrating how seemingly negligible per-token impacts can accumulate into non-trivial ecosystem damage at industry scale.

4.3 BI Source Analysis

To better understand the ecological drivers of LLM serving, we next analyze where BIs originate across both lifecycle stages and environmental pathways.

Operational impact dominates LLM serving BI. Figure 4 (left) decomposes BI_{fu} across lifecycle stages under the 100-year horizon. Across profiled serving configurations, the operational stage consistently dominates, contributing more than 95% of total BI on average. In contrast, manufacturing and transportation contribute at the percent level, while end-of-life impact remains negligible. This result suggests that BI in LLM serving is primarily a use-phase problem driven by electricity consumption rather than hardware disposal. As a result, reducing serving energy consumption and improving serving efficiency are likely more effective for lowering BI than optimizing end-of-life management alone.

Different endpoint horizons emphasize different ecological pathways. Figure 4 (right) further decomposes BI into midpoint ecological pathways under different endpoint time horizons. Under average U.S. grid conditions, global warming (GW) is the major contributor, but the relative importance of non-climate pathways changes over time. At the 20-year horizon, GW contributes 71.1% of total BI, while terrestrial acidification (TA) and photo-

Table 1: Grid midpoint intensities and operational biodiversity impact (OBI) for consuming 1 kWh of LLM serving energy across regions at the 100-year horizon. The optimal region for each column is highlighted in **bold**.

Region	$I_{\text{grid,GW}}$ kg CO_2eq	$I_{\text{grid,WC}}$ m^3	$I_{\text{grid,TA}}$ kg SO_2eq	$I_{\text{grid,POF}}$ kg NO_xeq	OBI_{1kWh} species·yr
Norway	$\mathbf{1.06\times10^{-2}}$	2.38×10^{-2}	$\mathbf{2.08\times10^{-5}}$	$\mathbf{1.60\times10^{-5}}$	3.57×10^{-10}
France	4.02×10^{-2}	6.40×10^{-3}	1.54×10^{-4}	1.53×10^{-4}	$\mathbf{2.52\times10^{-10}}$
United Kingdom	1.59×10^{-1}	$\mathbf{1.71\times10^{-3}}$	1.72×10^{-4}	2.14×10^{-4}	5.34×10^{-10}
California	1.98×10^{-1}	5.58×10^{-3}	6.52×10^{-5}	1.57×10^{-4}	6.63×10^{-10}

chemical ozone formation (POF) contribute shares of 18.3% and 10.6%, respectively. At the 100-year horizon, GW increases to 89.1%, with water consumption (WC), TA, and POF each contributing only a few percent. At the 1000-year horizon, GW almost fully dominates the endpoint BI, reaching 98.7%. These results show that BI interpretations depend on the endpoint horizon: longer horizons emphasize climate-related damage, while shorter horizons reveal additional stressors beyond carbon.

4.4 BI vs. Carbon and Water Footprints

Previous results show that operational biodiversity impact dominates the LLM serving case and that global warming is the largest contributor under average U.S. grid conditions. This raises an important question: if climate-related pathways already dominate BI, is carbon alone sufficient for environmentally aware LLM serving decisions? To answer this question, we compare BI against carbon and water footprints across multiple deployment regions.

Carbon-, water-, and BI-optimal regions differ. Table 1 reports grid midpoint intensities and operational BI for consuming 1 kWh of serving energy across four representative regions: Norway, France, United Kingdom, and California. These regions span diverse electricity mixes and environmental intensity profiles across Europe and North America. The carbon-optimal region is Norway. The water-optimal region is the United Kingdom. However, the BI-optimal region is France. These results show that minimizing carbon or water alone does not necessarily minimize BI. Biodiversity-aware LLM serving decisions can differ from carbon- or water-aware deployment strategies.

BI captures tradeoffs across multiple ecological pathways. The divergence arises because BI jointly aggregates multiple ecological pathways rather than optimizing a single environmental metric in isolation. Norway achieves the lowest carbon, terrestrial acidification (TA), and photochemical ozone formation (POF) intensities in this region set, but also exhibits the highest water consumption intensity. In contrast, the United Kingdom minimizes water consumption but has substantially higher carbon and air-pollution-related intensities. France is not individually optimal for either carbon or water, yet achieves the lowest integrated BI after midpoint-to-endpoint biodiversity characterization. For LLM serving, carbon- or water-only decisions can miss broader ecosystem impacts. BI-aware analysis adds a complementary lens for environmentally responsible deployment.

4.5 Quality-Aware Biodiversity Analysis

We next study how BI changes across LLM serving configurations, including model size, model architecture, task type, reasoning mode, traffic load, and GPU generation. Lower BI alone does not necessarily indicate a preferable serving configuration if response quality is too low. Therefore, we jointly analyze BI and QNBI to understand quality-aware ecological efficiency in LLM serving.

Model size. We first analyze how BI and response quality $Q(\theta)$ change with model size on ShareGPT, since chatbot interaction is a dominant real-world LLM usage scenario (Chatterji et al., 2025). Figure 5 shows a tradeoff between response quality and BI as model size increases. Larger models generally achieve higher response quality, but also induce higher BI due to increased computation, memory traffic, and GPU usage. While quality improves rapidly from small to mid-sized models, the gains gradually saturate for recent large models under the LLM-judge evaluation. In contrast, BI continues to increase with model scale, suggesting that the marginal quality gain of very large models can become relatively small compared to their additional BI for daily chat workloads.

To better analyze ecological efficiency, Figure 6 evaluates QNBI across model families. We observe several U-shaped trends within model families, particularly for dense models. Very small models achieve low BI but insufficient response quality, while very large models provide only modest quality gains at higher BI. As a result, the QNBI sweet spot for many dense model families emerges around the 3B–8B scale range. These results sug-

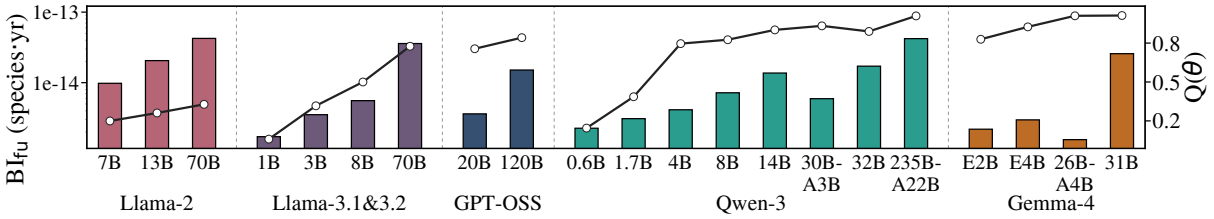

Figure 5: BI_{fu} and quality $Q(\theta)$ for everyday chat workload serving. Bars show BI_{fu} for the most energy-efficient serving configuration of each model, and the black line shows quality score. Models are grouped by family and ordered by size. For Qwen3 models with Instruct / Thinking variants, we report the Instruct models' results here.

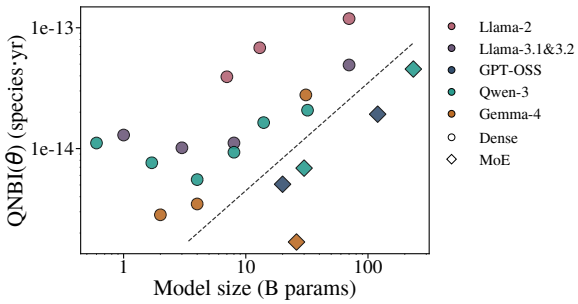

Figure 6: QNBI for daily chat workload serving. Each point corresponds to the most energy-efficient serving configuration of one model. Circles denote dense models and diamonds denote MoE models. The dashed line separates dense models from MoE models.

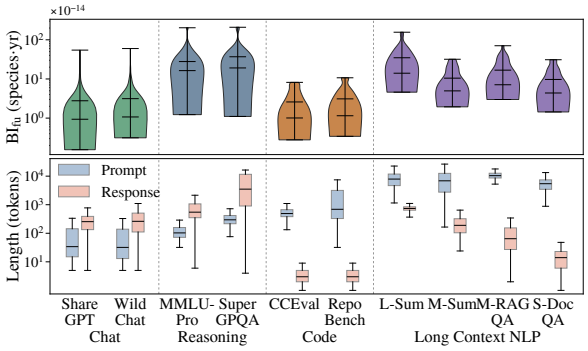

Figure 7: BI_{fu} (top) and request length (bottom) distributions across workload types on the H100 GPU.

gest that environmentally efficient LLM serving is not necessarily achieved by minimizing model size alone. Instead, quality-aware biodiversity analysis can identify intermediate operating points that better balance ecological impact and response quality.

Model Architecture. The BI trend in Figure 5 is not strictly monotonic; some MoE models achieve lower BI than smaller dense models because only a subset of parameters is activated, improving the MST greatly. This advantage is more visible in the QNBI results in Figure 6, where MoE models generally achieve lower QNBI than dense models at comparable total model scale. In practice, this suggests that sparse architectures can preserve much of the response-quality benefit of large-scale models while reducing BI during serving. However, the benefit is not unlimited. Extremely large MoE models, such as Qwen3-235B-A22B, still exhibit higher QNBI than some ∼30B dense models because serving the full model requires 8 H100 GPUs. In this case, the additional serving energy and hardware overhead are not fully offset by throughput or quality gains. These results suggest that sparse activation can improve ecological efficiency for LLM serving, but very large MoE deployments may still incur substantial ecological impact.

Task Type. Figure 7 shows that BI varies substantially across LLM workload types, primarily following the amount of computation required per FU. Chat and code-completion workloads generally induce lower BI because their requests are short or output-limited. Code-completion tasks can include large repository contexts, but their generated continuations are typically short, keeping serving time and energy consumption relatively modest.

In contrast, reasoning and long-context workloads induce higher BI because they increase either prompt length, response length, or both. Among them, long response generation is especially costly. SuperGPQA exhibits one of the highest BI distributions because responses can reach up to 10K tokens, substantially extending decoding time and energy consumption. Longer responses also reduce serving throughput since requests retain KV cache for longer and limit batching efficiency. Long-context NLP workloads exhibit a related but distinct behavior. Large prompts increase prefill computation and KV-cache pressure, while long responses prolong decoding occupancy. As a result, BI depends not only on total token count, but also on how prompt and response lengths interact with serving dynamics.

Reasoning Mode. We next study the tradeoffs

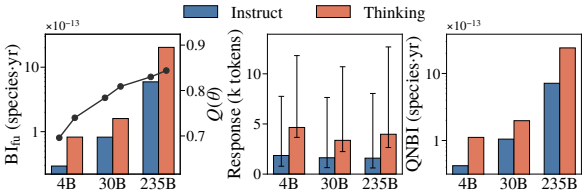

Figure 8: Effect of reasoning mode on MMLU-Pro serving for Qwen3 Instruct and Thinking variants. Left: $\mathrm{BI}_{\mathrm{fu}}$ and quality score $Q(\theta)$. Middle: response length distribution. Right: QNBI.

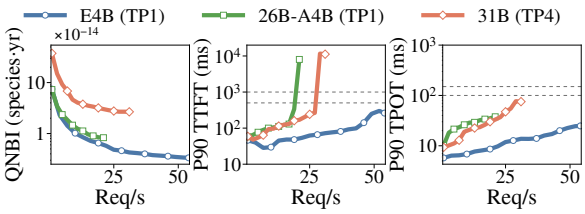

Figure 9: Traffic-load effect on QNBI and latency for ShareGPT serving. Each curve corresponds to one Gemma-4 model and serving configuration. Left: QNBI as incoming request rate increases. Middle and right: p90 TTFT and p90 TPOT; dashed horizontal lines indicate the corresponding SLO thresholds.

of reasoning-oriented serving using Qwen3 Instruct and Thinking variants on MMLU-Pro. As shown in Figure 8, Thinking models consistently improve response quality, but also induce substantially higher BI. The main reason is that thinking mode generates much longer responses, roughly doubling or more the response length, which increases decoding time and GPU energy consumption per request. QNBI results further show that the quality gains often do not fully offset the additional ecological impact, especially for larger models such as the 235B variant. These results suggest that reasoning mode should be enabled selectively for workloads where the expected quality improvement justifies the additional serving overhead.

Traffic Load. Figure 9 shows that QNBI decreases rapidly as traffic load increases, then gradually flattens near serving saturation. At low load, fixed idle power is amortized over few completed requests, leading to high BI per FU. As load increases, batching and GPU utilization improve, so the same serving instance completes more requests per unit energy and thus QNBI drops. The marginal gain becomes smaller near saturation, where TTFT and TPOT begin to rise and further load increases violates SLO. These results support using MST in our experiments since the most biodiversity-

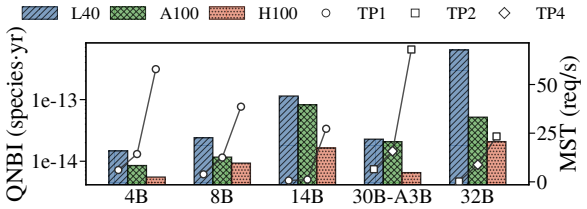

Figure 10: Effect of GPU generation on QNBI and maximum sustainable throughput (MST) for ShareGPT serving. Bar reports QNBI for the most energy-efficient configuration of each Qwen3 model on each GPU. Markers report the corresponding MST, with marker shapes denoting the selected tensor-parallelism (TP) degree.

efficient operating point is typically near the highest stable SLO-satisfying throughput.

GPU Choice. We next study how GPU generation changes quality-aware BI for LLM serving. Figure 10 compares Qwen3 models from 4B to 32B on ShareGPT across L40, A100, and H100. H100 consistently achieves the lowest QNBI, A100 is usually intermediate, and L40 has the highest QNBI, especially for larger models. This gap grows with model size because larger LLMs require more GPU memory and KV cache capacity. When a model cannot run efficiently on a single device, older GPUs often require more tensor parallelism (TP), which increases energy per request.

The MST markers explain this trend. H100 sustains higher request rates through stronger compute, larger memory capacity, higher bandwidth, and faster interconnect, allowing serving overheads to be amortized across more requests. In contrast, L40 often requires multi-GPU tensor parallelism for larger models, but PCIe communication limits the efficiency gain. As a result, GPU choice affects BI not only through power draw, but also through the serving regimes it enables: older GPUs remain effective for smaller single-GPU models, while newer GPUs are more suitable for large or KV-cache-heavy workloads.

5 Conclusion

We present BIRDS, a framework for biodiversity impact analysis of request-driven LLM serving. BIRDS enables quality-aware biodiversity accounting, revealing ecological tradeoffs that are not captured by carbon or water metrics alone. Our results show that biodiversity-aware analysis can provide actionable insight for more environmentally responsible LLM serving and deployment.

Limitations

We discuss the limitations of this work as follows.

Non-agentic Scope. BIRDS focuses on non-agentic LLM serving workloads where requests are self-contained and do not invoke external tools or retrieval systems. Agentic LLM systems may introduce additional ecological costs through tool usage, multi-step reasoning, external API calls, or repeated interaction loops that are not captured in our current modeling framework.

Uncertainty in Biodiversity Accounting. Our biodiversity accounting relies on life cycle assessment (LCA) characterization factors and regional environmental intensity data. Although these datasets are widely used in environmental assessment, biodiversity modeling remains inherently uncertain due to regional ecological variability, incomplete environmental reporting, and differences across LCA methodologies and endpoint horizons. Therefore, the reported BI values should be interpreted as comparative estimates rather than exact measurements of ecological damage.

Grid Assumptions. Our operational impact analysis uses average regional grid intensity factors and does not model fine-grained temporal variation in electricity generation. Incorporating spatiotemporal electricity characteristics may further improve biodiversity-aware serving analysis.

Quality Evaluation Dependency. Our quality-aware analysis depends on benchmark-specific evaluation protocols and LLM-judge scoring for open-ended generation tasks. Different evaluation metrics or judge models may affect the QNBI values and ecological tradeoff interpretations.

Serving-Focused Lifecycle Coverage. This work focuses on biodiversity impact during LLM serving and does not model the full lifecycle ecological impact of model training, data collection, or downstream user behavior. Extending biodiversity-aware analysis across the broader lifecycle of foundation models remains important future work.

Ethical considerations

This work studies the biodiversity impact of LLM serving systems and aims to support more environmentally responsible AI deployment. Our framework does not involve human subjects, personal data collection, or deployment of LLMs in real-world decision-making settings. All experiments use publicly available models, benchmarks, and serving systems.

The biodiversity impact estimates reported in this work are based on LCA methodologies and should be interpreted as comparative ecological indicators rather than exact measurements of ecosystem damage. Different LCA datasets, regional environmental assumptions, and endpoint horizons may produce different quantitative estimates.

We emphasize that biodiversity impact should complement rather than replace existing environmental metrics such as carbon emissions and water consumption. The goal of this work is not to prescribe a single universal objective, but to encourage broader ecological consideration in future LLM development and deployment decisions.

Quality-aware biodiversity analysis introduces the possibility of optimizing environmental efficiency at the cost of model capability or user experience. We therefore caution against using ecological metrics alone to justify degrading model quality or restricting access to useful AI systems, especially in educational, accessibility, or safety-critical applications.

The authors used AI assistants, including ChatGPT and Codex, during the preparation of this work. They were used for language and LaTeX polishing, public-data crawling and extraction, code generation, debugging, experiment orchestration scripts, and plotting. All AI-assisted text, code, and analysis were reviewed, edited, and validated by the authors. The scientific claims, numerical results, and final interpretations in this paper are based on the authors' modeling choices, profiling experiments, and post-processing scripts; citations were selected and verified by the authors. We did not use AI assistants to process private personal data or reviewer-confidential materials.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *Preprint*, arXiv:2308.14508.

Arlene Blum. 2024. Chip manufacturing shortcuts harm our health and environment. *Forbes*.

Andreas Busa, Malcolm Hegeman, Jeff Vickers, Natalia Duque-Ciceri, and Constantin Herrmann. 2019. Life cycle assessment of dell r740. Technical report, Dell Technologies. Accessed 19 May 2025.

Bradley J Cardinale, J Emmett Duffy, Andrew Gonzalez, David U Hooper, Charles Perrings, Patrick

Venail, Anita Narwani, Georgina M Mace, David Tilman, David A Wardle, and 1 others. 2012. Biodiversity loss and its impact on humanity. *Nature*, 486(7401):59–67.

Aaron Chatterji, Thomas Cunningham, David J Deming, Zoe Hitzig, Christopher Ong, Carl Yan Shan, and Kevin Wadman. 2025. How people use chatgpt. Technical report, National Bureau of Economic Research.

Yangruibo Ding, Zijian Wang, Wasi Uddin Ahmad, Hantian Ding, Ming Tan, Nihal Jain, Murali Krishna Ramanathan, Ramesh Nallapati, Parminder Bhatia, Dan Roth, and Bing Xiang. 2023. Crosscodeeval: A diverse and multilingual benchmark for cross-file code completion. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yi Ding and Tianyao Shi. 2024. Sustainable llm serving: Environmental implications, challenges, and opportunities. In *2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC)*, pages 37–38. IEEE.

European Commission Joint Research Centre (JRC) and Netherlands Environmental Assessment Agency. 2024. EDGAR_2024_GHG: Emissions database for global atmospheric research. `https://edgar.jrc.ec.europa.eu/dataset_ghg2024`. Accessed: 2025-05-19.

Ahmad Faiz, Sotaro Kaneda, Ruhan Wang, Rita Osi, Prateek Sharma, Fan Chen, and Lei Jiang. 2024. Llmcarbon: Modeling the end-to-end carbon footprint of large language models. In *International Conference on Learning Representations*, volume 2024, pages 24727–24741.

Sophia Falk, David Ekchajzer, Thibault Pirson, Etienne Lees-Perasso, Augustin Wattiez, Lisa Biber-Freudenberger, Sasha Luccioni, and Aimee van Wynsberghe. 2025. More than carbon: Cradle-to-grave environmental impacts of genai training on the nvidia a100 gpu. *arXiv preprint arXiv:2509.00093*.

Matthias Finkbeiner, Atsushi Inaba, Reginald Tan, Kim Christiansen, and Hans-Jürgen Klüppel. 2006. The new international standards for life cycle assessment: Iso 14040 and iso 14044. *The international journal of life cycle assessment*, 11:80–85.

Google DeepMind. 2026. Gemma 4 Model Card. `https://ai.google.dev/gemma/docs/core/model_card_4`. Accessed: 2026-05-14.

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

GreenDelta. 2026. openlca nexus: ELCD database. GreenDelta GmbH. Accessed: 2026-05-22.

Pranjol Sen Gupta, Md Rajib Hossen, Pengfei Li, Shaolei Ren, and Mohammad A Islam. 2024. A dataset for research on water sustainability. In *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, pages 442–446.

Udit Gupta, Mariam Elgamal, Gage Hills, Gu-Yeon Wei, Hsien-Hsin S. Lee, David Brooks, and Carole-Jean Wu. 2022. ACT: Designing sustainable computer systems with an architectural carbon modeling tool. In *ISCA*.

Mark A.J. Huijbregts, Zoran J.N. Steinmann, Pieter M.F. Elshout, Geert Stam, Francesca Verones, Marisa Vieira, Anne Hollander, Michiel Zijp, and Rosalie van Zelm. 2016. ReCiPe 2016: A Harmonized Life Cycle Impact Assessment Method at Midpoint and Endpoint Level. Report I: Characterization. Technical report, RIVM National Institute for Public Health and the Environment, Bilthoven, The Netherlands.

IPBES. 2019. Summary for policymakers of the global assessment report on biodiversity and ecosystem services of the intergovernmental science-policy platform on biodiversity and ecosystem services. Technical report, IPBES secretariat, Bonn, Germany. S. Díaz, J. Settele, E. S. Brondízio, H. T. Ngo, M. Guèze, J. Agard, A. Arneth, P. Balvanera, K. A. Brauman, S. H. M. Butchart, K. M. A. Chan, L. A. Garibaldi, K. Ichii, J. Liu, S. M. Subramanian, G. F. Midgley, P. Miloslavich, Z. Molnár, D. Obura, A. Pfaff, S. Polasky, A. Purvis, J. Razzaque, B. Reyers, R. Roy Chowdhury, Y. J. Shin, I. J. Visseren-Hamakers, K. J. Willis, and C. N. Zayas (eds.).

Scotten W Jones. 2023. Modeling 300mm wafer fab carbon emissions. In *2023 International Electron Devices Meeting (IEDM)*, pages 1–4. IEEE.

Walter Klöpffer and Birgit Grahl. 2014. *Life cycle assessment (LCA): a guide to best practice*. John Wiley & Sons.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Baolin Li, Rohan Basu Roy, Daniel Wang, Siddharth Samsi, Vijay Gadepally, and Devesh Tiwari. 2023. Toward sustainable hpc: Carbon footprint estimation and environmental implications of hpc systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. ACM.

Baolin Li, Yankai Jiang, Vijay Gadepally, and Devesh Tiwari. 2024. Sprout: Green generative ai with carbon-efficient llm inference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21799–21813.

Pengfei Li, Jianyi Yang, Mohammad A Islam, and Shaolei Ren. 2025. Making ai less' thirsty'. *Communications of the ACM*, 68(7):54–61.

Tianyang Liu, Canwen Xu, and Julian McAuley. 2024. Repobench: Benchmarking repository-level code auto-completion systems.

Gillian F Menzies, Seyhan Turan, and Philip FG Banfill. 2007. Life-cycle assessment and embodied energy: a review. *Proceedings of the Institution of Civil Engineers-Construction Materials*, 160(4):135–143.

Microsoft. 2026. Microsoft datacenters: Efficiency. Microsoft Corporation. Accessed: 2026-05-22.

NVIDIA. 2020. NVIDIA A100 Tensor Core GPU Datasheet. https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf. Accessed: 2026-05-10.

NVIDIA. 2022a. NVIDIA H100 Tensor Core GPU. https://www.nvidia.com/en-us/data-center/h100/. Accessed: 2026-05-10.

NVIDIA. 2022b. NVIDIA L40 GPU for Data Center. https://www.nvidia.com/en-us/data-center/l40/. Accessed: 2026-05-10.

NVIDIA Corporation. 2025. Nvidia management library (nvml).

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

OpenRouter. 2026. OpenRouter rankings. OpenRouter. Accessed: 2026-05-22.

Sundar Pichai. 2026. Google I/O 2026 keynote. Google The Keyword Blog. Accessed: 2026-05-22.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shaolei Ren, Bill Tomlinson, Rebecca W Black, and Andrew W Torrance. 2024. Reconciling the contrasting narratives on the environmental impact of large language models. *Scientific Reports*, 14(1):26310.

ShareGPT. Sharegpt - share and save your conversations with ai. https://sharegpt.com/.

Tianyao Shi, Ritbik Kumar, Inez Hua, and Yi Ding. 2025. When servers meet species: A fab-to-grave lens on computing's biodiversity impact. *ACM SIGENERGY Energy Informatics Review*, 5(2):34–40.

SK hynix Inc. 2025. Sustainability reports & policies. https://www.skhynix.com/sustainability/UI-FR-SA1601/. Web page hosting SK hynix annual sustainability reports; accessed 19 May 2025.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 3645–3650.

Taiwan Semiconductor Manufacturing Company Limited. 2025. Esg resources and documents. https://esg.tsmc.com/en/resources/documents.html. Web page, accessed 19 May 2025.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixing Deng, Shuyue Guo, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, and 76 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *Preprint*, arXiv:2502.14739.

TechPowerUp. 2026. NVIDIA H100 SXM5 80 GB specs. TechPowerUp GPU Database. Accessed: 2026-05-17.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

UNEP. 2024. Environmental impacts of e-waste: Heavy metal leaching and ecosystem disruption. *UNEP Reports on E-Waste*. E-waste releases heavy metals that contaminate soil and degrade freshwater and marine habitats.

United States Environmental Protection Agency (EPA). 2025. Emissions & generation resource integrated database (egrid), egrid2023rev1. https://www.epa.gov/egrid. Accessed: 2025-05-19.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Yanran Wu, Inez Hua, and Yi Ding. 2025a. Not all water consumption is equal: A water stress weighted metric for sustainable computing. *ACM SIGENERGY Energy Informatics Review*, 5(2):84–90.

Yanran Wu, Inez Hua, and Yi Ding. 2025b. Unveiling environmental impacts of large language model serving: A functional unit view. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10560–10576.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Table 2: Biodiversity-related midpoint impact categories from the ReCiPe2016 (Huijbregts et al., 2016) LCA framework studied in this paper.

Midpoint Impact Category	Unit	Definition
Global Warming (GW)	kg CO_2 eq.	Species loss from climate-change-driven ecosystem damage caused by greenhouse gas emissions, expressed relative to CO_2 equivalents.
Water Consumption (WC)	m^3	Species loss from reduced freshwater availability caused by consumptive water use.
Terrestrial Acidification (TA)	kg SO_2 eq.	Species loss caused by increased soil acidity from emissions such as SO_2, NO_x, and NH_3.
Terrestrial Toxicity (TT)	kg 1,4-DCB eq.	Species loss from toxic chemical releases to industrial soil, expressed relative to 1,4-dichlorobenzene equivalents.
Photochemical Ozone Formation (POF)	kg NO_x eq.	Species loss driven by ground-level ozone formation from reactions involving NO_x and volatile organic compounds.
Freshwater Eutrophication (FE)	kg P eq.	Species loss from phosphorus-driven nutrient enrichment that depletes oxygen in freshwater bodies.
Freshwater Toxicity (FT)	kg 1,4-DCB eq.	Species loss due to toxic chemical emissions to freshwater, expressed relative to 1,4-dichlorobenzene equivalents.
Marine Eutrophication (ME)	kg N eq.	Species loss from nitrogen-driven nutrient enrichment in marine environments.
Marine Toxicity (MT)	kg 1,4-DCB eq.	Species loss from toxic chemical releases to seawater, expressed relative to 1,4-dichlorobenzene equivalents.

A Appendix Overview

We summarize the appendix as follows:

- §B details the LCA modeling for biodiversity impact. §B.1 summarizes the ReCiPe2016 midpoint categories used to translate computing-related activities into biodiversity impact. §B.2 describes how we model embodied biodiversity impact (EBI) for datacenter GPUs across manufacturing, transportation, and end-of-life stages. §B.3 reports the public data sources used to instantiate operational grid intensities, GPU manufacturing impacts, transportation impacts, and end-of-life impacts.
- §C lists the evaluation specifications. It summarizes the LLM serving workloads, evaluated model families, and GPU platforms used in our experiments. It also defines the workload-specific latency SLOs used to determine valid serving operating points.
- §D describes the profiling details for LLM serving workloads. §E explains how we measure response quality for open-ended chat workloads and objective benchmarks. §D.2 describes how we collect GPU power traces, compute serving energy, identify stable SLO-satisfying operating points, and determine maximum sustainable throughput.

Table 3: Midpoint-to-endpoint conversion factors $\Phi_c(T)$ from ReCiPe2016.

Midpoint	$T = 20$	$T = 100$	$T = 1000$
GW	5.32×10^{-10}	2.80×10^{-9}	2.50×10^{-8}
WC	6.04×10^{-13}	1.35×10^{-8}	1.35×10^{-8}
TA	2.12×10^{-7}	2.12×10^{-7}	2.12×10^{-7}
TT	1.14×10^{-11}	1.14×10^{-11}	1.14×10^{-11}
POF	1.29×10^{-7}	1.29×10^{-7}	1.29×10^{-7}
FE	6.71×10^{-7}	6.71×10^{-7}	6.71×10^{-7}
FT	6.95×10^{-10}	6.95×10^{-10}	6.95×10^{-10}
ME	1.70×10^{-9}	1.70×10^{-9}	1.70×10^{-9}
MT	1.05×10^{-10}	1.05×10^{-10}	1.05×10^{-10}

For midpoint categories with multiple ecosystem endpoint rows in ReCiPe2016, e.g., GW and WC, $\Phi_c(T)$ sums the corresponding ecosystem-damage factors.

B LCA Modeling for Biodiversity Impact

B.1 Midpoint Categories

We detail the environmental impact categories used to translate computing-related physical activities into biodiversity impact. Table 2 reports the ReCiPe2016 (Huijbregts et al., 2016) midpoint impact categories used in this work, following its ecosystem-quality endpoint characterization factors. It lists each category, the reference unit, and the corresponding ecological interpretation. Table 3 reports the midpoint to endpoint conversion factor $\Phi_c(T)$ values with respect to different time horizon $T \in \{20, 100, 1000\}$.

B.2 EBI Modeling for Datacenter GPUs

Following FABRIC (Shi et al., 2025), we model the embodied biodiversity impact of a GPU by decomposing its non-operational life cycle into Manufacturing (Mfg), transportation (Trans), and end-of-life (EoL):

$$\mathrm{EBI}(d) = \sum_{l \in \{\mathrm{Mfg,Trans,EoL}\}} \sum_{c \in \mathcal{C}} M_{c,l}(d) \Phi_c(T), \tag{7}$$

where d denotes a GPU, c indexes midpoint categories, $M_{c,l}(d)$ is the midpoint impact of lifecycle stage l, and $\Phi_c(r)$ converts midpoint impact into endpoint ecosystem damage.

Manufacturing. We model GPU manufacturing as IC production, covering both the accelerator logic dies and attached memory dies. Following ACT (Gupta et al., 2022) and FABRIC (Shi et al., 2025), the manufacturing midpoint impact of IC component x is

$$M^{\mathrm{IC}}_{c,\mathrm{Mfg}}(x,r,t) = \frac{A_x}{Y_x} \left(\mathrm{EPA}_x \cdot I_{\mathrm{grid},c}(r,t) + I_{\mathrm{proc},c}(r,t) \right), \tag{8}$$

where x denotes either a logic or memory IC component, A_x is the active IC area attributed to component x, and Y_x is manufacturing yield. EPA_x is electricity use per unit active silicon area, $I_{\mathrm{grid}c}(r,t)$ is the grid electricity midpoint intensity for category c in region r and year t, and $I_{\mathrm{proc},c}(r,t)$ is the process-related midpoint intensity per unit active silicon area (i.e., direct fab pollutant emission in exhaust and wastewater). We focus on IC components because they dominate semiconductor manufacturing impacts (Falk et al., 2025), and omit peripheral components like print circuit board (PCB), surface mount device (SMD), and mechanical parts following prior LCA modeling practice (Gupta et al., 2022).

Transportation. Transportation midpoint impact is modeled from shipped mass and transportation mode (e.g., sea, air, truck, etc.):

$$M_{c,\mathrm{Trans}}(d) = m_d \sum_{u \in \mathcal{U}_{\mathrm{trans}}} D_u I^u_{c,\mathrm{Trans}}, \tag{9}$$

where m_d is the shipped mass of GPU d, u indexes transport modes, D_u is the distance traveled by mode u, and $I^u_{c,\mathrm{Trans}}$ is the corresponding midpoint intensity per mass-distance product.

End-of-Life. End-of-life midpoint impact is modeled using pathway-weighted treatment intensities:

$$M_{c,\mathrm{EoL}}(d) = m_d \sum_{p \in \mathcal{P}_{\mathrm{EoL}}} \rho_p I^p_{c,\mathrm{EoL}}, \quad \sum_{p \in \mathcal{P}_{\mathrm{EoL}}} \rho_p = 1, \tag{10}$$

where p indexes treatment pathways including recycling, incineration, and landfill, ρ_p is the pathway ratio, and $I^p_{c,\mathrm{EoL}}$ is the midpoint intensity of pathway p.

B.3 Data Sources

We use public and reproducible data sources to parameterize the LCA model. For operational electricity and electricity-related manufacturing impact, the grid midpoint intensity $I_{\mathrm{grid},c}(r,t)$ is derived from authoritative air pollutant emission inventories, including EPA eGRID (United States Environmental Protection Agency (EPA), 2025) and EDGAR (European Commission Joint Research Centre (JRC) and Netherlands Environmental Assessment Agency, 2024). The datacenter power usage effectiveness (PUE) and water usage effectiveness (WUE) data are from Microsoft (2026) public release. The operational water consumption data are from existing water studies (Gupta et al., 2024). The midpoint-to-endpoint conversion factors $\Phi_c(r)$ follow the ReCiPe2016 framework (Huijbregts et al., 2016).

For GPU manufacturing, logic-die area A_x is obtained from die-shot analysis or public die-area disclosures (TechPowerUp, 2026), while memory-die area is inferred from memory capacity and bit density (Jones, 2023). We set the manufacturing yield Y_x to 0.875, following prior carbon modeling practice (Li et al., 2023). The process-related manufacturing midpoint intensity $I^{\mathrm{proc}}_{x,c}(r,t)$, is derived from leading semiconductor vendors' sustainability disclosures, including Taiwan Semiconductor Manufacturing Company Limited (2025) and SK hynix Inc. (2025).

For transportation and end-of-life modeling, we use transportation mode, distance assumptions D_u, and recycling pathway ratios ρ_p from industrial server LCA reports (Busa et al., 2019). The transportation midpoint intensities $I^u_{c,\mathrm{Trans}}$ and end-of-life pathway intensities $I^p_{c,\mathrm{EoL}}$ are taken from the ELCD 3.2 LCA database through OpenLCA (GreenDelta, 2026). These sources provide the mass-distance and treatment-pathway intensities needed to instantiate Equations (9)

and (10).

C List of Workloads, Models, and GPUs

Table 4 lists the LLM serving workloads and their corresponding request datasets with prompt and response length statistics used in the measurement studies in this work. Table 5 summarizes the evaluated model families and their parameter-size buckets. Table 6 list the specifications of GPU platforms used in the experiments.

Latency SLOs specify the service condition under which a request-level functional unit is defined. A serving operating point is considered valid only when the achieved throughput satisfies the workload-specific p90 TTFT and p90 TPOT constraints. Table 7 summarizes the consolidated SLOs by workload family and model-size bucket. Chat workloads use interactive latency targets; code workloads use tighter TPOT constraints for completion responsiveness; reasoning workloads allow longer generation latency; and LongBench uses a prompt-length-scaled TTFT threshold because input lengths vary substantially across tasks.

For LongBench, a static TTFT threshold would either over-constrain long-prompt tasks or over-relax shorter document QA tasks. We therefore use a length-scaled TTFT threshold. For a request i in LongBench subtask b with prompt length L_i input tokens, the TTFT threshold is

$$T_{i,b}^{\mathrm{TTFT}} = 1000 \cdot \min\left(C_b,\ B_b + \alpha_b \frac{L_i}{1000}\right) \quad \mathrm{ms}, \tag{11}$$

where B_b is the base TTFT budget in seconds, α_b is the additional budget per 1K input tokens, and C_b is the cap in seconds. All LongBench subtasks use a fixed p90 TPOT threshold of 150 ms.

D Profiling Harness

D.1 Output Quality

We evaluate output quality using two tracks: a chat-style LLM-as-judge track and a ground-truth benchmark track. For open-ended chat workloads, we compute a pairwise quality score against a chosen reference response:

$$q_{\mathrm{chat}} = \frac{N_{\mathrm{win}} + 0.5 N_{\mathrm{tie}}}{N_{\mathrm{win}} + N_{\mathrm{tie}} + N_{\mathrm{loss}}}, \tag{12}$$

where a win means the judge prefers the candidate response, a tie receives half credit, and invalid judgments are excluded from the denominator. Listing 1 shows the LLM-judge prompt template we used. The reference model is selected as Llama-3.1-8B and the evaluator model is GPT-5.4. We randomize A/B ordering when constructing judge batches and keep position-wise aggregation statistics to check for judge-position bias. This score is used as $Q(\theta)$ for chat-style workloads in QNBI.

Listing 1: Pairwise LLM-judge prompt for everyday text conversation quality evaluation.

```
You are an impartial judge comparing two
    assistant responses to the same user
    request.

User request:
{prompt}

Assistant A:
{response_a}

Assistant B:
{response_b}

Judge which response better satisfies the
    user request.

For objective or technical prompts,
    prioritize factual correctness, reasoning
    correctness, and functional correctness.
For subjective or open-ended prompts,
    consider helpfulness, relevance, factual
    soundness, clarity, and conciseness.
Do not prefer a response merely because it is
    longer or more structured in terms of
    formatting.
If both responses are similarly good or
    similarly flawed, choose Tie.

Return JSON only:
{
  "winner": "A" | "B" | "Tie",
  "reason": "one concise sentence"
}
```

For objective benchmarks, we use each benchmark's native or adapter-selected quality metric. MMLU-Pro and SuperGPQA are scored by final-answer accuracy against the ground-truth label. Their results are collected from each model's official technical report or Huggingface model card when they are available. For Llama and Gemma models missing SuperGPQA score, we run the benchmarks following the original code. LongBench tasks are scored with the original task-to-metric mapping and reported primarily by workload bucket: ROUGE-style scores for summarization and Chinese QA (DuReader) tasks, and F1-style scores for document QA tasks. For CrossCodeEval, we use edit similarity (ES) as the primary score, with exact match (EM) and identifier F1 as supporting metrics. For RepoBench, we re-

Table 4: Selected LLM serving workloads and their prompt/response length statistics.

Task	Description	Request Dataset	Prompt Length			Response Length		
			P50	P90	P95	P50	P90	P95
Text Conversation	Everyday chatbot conversations sampled from real-world user–LLM interactions.	ShareGPT (ShareGPT)	31	701	1,377	243	568	703
		WildChat (Zhao et al., 2024)	27	562	1,004	233	777	958
Problem Solving	Knowledge-intensive question answering and reasoning workloads.	MMLU-Pro (Wang et al., 2024)	103	245	317		–	
		SuperGPQA-Hard (Team et al., 2025)	296	626	815		–	
Code Completion	IDE-style code completion requests with repository context or retrieval content embedded in the prompt.	CrossCodeEval (Ding et al., 2023)	490	963	1,221	3	8	10
		RepoBench (Liu et al., 2024)	691	5,366	6,732	3	7	9
Long Context NLP	Long-output, low-interactivity document summarization.	LongBench-GovReport (Bai et al., 2023)	8,432	17,316	21,185	655	876	934
	Medium-output multi-document/news and meeting summarization.	LongBench-MultiNews, QMSum, VCSum	2,008	11,752	16,221	160	358	395
	Medium-output document-based question answering.	LongBench-DuReader	10,396	14,258	15,196	65	222	263
	Short-output document-based question answering.	LongBench-MultiFieldQA, Qasper	5,059	8,780	10,581	14	49	70

port ES as the primary score and EM as the companion metric.The selected primary score for each workload is used as $Q(\theta)$ (ROUGE-style scores are divided by 100).

Table 9 list the exact quality score values of studied models from our quality profiling experiments. LongBench summarization scores should be interpreted with caution: its ROUGE-style metrics often reward compact responses that closely match the reference wording, so larger instruction-tuned models may be penalized for producing more verbose or hedged answers even when the summary is substantively acceptable.

Quality-generation runs use deterministic single-sample decoding with `temperature=0.0`. SuperGPQA uses a 16K-token completion budget, LongBench uses 4K, and both code-completion benchmarks use 128 tokens. For models with explicit thinking modes, we disable thinking for non-thinking model variants where supported, and remove visible reasoning traces before scoring always-thinking variants when required by the benchmark protocol.

For IDE-style code-completion workloads, retrieval context is materialized offline rather than executed live. RepoBench and CrossCodeEval prompts serialize the already-available repository context, imports, and current-file prefix into a single request payload; no tool call, retrieval step, or agent loop occurs during serving. For compatibility with modern instruction-tuned models, some reruns use a chat-formatted completion prompt that marks the cursor position and asks the model to return only the code continuation. This changes the delivery format but not the benchmark semantics: each request remains a non-agentic single-turn completion task with fixed embedded context.

D.2 Energy Profiling

We measure GPU-board energy during serving by sampling NVML power counters at 25 ms intervals and numerically integrating the power trace over the measurement window. For multi-GPU serving instances, we collect traces from all visible GPUs and sum their energy. The energy measurement is then multiplied by the datacenter PUE to account for extra overhead like cooling at the infrasturacture level. Each profiling run includes an idle-baseline period, request-traffic warmup, steady-state measurement, and cooldown. Unless otherwise noted, we use 30 s idle-baseline measurement, 60 s traffic warmup, 180 s profiling duration, 15 s cooldown, and three repeated trials with 30 s between-repeat cooldown.

To reduce boundary artifacts, we truncate the first and last 5 s of GPU-monitor traces before computing energy statistics. Runtime serving metrics,

Table 5: Evaluated model families grouped by parameter-size bucket.

Family	Size Class			
	< 2B	3–8B	13–34B	$\geq$70B
Llama 2 (Touvron et al., 2023)	–	7B-Chat	13B-Chat	70B-Chat
Llama 3.1/3.2 (Grattafiori et al., 2024)	1B-Instruct	3B-Instruct, 8B-Instruct	–	70B-Instruct
GPT-OSS (OpenAI, 2025)	–	–	*20B	*120B
Qwen3 (Qwen Team, 2025)	0.6B, 1.7B	4B-Instruct / Thinking, 8B	14B, *30B-A3B-Instruct / Thinking, 32B	*235B-A22B-Instruct / Thinking
Gemma 4 (Google DeepMind, 2026)	E2B-it	E4B-it	*26B-A4B-it, 31B-it	–

Note. Asterisks mark MoE models; all other listed models are dense models. MoE models are bucketed by total parameter count. Qwen3 models with Instruct / Thinking variants are the latest 2507 version.

Table 6: NVIDIA GPU platforms used in our LLM serving experiments.

GPU	Architecture	Memory	Memory Bandwidth	TDP
L40 (NVIDIA, 2022b)	Ada Lovelace	48 GB GDDR6	864 GB/s	300 W
A100 SXM (NVIDIA, 2020)	Ampere	40 GB HBM2	1,555 GB/s	400 W
H100 SXM (NVIDIA, 2022a)	Hopper	80 GB HBM3	3.35 TB/s	Up to 700 W

Table 7: Consolidated latency SLOs used, reported as p90 TTFT and p90 TPOT.

Workload	Model size	TTFT	TPOT
Text Conversation	< 3B	250 ms	50 ms
	3B–4B	500 ms	100 ms
	$\geq$7B	1000 ms	150 ms
Problem Solving	< 3B	1500 ms	150 ms
	3B–4B	1500 ms	150 ms
	$\geq$8B	2000 ms	200 ms
CrossCodeEval	< 3B	500 ms	40 ms
	3B–4B	750 ms	50 ms
	$\geq$8B	1500 ms	75 ms
RepoBench	< 3B	5000 ms	50 ms
	3B–4B	5000 ms	50 ms
	$\geq$8B	5000 ms	75 ms
Long-Context NLP	all buckets	length-scaled	150 ms

including throughput, TTFT, TPOT, and request completion statistics, are recorded at 1 s intervals. We evaluate stability over sliding 10 s windows and retain only operating points that complete requests without queue divergence and satisfy the workload-specific TTFT and TPOT constraints.

For each workload–model–GPU setting, we sweep incoming request rates to identify stable SLO-satisfying operating points. The maximum sustainable throughput (MST) is the largest achieved throughput that remains stable under the latency constraints. Since fixed serving overheads and idle power are amortized over more completed requests at higher stable throughput, MST typically gives the lowest energy and biodiversity impact per request among satisfactory operating points. Sub-MST runs are retained for the traffic-load analysis.

Table 8: Length-scaled TTFT parameters for LongBench workloads. The active TTFT threshold is computed by Equation (11); TPOT is fixed at 150 ms for all LongBench subtasks.

LongBench subtask	B_b (s)	α_b (s / 1K tok.)	C_b (s)
Long Summarization	8.0	1.4	45.0
Medium Summarization	6.0	1.2	40.0
Medium RAG QA	5.0	0.9	30.0
Short Document QA	4.0	0.8	20.0

E Additional Results

This section provides additional results following the same visualization style of Figure 5, Figure 6, and Figure 10. For code-completion workloads, Figures 11 and 19 report CrossCodeEval results and Figures 12 and 20 report RepoBench results. For reasoning workloads, Figures 13 and 21 report MMLU-Pro results and Figures 14 and 22 report SuperGPQA results. For LongBench workloads, Figures 15 to 18 and 23 to 26 report the four evaluated task buckets.

Table 9: Output-quality scores used for QNBI. All values are percentages. The best scores are in bold for each benchmark. Chat denotes the LLM-judge win/tie quality score; MMLU-Pro and SuperGPQA denote benchmark accuracy; CrossCodeEval reports edit similarity (ES), exact match (EM), and identifier F1 (ID-F1); RepoBench reports edit similarity and exact match; LongBench reports bucket-level scores for long-output summarization (L-Sum), medium-output summarization (M-Sum), medium-answer RAG QA, and short-answer document QA. Missing entries indicate incompatible model–workload pairs: Llama 2 models are excluded from long-context tasks due to their 4K context-length limit; Reasoning models (GPT-OSS and Qwen3-Thinking variants) are excluded from code-completion tasks.

Model	Chat	MMLU-Pro	Super GPQA	CrossCodeEval			RepoBench		LongBench			
				ES	EM	ID-F1	ES	EM	L-Sum	M-Sum	RAG-QA	Doc-QA
Llama-2-7B	24.9	20.3	12.8	51.8	6.8	40.7	30.3	0.1	–	–	–	–
Llama-2-13B	30.1	25.3	14.7	52.9	7.8	42.1	34.4	0.01	–	–	–	–
Llama-2-70B	35.6	37.5	18.0	53.7	9.5	44.0	31.6	1.62	–	–	–	–
Llama-3.2-1B	13.3	22.2	11.6	46.5	4.4	33.9	24.9	0.2	30.4	21.7	26.9	32.6
Llama-3.2-3B	34.7	11.9	18.5	49.5	6.5	38.3	38.3	3.2	36.1	22.8	**37.1**	49.7
Llama-3.1-8B	50.0	44.2	21.0	52.3	9.1	42.4	43.5	8.2	36.6	**23.9**	34.6	53.5
Llama-3.1-70B	73.0	62.8	35.4	54.9	11.6	45.9	56.0	24.8	**37.7**	23.8	32.6	55.1
GPT-OSS-20B	71.4	73.6	44.9	–	–	–	–	–	30.9	18.6	19.8	35.0
GPT-OSS-120B	78.5	80.8	51.9	–	–	–	–	–	28.8	17.8	22.0	35.0
Qwen3-0.6B	20.4	24.7	15.0	46.8	4.8	34.5	36.5	0.2	28.1	21.0	33.0	30.7
Qwen3-1.7B	40.5	36.8	20.9	49.2	6.6	38.2	39.6	4.6	31.3	22.3	33.2	41.0
Qwen3-4B-I	74.7	69.6	42.8	52.4	9.4	42.3	44.6	9.2	30.9	20.5	26.4	51.1
Qwen3-4B-T	62.3	74.0	47.8	–	–	–	–	–	33.5	21.3	24.5	53.5
Qwen3-8B	77.1	56.7	31.6	52.4	10.0	42.7	46.5	12.2	33.5	21.8	26.9	51.8
Qwen3-14B	83.5	61.0	34.3	53.1	10.6	43.8	58.1	22.3	33.6	22.1	31.0	53.4
Qwen3-30B-A3B-I	86.1	78.4	53.4	54.8	12.2	46.0	51.1	11.5	31.6	19.9	24.6	53.0
Qwen3-30B-A3B-T	80.8	80.9	56.8	–	–	–	–	–	31.1	20.6	27.6	53.0
Qwen3-32B	82.5	65.5	39.8	53.4	11.4	44.2	58.0	20.5	33.3	22.0	29.5	53.2
Qwen3-235B-A22B-I	92.5	83.0	62.6	57.1	15.0	48.9	**66.7**	**34.7**	32.5	20.1	25.6	53.6
Qwen3-235B-A22B-T	87.9	84.4	**64.9**	–	–	–	–	–	32.4	18.8	23.5	52.6
Gemma-4-E2B	77.5	60.0	32.4	42.4	3.1	29.5	34.6	1.85	32.7	21.7	33.5	49.4
Gemma-4-E4B	85.4	69.4	40.4	53.9	6.5	41.4	43.6	8.39	32.0	21.6	27.2	54.0
Gemma-4-26B-A4B	92.5	82.6	56.9	54.9	9.2	45.2	39.4	5.6	32.9	20.9	26.2	54.7
Gemma-4-31B	**92.7**	**85.2**	62.3	**65.3**	**16.5**	**54.8**	52.2	18.1	32.1	21.7	25.0	**56.2**

Across these additional workloads, the overall BI–quality trend is consistent with the main-text ShareGPT case: larger models generally improve quality while increasing BI_{fu}, and QNBI often favors intermediate-scale or sparse models rather than the largest configuration. The ∼30B MoE models frequently form a BI_{fu} valley, reflecting their ability to provide higher quality without activating all parameters per token. Gemma4-26B-A4B, however, often incurs higher BI_{fu} than Gemma4-E4B, especially on longer-context tasks where memory pressure and longer serving time increase per-request energy. In the size–QNBI space, MoE and dense models remain mostly separable, with MoE models usually achieving lower QNBI at comparable total model size. RepoBench and reasoning workloads are weaker exceptions, where longer prompts or responses may reduce the sparse-activation advantage by increasing serving overhead and expert activity. LongBench summarization quality scores do not show a clear trend due to the reason mentioned in .

Finally, the Gemma GPU-generation study in Figure 27 follows the same pattern as the Qwen3 case: newer GPUs reduce QNBI by sustaining higher throughput and avoiding inefficient multi-GPU execution.

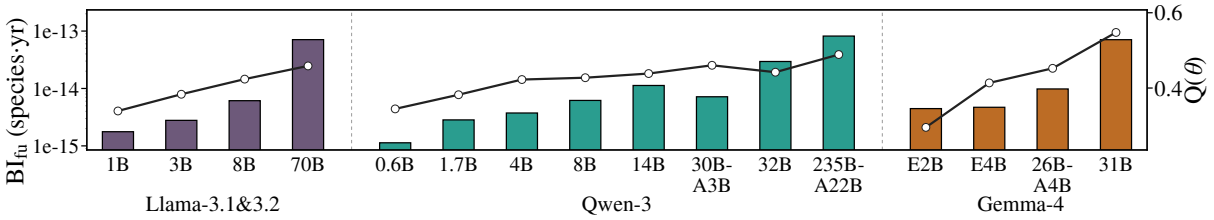

Figure 11: BI_{fu} and quality score for CrossCodeEval.

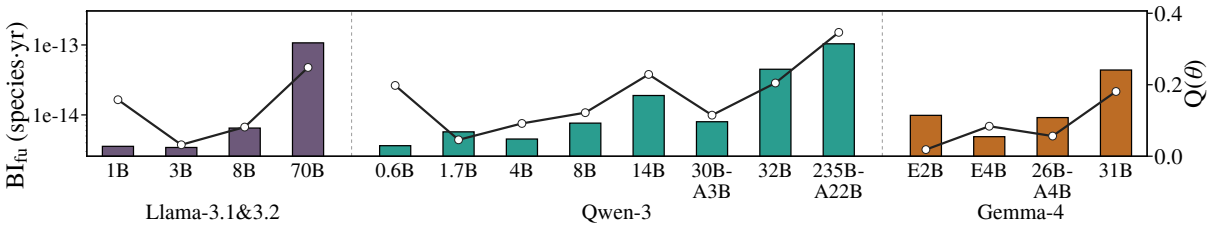

Figure 12: BI_{fu} and quality score for RepoBench.

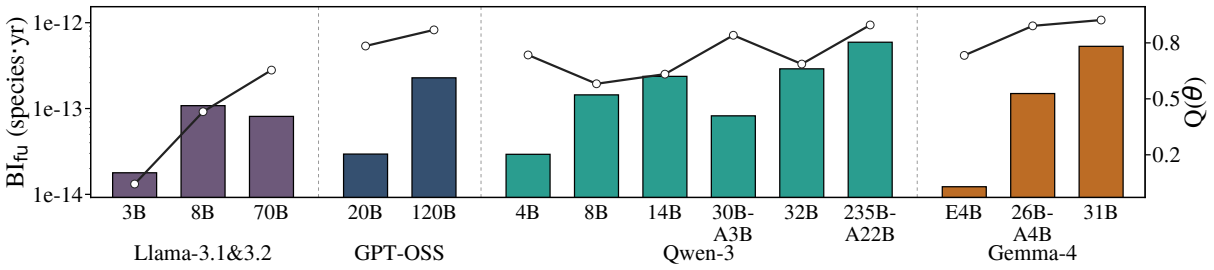

Figure 13: BI_{fu} and quality score for MMLU-Pro.

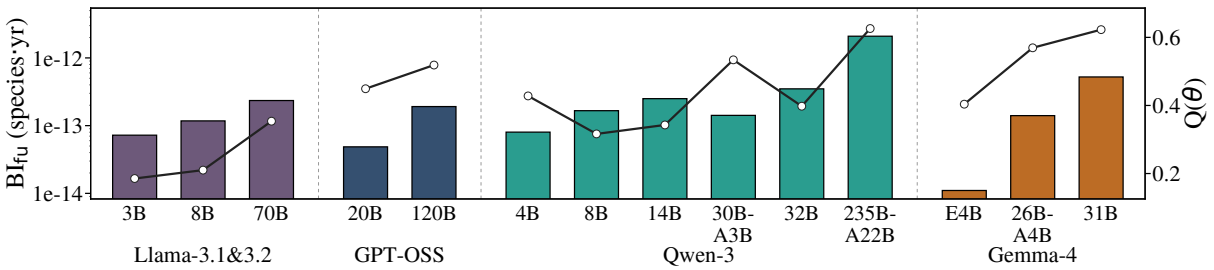

Figure 14: BI_{fu} and quality score for SuperGPQA.

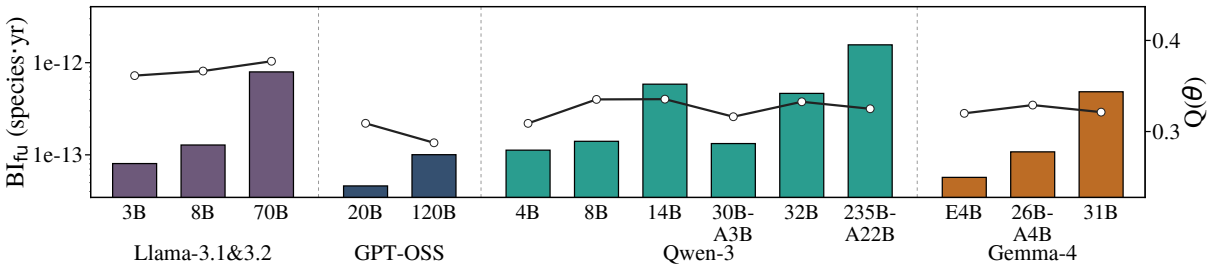

Figure 15: BI_{fu} and quality score for LongBench long-output summarization.

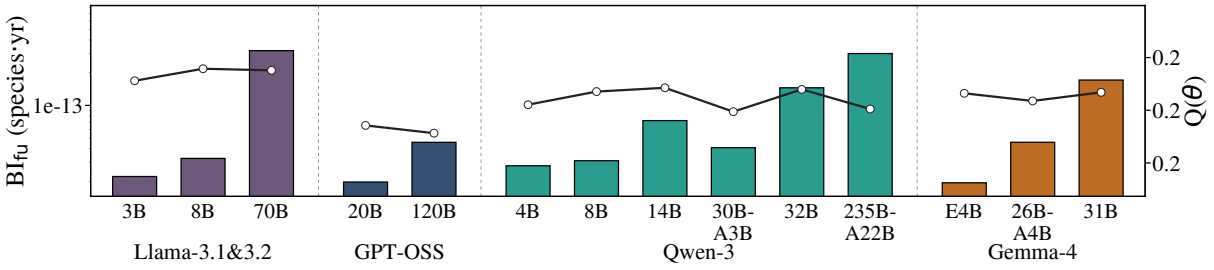

Figure 16: BI_{fu} and quality score for LongBench medium-output summarization.

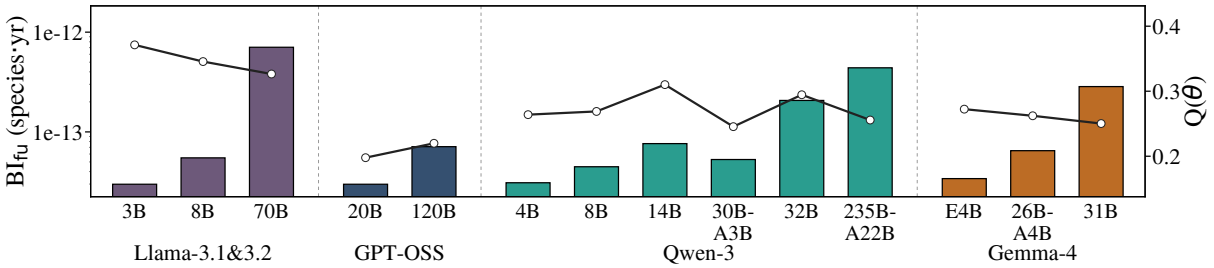

Figure 17: BI_{fu} and quality score for LongBench medium-answer RAG QA.

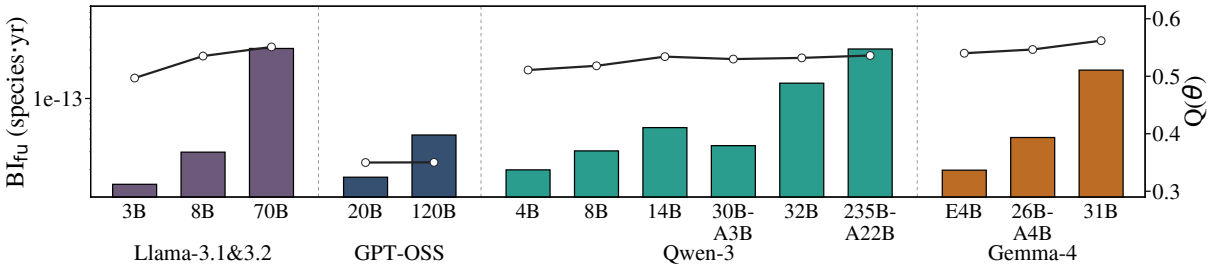

Figure 18: BI_{fu} and quality score for LongBench short-answer document QA.

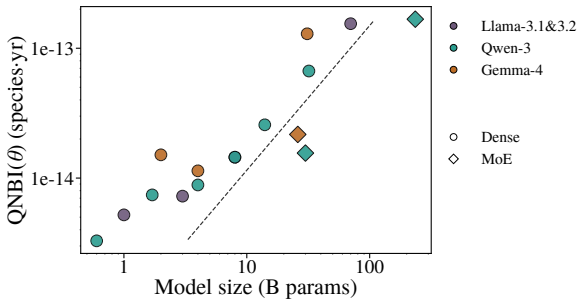

Figure 19: QNBI for CrossCodeEval.

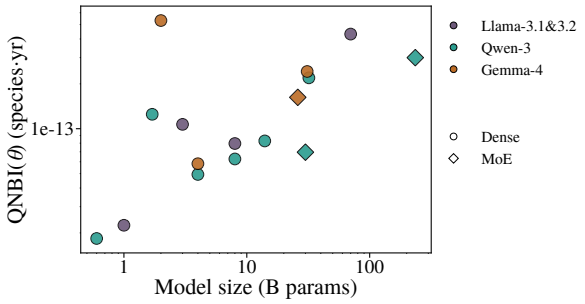

Figure 20: QNBI for RepoBench.

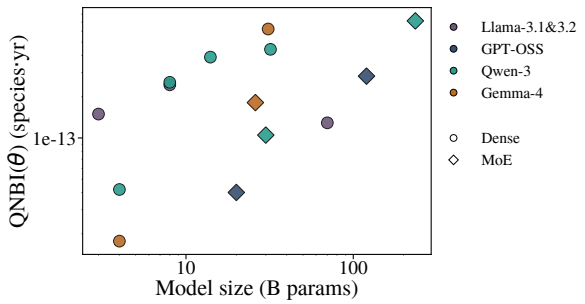

Figure 21: QNBI for MMLU-Pro.

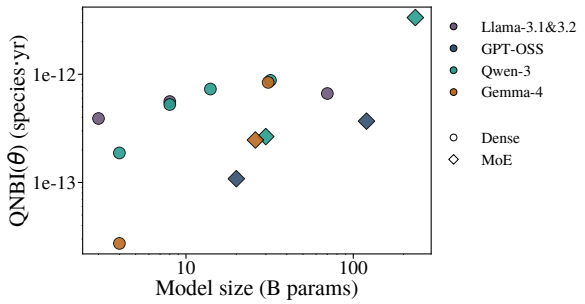

Figure 22: QNBI for SuperGPQA.

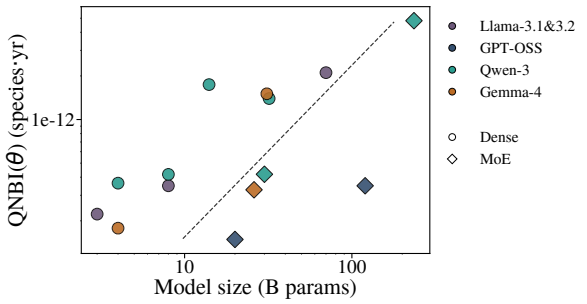

Figure 23: QNBI for LongBench long-output summarization.

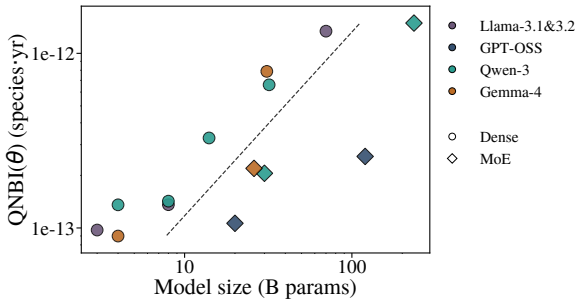

Figure 24: QNBI for LongBench medium-output summarization.

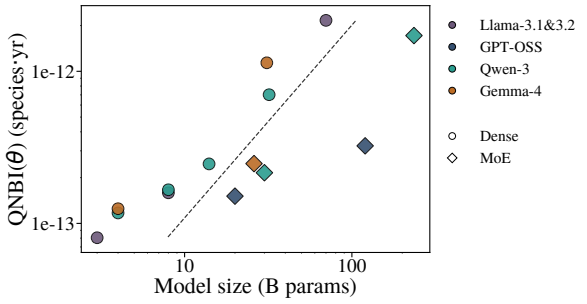

Figure 25: QNBI for LongBench medium-answer RAG QA.

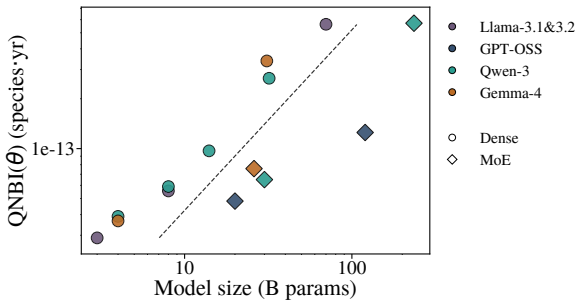

Figure 26: QNBI for LongBench short-answer document QA.

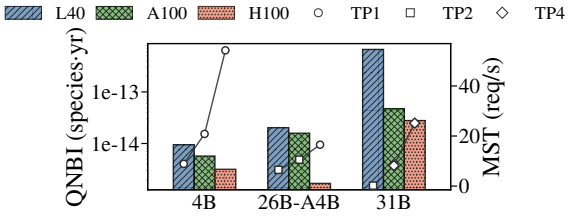

Figure 27: GPU-generation effect on QNBI for Gemma models on ShareGPT.