arXiv:2511.07885v7 [cs.DC] 1 Oct 2026

Intelligence per Watt:
Measuring Intelligence Efficiency of Local AI

Jon Saad-Falcon[*1] · Avanika Narayan[*1] · Hakki Orhun Akengin[1]
J. Wes Griffin[1] · Herumb Shandilya[1] · Adrian Gamarra Lafuente[1]
Medhya Goel[1] · Rebecca Joseph[1] · Shlok Natarajan[1]
Etash Kumar Guha[1] · Shang Zhu[2] · Ben Athiwaratkun[2]
John Hennessy[1] · Azalia Mirhoseini[1] · Christopher Ré[1]

[*] Equal contribution. [1] Stanford University [2] Together AI

Large language model (LLM) queries are predominantly processed by frontier models in centralized cloud infrastructure. Rapidly growing demand strains this paradigm, and cloud providers struggle to scale infrastructure at pace. Two advances create an opportunity to rethink this paradigm: small, local LMs ($\leq$ 20B active parameters) now achieve competitive performance to frontier models on many tasks, and local accelerators (e.g., Apple M4 Max) can host these models at interactive latencies. This raises the question: *can local inference viably redistribute demand from centralized infrastructure?* Answering this requires measuring both whether local LMs can accurately answer real-world queries *and* whether they can do so efficiently enough to be practical on power-constrained devices (i.e., laptops). We propose *intelligence per watt* (IPW), task accuracy divided by unit of power, as a unified metric for assessing both the capability and efficiency of local inference across model-accelerator configurations. We conduct a large-scale empirical study across 20+ state-of-the-art local LMs, 8 hardware accelerators (local and cloud), and a representative subset of LLM traffic: 1M real-world single-turn chat and reasoning queries. For each query, we measure accuracy (local LM win rate against frontier models), energy consumption, latency, and power. Our analysis reveals three key findings. **First**, local LMs can successfully answer 88.7% of single-turn chat and reasoning queries with accuracy varying by domain. **Second**, longitudinal analysis from 2023-2025 shows progress in local inference viability: IPW improved $5.3\times$, driven by both algorithmic advances and accelerator improvements, with locally-serviceable query coverage increasing from 23.2% to 71.3%. **Third**, local accelerators achieve at least $1.4\times$ lower IPW than cloud accelerators running identical models, revealing significant headroom for local accelerator optimization. These findings demonstrate that local inference can meaningfully redistribute demand from centralized infrastructure for a substantial subset of queries, with IPW serving as the critical metric for tracking this transition.

Code: https://github.com/HazyResearch/intelligence-per-watt
Website: https://hazyresearch.stanford.edu/intelligence-per-watt/

1 Introduction

Large language model (LLM) queries are predominantly processed by frontier models deployed in centralized cloud infrastructure [55, 2]. This centralized approach faces mounting resource constraints as inference workloads scale from billions to trillions of queries daily [2]. History suggests an alternative path forward. From 1946-2009, computing efficiency (performance-per-watt) doubled every 1.5 years [37], enabling a redistribution of computing workloads from data center mainframes to personal computers. This transition occurred when efficiency improvements enabled computing to meet user needs within personal device power constraints, not when PCs surpassed mainframes in raw performance.

Three converging trends suggest a similar inflection point may be emerging for LLM inference. First, recent advances have produced *local LMs*: small models ($\leq 20B$ active parameters) such as QWEN3 [62], LLAMA3.1 [26], and GPT-OSS [1] that achieve competitive performance on many benchmarks while requiring less energy and compute than larger, frontier models [1].

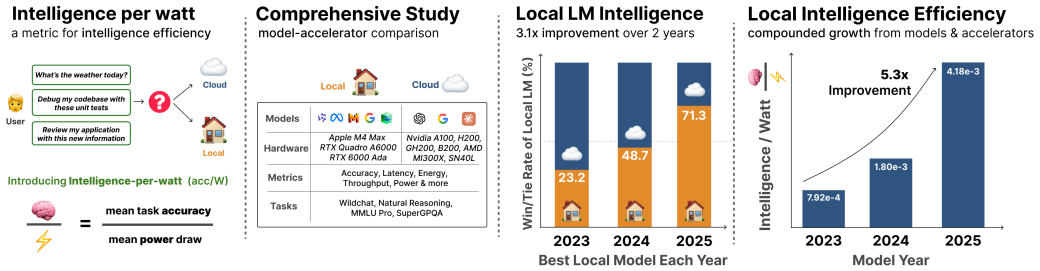


Figure 1: **Intelligence per Watt: A Study of Local Intelligence Efficiency.** We present the *first systematic study of local AI inference efficiency* across models, hardware, and real-world workloads. **(Left)** *Intelligence efficiency is defined as task accuracy per unit of power*, capturing both capabilities delivered and energy consumed. **(Left-Middle)** We conduct *comprehensive performance profiling* across 20+ state-of-the-art local LMs ($\leq 20B$ active parameters), diverse hardware accelerators (APPLE, NVIDIA, AMD), multiple performance metrics, and 1M+ real-world queries spanning chat and reasoning tasks. **(Right-Middle)** *Local LM capabilities are improving rapidly*: win/tie rate versus frontier models increases from 23.2% (2023) to 71.3% (2025), a $3.1\times$ improvement in accuracy, demonstrating that local models can accurately handle significant portions of single-turn chat and reasoning queries. **(Right)** *Intelligence per watt improves $5.3\times$ from 2023–2025*, driven by advances in both model architectures and hardware accelerators, with local accelerators showing $1.5\times$ efficiency headroom compared to enterprise-grade systems.

Second, local accelerators (e.g., Apple M4 Max, AMD Ryzen AI) now have sufficient memory capacity and compute throughput to host these models with interactive latencies [8]. Third, a wave of open-source personal-AI agent stacks designed for on-device execution (e.g., OPENCLAW [74], HERMES AGENT [44], OPENJARVIS [63]), PICOCLAW [71], and ZEROCLAW [88] has emerged, reflecting growing interest in local-first system design. This raises the question: *Can local inference viably redistribute demand from centralized infrastructure?*

Answering this requires measuring two factors: the *capability* of local LMs to accurately respond to a subset of real-world queries, and the *efficiency* with which local accelerators convert power into useful computation. To assess this, we need a unified metric that captures both the intelligence delivered (model capability) and the energy required (accelerator efficiency). We introduce *intelligence per watt* (IPW): task accuracy per unit of power consumption. IPW directly measures the fundamental tradeoff facing local inference: achieving sufficient task performance within constrained power budgets. This metric enables systematic comparison across model-accelerator configurations and quantifies efficiency gains from model architecture innovations [62, 1, 24, 35], post-training techniques [30, 57, 68, 22], and accelerator improvements [39, 51, 4].

To evaluate the viability of local inference and measure progress in IPW, we conduct a large-scale empirical study addressing three questions:

- **Q1:** What fraction of current inference queries can be solved by local LMs on local accelerators, and how has this changed over time?
- **Q2:** How has intelligence per watt improved across successive generations of local models and accelerators, and what are the relative contributions of model versus accelerator advances?
- **Q3:** What resource savings (e.g. compute, energy, dollar cost) are possible by distributing workloads across local and cloud infrastructure?

Our study evaluates 20+ local LMs across 8 hardware accelerators on 1M queries spanning naturalistic user conversations [20], general reasoning tasks [87], and standardized benchmarks measuring knowledge breadth (MMLU PRO [80]) and expert-level reasoning (SUPERGPQA [59]). We focus on single-turn interactions because they constitute a sub-

stantial portion of LLM usage [21, 79, 70].[1] We compare state-of-the-art local LMs from October 2025 (QWEN3, GPT-OSS, GEMMA3, IBM GRANITE4) alongside 2023-2024 models (MIXTRAL-8X7B, LLAMA-3.1-8B) on NVIDIA, AMD, and APPLE accelerators, measuring accuracy, latency, energy, compute, cost, and memory per query (Section 3). We release our hardware-agnostic profiling harness to support reproducible efficiency benchmarking.

Our work makes three primary contributions. **(1)** We introduce intelligence per watt as a unified metric for evaluating local inference viability, and conduct the first large-scale empirical study measuring its evolution across 1M+ queries, 20+ models, and 8 hardware accelerators spanning 2023-2025. **(2)** *(Q1, Q2)* We demonstrate that 88.7% of single-turn chat and reasoning queries can be successfully handled by small local models (with coverage varying by domain), and that IPW has improved $5.3\times$ over two years through compounding model ($3.1\times$) and hardware ($1.7\times$) advances.[2] **(3)** *(Q3)* We show that hybrid local-cloud routing yields 60–80% reductions in energy, compute, and cost compared to a batched cloud baseline; even an 80%-accurate router (a realistic target) captures $\sim 80\%$ of oracle gains while maintaining answer quality. Together, these findings establish local inference as a practical complement to centralized infrastructure whose viability continues expanding.

2 Preliminaries

We formalize local and cloud inference infrastructure and introduce metrics for measuring intelligence efficiency.

Inference Infrastructure: Queries, Models and Accelerators. We consider an inference infrastructure serving a stream of user queries $\mathcal{Q} = \{q_1, q_2, \ldots, q_n\}$, where each query q_i represents a user-generated request (e.g., chat messages, reasoning tasks). Let $\mathcal{M}_{\text{local}} = \{m_1, \ldots, m_k\}$ denote a set of *local LMs* with $\leq 20B$ active parameters each, and $\mathcal{M}_{\text{cloud}} = \{M_1, \ldots, M_\ell\}$ denote *frontier LMs* with $\geq 100B$ parameters. Similarly, let $\mathcal{H}_{\text{local}}$ represent *local accelerators* (e.g., APPLE M4, AMD RYZEN) and $\mathcal{H}_{\text{cloud}}$ represent *cloud accelerators* (e.g., NVIDIA H200, AMD MI300X). We define $\mathcal{M}_{\text{local}}$ by active parameters rather than total parameters because per-query inference efficiency depends on parameters touched per forward pass; total parameters instead govern storage, a separate constraint we verify per accelerator.

Inference Serving: Local and Cloud. We distinguish between two inference paradigms: *local inference*, where queries are processed by models $m \in \mathcal{M}_{\text{local}}$ on accelerator $h \in \mathcal{H}_{\text{local}}$, and *cloud inference*, where queries are processed by models $M \in \mathcal{M}_{\text{cloud}}$ on accelerator $H \in \mathcal{H}_{\text{cloud}}$. A *routing function* $r : \mathcal{Q} \to \mathcal{M}_{\text{local}} \cup \mathcal{M}_{\text{cloud}}$ assigns each query to either a local model (up to 20B active parameters) or a cloud model (at least 100B parameters).

Intelligence Efficiency Metrics. We introduce a family of metrics to quantify how efficiently inference systems convert energy into useful computation. For a model-accelerator pair (m, h), let $\text{acc}(m, q)$ denote the accuracy of model m on query q, $\text{ppl}(m, q)$ denote the perplexity, $P(m, h, q)$ denote the average power consumption (in watts) during inference for query q, and $\tau(m,h,q)$ denote the total latency (in seconds) for generating the response, including both prefill and decoding phases.

We define four complementary efficiency metrics:

Power-based metrics measure efficiency relative to instantaneous power draw:

- *Accuracy per watt*: $\text{APW}(m,h) = \frac{\mathbb{E}_{q\sim\mathcal{Q}}[\text{acc}(m,q)]}{\mathbb{E}_{q\sim\mathcal{Q}}[P(m,h,q)]}$
- *Perplexity per watt*: $\text{PPW}(m,h) = \frac{1}{\mathbb{E}_{q\sim\mathcal{Q}}[\text{ppl}(m,q)]\cdot\mathbb{E}_{q\sim\mathcal{Q}}[P(m,h,q)]}$

[1] We report a multi-turn extension on GAIA and TerminalBenchV2 in App. E.12 confirming the qualitative patterns generalize.

[2] The corresponding per-joule decomposition (Figure 3) yields $18.0\times$ overall, $3.1\times$ model and $5.9\times$ hardware, since hardware progress disproportionately reduces latency. Both MIXTRAL-8X7B (47B total, $\sim$12.9B active per token) and GPT-OSS-120B (120B total, $\leq$20B active per token) are mixture-of-experts models; our $\leq$20B threshold refers to *active* parameters per forward pass, which determine per-query power and latency. Total parameter count governs storage: at FP4, MIXTRAL-8X7B fits within 24 GB GDDR6 (Quadro RTX 6000), and GPT-OSS-120B fits within 128 GB unified memory (Apple M4 Max).

Energy-based metrics measure efficiency relative to total energy consumed per query:

- *Accuracy per joule*:
  $\text{APJ}(m,h) = \frac{\mathbb{E}_{q\sim\mathcal{Q}}[\text{acc}(m,q)]}{\mathbb{E}_{q\sim\mathcal{Q}}[P(m,h,q)\cdot\tau(m,h,q)]}$
- *Perplexity per joule*:
  $\text{PPJ}(m,h) = \frac{1}{\mathbb{E}_{q\sim\mathcal{Q}}[\text{ppl}(m,q)]\cdot\mathbb{E}_{q\sim\mathcal{Q}}[P(m,h,q)\cdot\tau(m,h,q)]}$

where $P(m,h,q)\cdot\tau(m,h,q)$ represents the energy consumption (in joules) for processing query q.

Power-based metrics (APW, PPW) capture the instantaneous efficiency of the inference system, reflecting the hardware's ability to deliver performance at a given power draw. Energy-based metrics (APJ, PPJ) capture the total efficiency per query, accounting for both power consumption and generation latency. Together, these metrics provide a comprehensive view of inference efficiency: *intelligence per watt* quantifies the steady-state efficiency of model-accelerator pairs, while *intelligence per joule* quantifies the end-to-end efficiency from a user's perspective, including the time cost of generation. We report results across all four metrics throughout the paper (Table 2, Figures 3 and 8, Tables 13–14) so that conclusions are robust to the choice of formulation; relative rankings and qualitative trends are preserved across IPW, IPJ, PPW, and PPJ, while absolute rates of change differ in informative ways (e.g., per-joule gains exceed per-watt gains because hardware progress reduces both power draw and generation latency). Throughout the main text we use *accuracy* as a binary indicator $\text{acc}(m,q)\in\{0,1\}$: for benchmarks with ground-truth answers (MMLU PRO, SUPERGPQA, NATURALREASONING) we use exact-match correctness, while for open-ended chat queries (WILDCHAT) we follow established practice in chat evaluation [14] and define $\text{acc}(m,q)=1$ whenever the LLM-judge verdict is `[[A>B]]`, `[[A≫B]]`, or `[[A=B]]` (i.e., the local model wins or ties against the frontier reference).

3 Dataset and Profiling Harness

In this section, we provide details on the dataset selection and profiling harness.

3.1 Dataset Selection

Query Curation We curate over $1M$ queries across four complementary benchmarks designed to measure both naturalistic deployment scenarios and controlled capability assessment. To ensure our findings about local inference efficiency generalize across task distributions, we combine naturalistic queries that reflect real-world LLM usage patterns with standardized benchmarks that enable systematic evaluation of knowledge breadth and reasoning capabilities across diverse domains.

For *naturalistic chat tasks*, we source queries from WILDCHAT [20]: a dataset of $1M$ real ChatGPT prompts, spanning 1 month of user traffic. For *general reasoning tasks*, we source queries from NATURALREASONING [87], which provides approximately 1.2 million reasoning-focused queries spanning diverse domains including mathematics, physics, and chemistry. For *standardized knowledge evaluation*, we use MMLU PRO [80]: an enhanced version of MMLU with increased difficulty (10 vs. 4 answer choices) and improved robustness to prompt variations, measuring multi-domain knowledge understanding. For *expert-level reasoning across specialized disciplines*, we evaluate on SUPERGPQA [59]: a comprehensive benchmark spanning 285 graduate-level disciplines with emphasis on technical domains and specialized fields underrepresented in typical evaluations (e.g., light industry, agriculture, service sciences).

We perform robust data cleaning and filtering (see App. B.1) on each dataset before sampling queries: $500K$ from WILDCHAT, $500K$ from NATURALREASONING, $12K$ from MMLU PRO, and $26.5K$ from SUPERGPQA (see Table 1). Furthermore, we use GPT-4O-MINI to annotate each query with a category from the Anthropic Economic Index [27], which maps AI queries to occupations in the U.S. Department of Labor's O*NET. We consider 22 categories, spanning "Architecture and Engineering" to "Healthcare Support" (full list and category breakdown in App. B.1, Table 3).

Dataset Origin	Category	$\lvert N \rvert$
WILDCHAT	Chat	500K
NATURALREASONING	Reasoning	500K
MMLU PRO	Knowledge	12K
SUPERGPQA	Grad. Reasoning	26.5K

Category	Items
Model Families	QWEN3, GPT-OSS, GEMMA, IBM GRANITE 4.0
Accelerators	NVIDIA A100, H200, GH200, B200, QUADRO RTX 6000, RTX 6000 ADA, AMD MI300X, APPLE M4 MAX, SAMBANOVA SN40L

Table 1: **Dataset Overview. (Left)** Query composition with sizes. **(Right)** Models and Accelerators.

Hardware Accelerators We profile diverse accelerators spanning local, workstation, and datacenter tiers: the NVIDIA A100 40 GB SXM4 (AMPERE) [45], NVIDIA H200 SXM (HOPPER) [46], NVIDIA GH200 GRACE HOPPER SUPERCHIP [49], NVIDIA B200 (BLACKWELL) [50], NVIDIA QUADRO RTX 6000 [47], NVIDIA RTX 6000 Ada [48], AMD INSTINCT MI300X (CDNA 3, OAM) [3], SambaNova SN40L [64] and APPLE MAC STUDIO (M4 MAX) [8]. We additionally evaluate a smartphone-class accelerator (APPLE A18 PRO on iPhone 16 Pro) in App. E.11. These systems were chosen because of their different memory capacities (ranging from 40 GB to 768 GB), memory bandwidth (from 546 GB/s to 8 TB/s), and power consumption (145W to 1000W) (see Table 7 for more details).

Models We collect model generations over the QWEN3 [62], GPT-OSS [1], GEMMA3 [24], and IBM GRANITE 4.0 [35] families. For QWEN3, we use QWEN3-4B, QWEN3-8B, QWEN3-14B, QWEN3-32B, and QWEN3-235B. For GPT-OSS, we consider the GPT-OSS-20B and GPT-OSS-120B models. For the GEMMA3 family, we use GEMMA3 1B INSTRUCT, GEMMA3 4B INSTRUCT, and GEMMA3 12B INSTRUCT models. For IBM GRANITE 4.0, we use GRANITE-4.0-H-MICRO, GRANITE-4.0-H-TINY, and GRANITE-4.0-H-SMALL models. We evaluate state-of-the-art cloud models as of October 2025, including CLAUDE SONNET 4.5 [6], GEMINI 2.5 PRO [17], and GPT-5 (2025-08-07) [54]. For our longitudinal analysis, we evaluate MIXTRAL-8X7B [36] and LLAMA3.1-8B [26]. For each model, we generate responses across all dataset queries on each of the hardware backends. Full details of inference hyperparameters can be found in App. B.1.

Metrics For each `(query, model, hardware)` triple we collect accuracy plus efficiency metrics: latency, throughput, time-to-first-token (TTFT), and more (see Table 6) We use LLM-as-a-judge (prompts in App. B.1) to score generated responses against reference answers. For WILDCHAT, reference answers are responses from QWEN3-235B, the SOTA open-source model on LMArena (as of August 2025) [14]. For NATURALREASONING, MMLU PRO, and SUPERGPQA, we use the provided ground truth answers from each benchmark.

3.2 Profiling Harness

We develop an end-to-end, cross-platform profiling harness for inference workloads that ensures reproducible results and easily accommodates new models, tasks, and hardware backends. It comprises three components (distributed multi-GPU inference, response evaluation, and system-level telemetry collection) and currently supports NVIDIA, MACOS (Apple Silicon), and AMD systems. Given a dataset, model, and backend, the harness orchestrates inference over all input queries, evaluates outputs (via exact match or LLM-as-a-judge), and records detailed telemetry: latency, throughput, time-to-first-token (TTFT), energy consumption, and more (Table 6). Telemetry is collected via vendor APIs, synchronized at nanosecond resolution, and normalized to common units (watts, joules, megabytes). For energy measurements, we follow standard practices [65, 23, 82]. On NVIDIA systems we query NVML for per-device power, energy, memory usage, and temperature (accelerator-only scope); on AMD systems we query ROCm SMI for power, temperature, and VRAM usage (accelerator-only scope); on MACOS systems we extract GPU power from `powermetrics` (`processor_power.actual` on Apple Silicon, isolating the GPU subsystem rather than full SoC package power) so that all per-query power measurements correspond to the AI-accelerator subsystem on each platform. In all cases, we compute energy via numerical integration over time and sample at 50 ms intervals, providing higher temporal resolution than prior work (100 ms [65] or 15 s [23]). For multi-GPU configurations, we aggregate energy from each GPU individually rather than extrapolating from a single device [65]. We use a custom harness rather than CODECARBON [16], which similarly relies on NVML and

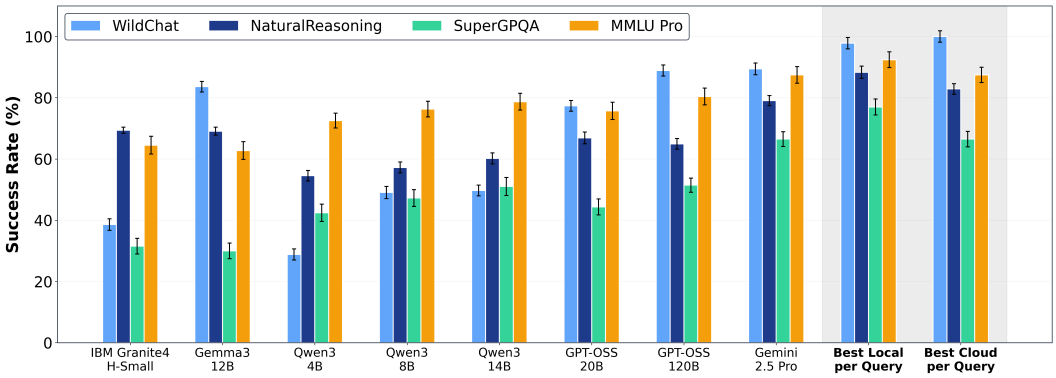


Figure 2: **Local Models Rival Cloud Models Across Diverse Benchmarks:** Individual model performance scales with size, ranging from 31.5–69.4% for IBM GRANITE4-H-SMALL, 30.0–83.6% for GEMMA3-12B, 51.5–80.4% for GPT-OSS-120B, and 66.5–89.5% for GEMINI 2.5 PRO. Local routing (best local LM per query) achieves 97.8%, 88.3%, 77.0%, and 92.4% on WILDCHAT, NATURALREASONING, SUPERGPQA, and MMLU PRO respectively, surpassing cloud routing (100%, 82.9%, 66.5%, 87.4%) on three of four benchmarks.

RAPL, primarily for two reasons: (i) CODECARBON's default sampling cadence (15 s) is too coarse for fine-grained per-query attribution on short generations, and (ii) our cross-platform requirements include AMD (ROCm SMI) and Apple Silicon (`powermetrics`) telemetry that CODECARBON does not natively support. Software-based power measurements can introduce inaccuracies of 10–15%, with variations distributed across different hardware components due to architectural differences in workloads between CPUs, GPUs, and NPUs [85]. Even hardware wattage meters may fall short for milliwatt-level precision, though our approach aligns with established practices and provides consistent relative comparisons across configurations. Full implementation details are provided in App. B.1.

4 Intelligence Efficiency Study

We investigate whether recent advances in local LMs and local accelerators enable local inference to viably complement centralized cloud infrastructure by handling a substantial fraction of inference queries. Using our curated dataset, we examine three interconnected questions: **(1)** the extent to which current workloads can be handled locally (Section 4.1), **(2)** how intelligence efficiency has evolved from 2023–2025 (Section 4.2), and **(3)** what gains query routing across local and cloud models can deliver in practice (Section 4.3). We use single-query inference (bs $= 1$) to isolate intrinsic model-accelerator efficiency from system-level scheduling and follow standard local-inference benchmarking practice [28]; the §4.3 routing simulation is the exception, running the cloud baseline at bs$=16$ to reflect production serving.

4.1 Can Local Models and Accelerators Handle Current Inference Workloads? *(Q1)*

We measure *query coverage* (the percentage of dataset queries answered correctly) across three configurations (Figure 2): individual local LMs, the best-of-local ensemble (routing to the best local LM), and the best-of-cloud baseline (routing to the best frontier model). Our findings are as follows:

Local LM coverage increases with scale and time. Across WILDCHAT, NATURALREASONING, SUPERGPQA, and MMLU PRO, individual model coverage ranges from 49.6% for QWEN3-4B, on average, to 71.4% for GPT-OSS-120B, with consistent improvements at each scale point: QWEN3-8B achieves 57.5% and QWEN3-14B reaches 60.0%. Coverage has improved substantially from 2023 to 2025 (Figure 12): the best local LMs achieved a 32.2% relative improvement on chat queries and a 50.1% relative improvement on reasoning queries over this period. While improvements from 2023-2025 are relatively uniform across difficulty levels for chat tasks, reasoning tasks show markedly slower progress on the hardest problems (see App E).

	2023	2024	2025
SOTA Local Model	Mixtral-8x7B-v0.1	Llama-3.1-8B-Instruct	GPT-OSS-120B
SOTA Accelerator	NVIDIA Quadro RTX 6000	NVIDIA RTX 6000 Ada	Apple M4 Max
Success Rate	$23.2 \pm 1.9\%$	$48.7 \pm 2.7\%$	$71.3 \pm 2.2\%$
Intelligence per Watt	$(7.92 \pm 0.32) \times 10^{-4}$	$(1.80 \pm 0.21) \times 10^{-3}$	$(4.18 \pm 0.53) \times 10^{-3}$
YoY Efficiency Gain	—	2.27×	2.32×

Table 2: **Increase in Intelligence per Watt for Local LMs**: Accuracy per watt has improved over $5\times$ in two years, driven by advances in both model architectures (from MIXTRAL-8X7B to GPT-OSS-120B) and accelerator hardware (from NVIDIA Quadro RTX 6000 to Apple M4 Max). Values are mean $\pm$ 1-σ standard deviation across measurement runs.

These results demonstrate that larger local LMs can handle progressively more queries without requiring cloud infrastructure, with the best individual local LM (GPT-OSS-120B) successfully answering almost three-fourths of the single-turn chat and reasoning queries studied.

Model diversity substantially improves coverage. Routing queries to the most appropriate local LM rather than using a single model achieves 88.7% overall coverage, a 28.8 percentage point improvement over QWEN3-14B and 16.3 percentage points over individual GPT-OSS-120B performance, on average. This gap between individual models and best-of-local demonstrates that architectural, pretraining, and post-training diversity captures complementary capabilities: different models excel on different query types, and intelligent routing can exploit these complementary strengths. On reasoning benchmarks, best-of-local even surpasses best-of-cloud (Figure 2); this is a best-of-N selection effect, since best-of-local selects from $20+$ diverse local models while best-of-cloud selects from three frontier models, and a sufficiently diverse local ensemble can therefore exceed any single frontier model on subsets of queries where local strengths are complementary.

Chat queries are more amenable to local processing than reasoning queries. The best local LM achieves 88.9% coverage on WILDCHAT versus 64.9% on NATURALREASONING, a 24.0 pp gap consistent with findings that 77% of real-world ChatGPT queries involve practical guidance, information seeking, or writing [10]. These tasks are well-suited to local models, while reasoning-intensive queries more often require frontier capabilities for technical domains (Architecture, Engineering, Life & Physical Science; see Figure 5 in App. B). Even on NATURALREASONING/SUPERGPQA/MMLU PRO, local LMs handle over four-fifths of reasoning queries studied, suggesting significant opportunities for local inference even in technically demanding domains.

Evaluation on standardized benchmarks confirms local LM viability across task distributions. On MMLU PRO (multi-domain knowledge) and SUPERGPQA (graduate-level reasoning), best-of-local achieves 93.4% and 83.6% coverage respectively (vs. 80.4% and 51.5% for the best individual local model), with coverage exceeding 93% for creative/humanities fields but dropping to 60% for technical disciplines like Architecture & Engineering (Figure 5), confirming that local LMs handle most conversational and knowledge-recall tasks while complex specialized reasoning still benefits from frontier capabilities.

Local accelerator memory capacity is expanding rapidly. From 2012 to 2025, local accelerator memory grew $\sim 126\times$ (Figure 13 in App. E.8); the jump from sub-20 GB to 200+ GB through unified-memory architectures like Apple Silicon removes the key constraint that previously forced workloads to cloud infrastructure, enabling the 8–20B-active-parameter models that handle the majority of queries today to run efficiently on local hardware.

4.2 How Intelligence Efficient is Local Inference? *(Q2)*

Intelligence efficiency is improving over time. Table 2 tracks the evolution of local LM capabilities from 2023 to 2025, measuring the best available local LM ($\leq 20B$ active

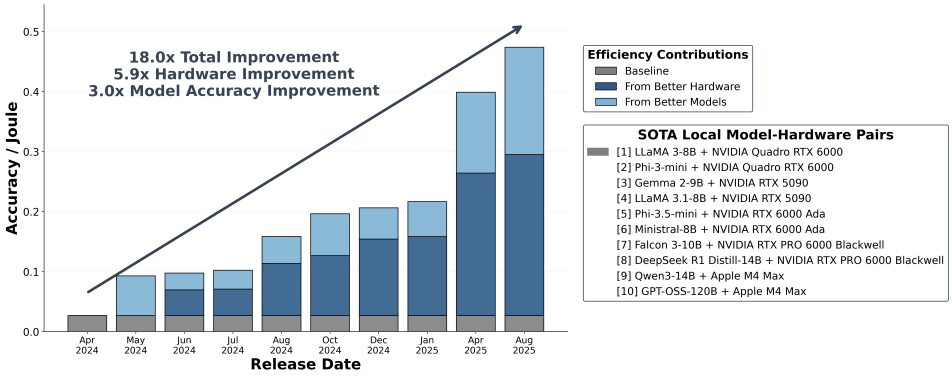


Figure 3: **Increase in Intelligence per Joule for Local LMs and Accelerators**: Efficiency improved $18.0\times$ over 16 months, decomposed into $3.0\times$ from local LMs and $5.9\times$ from local accelerators.

parameters) paired with state-of-the-art accelerators each year. On our curated dataset of chat and reasoning queries, accuracy per watt has improved $5.3\times$ over this two-year period: in 2023, MIXTRAL-8X7B-V0.1 on NVIDIA QUADRO RTX 6000 achieved 7.92×10^{-4} accuracy per watt; by 2024, LLAMA-3.1-8B-INSTRUCT on NVIDIA RTX 6000 ADA reached 1.80×10^{-3} (a $2.27\times$ year-over-year gain); and in 2025, GPT-OSS-120B on APPLE M4 MAX achieved 4.18×10^{-3} (a $2.32\times$ gain). Notably, local LM coverage on single-turn chat and reasoning queries has increased in lockstep with efficiency gains: from 23.2% in 2023 to 48.7% in 2024 to 71.3% in 2025. This progression reflects compounding improvements in both model architectures, which *achieve higher accuracy* through advances in pretraining [15, 32, 53, 18], post-training [9, 68, 19], and parameter utilization via mixture-of-experts (MoE) architectures [69, 18], and hardware accelerators, which deliver *more compute (FLOPs) and memory per watt* [45, 46].

The decomposition depends on the metric: under accuracy-per-watt, model progress contributes 3.1× and hardware 1.7× (Table 2), while under accuracy-per-joule it yields 3.1× and 5.9× respectively (Figure 3). Model progress dominates per-watt efficiency; hardware progress dominates per-joule efficiency, because newer accelerators (HBM3e bandwidth, dedicated tensor units) reduce latency as well as power. Both matter — per-watt for thermally-constrained deployment, per-joule for end-to-end energy budgets — and Figure 3 shows the per-joule trend is consistent across nine model families (LLAMA, PHI, GEMMA, MISTRAL, FALCON, DEEPSEEK, QWEN, GPT-OSS), with perplexity-based confirmation in Figure 8 (App. E.3). Sustained progress in MoE architectures, quantization, and unified-memory capacity is the precondition for these trends to continue. We explore quantization tradeoffs (App. E.5) and serving-stack sensitivity — batching (App. E.9) and framework choice (App. E.10), finding that FP4 saves 3–3.5× energy per precision step, cloud bs=64 gives 11–20× higher IPJ than bs=1, and IPW rankings across vLLM, SGLang, and llama.cpp are preserved.

Local accelerator efficiency has room for improvement: While local accelerators enable deployment of capable models outside data centers, cloud-grade hardware maintains a substantial efficiency advantage on the same workloads. Across QWEN3 and GPT-OSS variants, the NVIDIA B200 achieves $1.40\times$ higher IPW and 1.6–$2.3\times$ higher IPJ than APPLE M4 MAX, and SAMBANOVA SN40L achieves up to $1.78\times$ higher IPW and 6.5–$7.4\times$ higher IPJ (Tables 13–14 in App. E.8). Similar trends are observed on multi-turn agentic workloads (see App. E.12). Per-joule gaps widen relative to per-watt gaps because cloud accelerators not only consume less power per unit of accuracy but also complete queries faster. These gaps stem from purpose-built components in enterprise accelerators (HBM3e, dedicated tensor units, optimized memory hierarchies), whereas local accelerators use unified-memory architectures that balance diverse workloads under thermal and power constraints. For these comparisons we use bs $= 1$ for both local and cloud, so the gap is intrinsic rather than a batching artifact, revealing substantial headroom for on-device AI components.

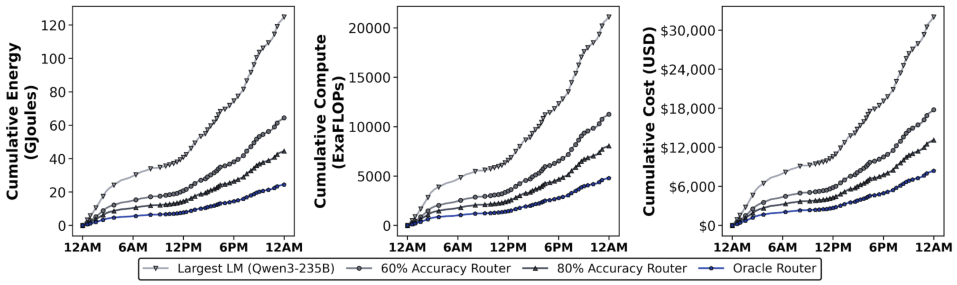


Figure 4: **Energy, Compute, and Capital Gains from Model Routing**. Cumulative resource consumption over 24 hours and 80.2M LLM queries [81], routing between 4 small LMs on Apple M4 Max (bs = 1) and QWEN3-235B on an H200 (bs = 16, batched cloud baseline). The 80%-accurate router achieves 64.3% energy, 61.8% compute, and 59.0% cost savings versus routing all queries to QWEN3-235B, capturing the majority of gains achievable by the Oracle.

4.3 What Efficiency Gains Can Effective Query Routing Deliver? *(Q3)*

We simulate a hybrid local-cloud system serving 80.2M queries over 24 hours, representative of realistic daily inference workloads [81]. Queries are routed between four small local LMs (QWEN3-4B/8B/14B, GPT-OSS-20B) on APPLE M4 MAX (bs = 1) and a frontier model (QWEN3-235B) on NVIDIA H200 (bs = 16, reflecting production serving conditions). Figure 4 compares five strategies: routing all queries to the largest model (baseline), oracle, and realistic routers at 60%/80% routing accuracy; misrouted queries fall back to the cloud model.

Oracle routing establishes theoretical upper bounds. Assuming perfect query-to-model assignment, oracle routing reduces energy by 80.4%, compute by 77.3%, and cost by 73.8% versus cloud-only deployment to the largest model (Figure 4). The 80.7% of queries that local models can handle correctly are assigned to substantially smaller local models, while frontier compute is reserved for the remaining 19.3%. The dominant effect is the reduction in model size, not a hardware efficiency advantage. The cloud-only baseline already operates at bs = 16, so the reported savings are computed against a batched cloud baseline rather than bs = 1 (see App. E.9).

Practical routers achieve substantial gains without perfect accuracy. A router with 80% accuracy (a realistic target, with prior work such as RouteLLM [52] reporting 70–85% on real workloads) captures ∼80% of oracle gains: 64.3% energy, 61.8% compute, and 59.0% cost reduction. Even a 60%-accurate router delivers 48.4%/46.7%/44.5% savings, with misrouted queries falling back to the frontier model so end-to-end answer quality is maintained. The oracle uses ground-truth correctness as a theoretical upper bound; deployable routers instead use *pre-routing* confidence estimation, and the 60%/80% scenarios above explicitly model this imperfection. At billions of queries daily, these savings scale linearly to annual energy savings in terawatt-hours. Although local accelerators are less efficient per query (§4.2), routing inverts this at the system level: efficient AI infrastructure comes not from local hardware matching cloud, but from routing across both.

5 Conclusion and Key Takeaways

We ask whether local inference can viably redistribute demand from centralized cloud infrastructure, and introduce *intelligence per watt* (IPW)—task accuracy per unit of power—as a unified metric for jointly evaluating local LM capability and local accelerator efficiency. We take a first step towards answering this question through a longitudinal study of 20+ models, 8 accelerators, and 1M queries spanning 2023–2025. We find that 88.7% of single-turn chat and reasoning queries can be handled locally, IPW has improved 5.3× over two years through compounding model (3.1×) and hardware (1.7×) advances, and although cloud accelerators retain a 1.4–7.4× per-query efficiency edge (App. E.8), hybrid local-cloud routing yields 60–80% aggregate energy, compute, and cost reductions at realistic routing

accuracy. Our study provides three practical takeaways (App. E.1): MoE architectures deliver the best IPW on memory-rich local devices, aggressive FP4 quantization beats smaller-but-higher-precision models, and router accuracy past ∼80% matters less than expanding the local-model ensemble. We release our profiling harness to support periodic re-evaluation as the local-inference ecosystem evolves, and discuss the scope and limitations of our study in App. C. In future work, we hope to extend IPW characterization to broader workload regimes (i.e., multi-modal inference) and hybrid local-cloud execution patterns, and to push the local AI frontier through model-hardware co-design (App. D).

References

[1] S. Agarwal et al. gpt-oss-120b and gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. URL https://arxiv.org/abs/2508.10925.

[2] Alvarez & Marsal. Rethinking ai demand part 1: Ai data centers are experiencing a surge of training demand - what happens when the surge is over?, 2025. Accessed: 2025-10-06.

[3] AMD. Amd instinct mi300x accelerators — product specifications. AMD official website, 2023. URL https://www.amd.com/en/products/accelerators/instinct/mi300/mi300x.html. 192GB HBM3, ∼5.3TB/s bandwidth, 750W TDP.

[4] AMD. Accelerator specifications. https://www.amd.com/en/products/specifications/accelerators.html, 2025. Accessed: 2025-10-30.

[5] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. In *ICML Workshop on Challenges in Deploying and Monitoring Machine Learning Systems*, 2020. URL https://arxiv.org/abs/2007.03051. arXiv:2007.03051.

[6] Anthropic. System card: Claude sonnet 4.5. System card, Anthropic, September 2025. URL https://assets.anthropic.com/m/12f214efcc2f457a/original/Claude-Sonnet-4-5-System-Card.pdf.

[7] Ruth Appel, Peter McCrory, Alex Tamkin, Michael Stern, Miles McCain, and Tyler Neylon. Anthropic economic index report: Uneven geographic and enterprise ai adoption. Technical report, Anthropic, September 2025.

[8] Apple. Apple m4 max — tech specs. Apple Support / Press Releases, 2024. URL https://support.apple.com/en-us/121553. Unified memory bandwidth up to 546 GB/s.

[9] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[10] Aaron Chatterji, Tom Cunningham, David Deming, Zoë Hitzig, Christopher Ong, Carl Shan, and Kevin Wadman. How people use chatgpt. Technical report, OpenAI, September 2025. Available at: https://cdn.openai.com/pdf/a253471f-8260-40c6-a2cc-aa93fe9f142e/economic-research-chatgpt-usage-paper.pdf.

[11] Lingjiao Chen, Matei Zaharia, and James Y. Zou. Frugalml: How to use ML prediction apis more accurately and cheaply. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/789ba2ae4d335e8a2ad283a3f7effced-Abstract.html.

[12] Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023. URL https://arxiv.org/abs/2305.05176.

[13] Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, 37:66305–66328, 2024.

[14] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024.

[15] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[16] CodeCarbon Contributors. mlco2/CodeCarbon (v2.8.0). Zenodo, 2024. URL https://zenodo.org/doi/10.5281/zenodo.14212766.

[17] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

[18] DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[19] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[20] Yuntian Deng, Wenting Zhao, Jack Hessel, Xiang Ren, Claire Cardie, and Yejin Choi. Wildvis: Open source visualizer for million-scale chat logs in the wild, 2024. URL https://arxiv.org/abs/2409.03753.

[21] Yuyang Deng, Ni Zhao, and Xin Huang. Early chatgpt user portrait through the lens of data. *arXiv preprint arXiv:2312.10078*, 2023.

[22] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332, 2022.

[23] Jared Fernandez, Clara Na, Vashisth Tiwari, Yonatan Bisk, Sasha Luccioni, and Emma Strubell. Energy considerations of large language model inference and efficiency optimizations. *arXiv preprint arXiv:2504.17674*, 2025.

[24] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

[25] Waren Gonzaga. Tinyclaw: The original tiny claw as your personal autonomous ai companion. https://github.com/warengonzaga/tinyclaw, 2026.

[26] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[27] Kunal Handa, Alex Tamkin, Miles McCain, Saffron Huang, Esin Durmus, Sarah Heck, Jared Mueller, Jerry Hong, Stuart Ritchie, Tim Belonax, Kevin K. Troy, Dario Amodei, Jared Kaplan, Jack Clark, and Deep Ganguli. Which economic tasks are performed with ai? evidence from millions of claude conversations, 2025. URL https://arxiv.org/abs/2503.04761.

[28] Jianwei Hao, Piyush Subedi, Lakshmish Ramaswamy, and In Kee Kim. Reaching for the sky: Maximizing deep learning inference throughput on edge devices with ai multi-tenancy. *ACM Transactions on Internet Technology*, 23(1):1–33, 2023.

[29] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020. URL http://jmlr.org/papers/v21/20-312.html.

[30] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[31] HKUDS. Nanobot: The ultra-lightweight personal ai agent. https://github.com/HKUDS/nanobot, 2026.

[32] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030, 2022.

[33] Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*, 2024. URL https://arxiv.org/abs/2403.12031.

[34] Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. *arXiv preprint arXiv:2503.10657*, 2025. doi: 10.48550/arXiv.2503.10657. URL https://arxiv.org/abs/2503.10657.

[35] IBM Research. Granite 4.0 language models. https://github.com/ibm-granite/granite-4.0-language-models, 2025. Accessed: 2025-10-01.

[36] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[37] Jonathan Koomey, Stephen Berard, Marla Sanchez, and Henry Wong. Implications of historical trends in the electrical efficiency of computing. *IEEE Annals of the History of Computing*, 33(3):46–54, 2010.

[38] Xiangchen Li, Dimitrios Spatharakis, Saeid Ghafouri, Jiakun Fan, and Dimitrios Nikolopoulos. Sled: A speculative llm decoding framework for efficient edge serving. *arXiv preprint arXiv:2506.09397*, 2025. URL https://arxiv.org/abs/2506.09397.

[39] Median-Group. numbers. https://github.com/Median-Group/numbers, 2019. GitHub repository. Accessed: 2025-10-30.

[40] Memovai. Mimiclaw: Openclaw on a $5 chip. https://github.com/memovai/mimiclaw, 2026.

[41] Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. Specinfer: Accelerating generative large language model serving with tree-based speculative inference and verification. In *arXiv preprint arXiv:2305.09781*, 2023. URL https://arxiv.org/abs/2305.09781.

[42] Avanika Narayan, Dan Biderman, Sabri Eyuboglu, Avner May, Scott Linderman, James Zou, and Christopher Ré. Minions: Cost-efficient collaboration between on-device and cloud language models. *arXiv preprint arXiv:2502.15964*, 2025. URL https://arxiv.org/abs/2502.15964.

[43] NEAR AI. Ironclaw: An agent os focused on privacy, security, and extensibility. https://github.com/nearai/ironclaw, 2026.

[44] Nous Research. Hermes agent: The agent that grows with you. https://github.com/NousResearch/hermes-agent, 2026. Self-improving agent with a built-in learning loop: autonomous skill creation, cross-session conversation recall, and persistent user memory.

[45] NVIDIA. Nvidia a100 tensor core gpu — data sheet. NVIDIA official documentation, 2021. URL https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf. SXM4 version, ∼2.0TB/s memory bandwidth, 400W.

[46] NVIDIA. Nvidia h200 tensor core gpu — data sheet. NVIDIA official documentation, 2024. URL https://www.nvidia.com/en-us/data-center/h200/. 141GB HBM3e memory, 4.8TB/s bandwidth, up to 700W (SXM variant).

[47] NVIDIA Corporation. *NVIDIA Quadro RTX 6000 Datasheet*. NVIDIA Corporation, March 2019. URL https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/quadro-product-literature/quadro-rtx-6000-us-nvidia-704093-r4-web.pdf. Document 704093-r4.

[48] NVIDIA Corporation. *NVIDIA RTX 6000 Ada Generation Datasheet*. NVIDIA Corporation, February 2023. URL https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/rtx-6000/proviz-print-rtx6000-datasheet-web-2504660.pdf. Document 2647623.

[49] NVIDIA Corporation. NVIDIA Grace Hopper Superchip Architecture, 2024. URL https://resources.nvidia.com/en-us-data-center-overview-mc/en-us-data-center-overview/grace-hopper-superchip-datasheet-partner. Accessed: 2025-01-15.

[50] NVIDIA Corporation. NVIDIA DGX B200 System Architecture, 2025. URL https://resources.nvidia.com/en-us-dgx-systems/dgx-b200-datasheet. Accessed: 2025-01-15.

[51] NVIDIA Corporation. Nvidia data center gpu resource center. https://resources.nvidia.com/l/en-us-gpu, 2025. Accessed: 2025-10-30.

[52] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[53] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[54] OpenAI. Introducing GPT-5. OpenAI Blog, 2025. URL https://openai.com/index/introducing-gpt-5/. Accessed 2025-09-13.

[55] OpenAI. Announcing the stargate project. https://openai.com/index/announcing-the-stargate-project/, January 2025. Accessed: 2025-10-06.

[56] OpenRouter. OpenRouter. https://openrouter.ai, 2025. Accessed: 23 September 2025.

[57] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[58] Felipe Oviedo, Fiodar Kazhamiaka, Esha Choukse, Allen Kim, Amy Luers, Melanie Nakagawa, Ricardo Bianchini, and Juan M. Lavista Ferres. Energy use of ai inference: Efficiency pathways and test-time compute, 2025. URL https://arxiv.org/abs/2509.20241.

[59] P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

[60] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021. URL https://arxiv.org/abs/2104.10350.

[61] Haoran Qiu, Weichao Mao, Archit Patke, Shengkun Cui, Saurabh Jha, Chen Wang, Hubertus Franke, Zbigniew T. Kalbarczyk, Tamer Başar, and Ravishankar K. Iyer. Power-aware deep learning model serving with μ-serve. In *2024 USENIX Annual Technical Conference (USENIX ATC 24)*, 2024. URL https://www.usenix.org/conference/atc24/presentation/qiu.

[62] Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[63] Jon Saad-Falcon, Avanika Narayan, Herumb Shandilya, Hakki Orhun Akengin, Robby Manihani, Gabriel Bo, John Hennessy, Christopher Ré, and Azalia Mirhoseini. Openjarvis: Personal ai, on personal devices. https://scalingintelligence.stanford.edu/blogs/openjarvis/, 2026.

[64] SambaNova Systems. Sambanova unveils new ai chip, the SN40L, powering its full stack ai platform. Announcement, SambaNova Systems, Inc., Palo Alto, California, September 2023. URL https://sambanova.ai/blog/sn40l-chip-best-inference-solution. Fourth-generation Reconfigurable Dataflow Unit (RDU); 5nm TSMC, three-tier memory architecture. Accessed: November 10, 2025.

[65] Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–9. IEEE, 2023.

[66] Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022.

[67] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *Communications of the ACM*, 63(12):54–63, 2020. doi: 10.1145/3381831. URL https://dl.acm.org/doi/10.1145/3381831.

[68] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[69] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

[70] Trevin Shirey. How people use chatgpt: Stats from 13,252 conversations. https://www.webfx.com/blog/ai/chatgpt-usage-statistics/, September 2025. Accessed: November 2025.

[71] Sipeed. Picoclaw: Tiny, fast, and deployable anywhere. https://github.com/sipeed/picoclaw, 2026.

[72] Seamus Somerstep, Felipe Maia Polo, Allysson Flavio Melo de Oliveira, Prattyush Mangal, Mírian Silva, Onkar Bhardwaj, Mikhail Yurochkin, and Subha Maity. Carrot: A cost aware rate optimal router, 2025. URL https://arxiv.org/abs/2502.03261.

[73] Yixin Song, Zeyu Mi, Haotong Xie, and Haibo Chen. Powerinfer: Fast large language model serving with a consumer-grade GPU. In *Proceedings of the ACM SIGOPS 30th Symposium on Operating Systems Principles (SOSP '24)*. ACM, 2024. doi: 10.1145/3694715.3695964. URL https://dl.acm.org/doi/10.1145/3694715.3695964.

[74] Peter Steinberger. OpenClaw: Open-source personal AI assistant. https://github.com/openclaw/openclaw, 2026. 131K+ GitHub stars as of March 2026. Originally released as Clawdbot (Nov. 2025), briefly renamed Moltbot before becoming OpenClaw in January 2026.

[75] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 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, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1355. URL https://aclanthology.org/P19-1355.

[76] Thierry Tambe, Coleman Hooper, Lillian Pentecost, Tianyu Jia, En-Yu Yang, Marco Donato, Victor Sanh, Paul N. Whatmough, Alexander M. Rush, David Brooks, and Gu-Yeon Wei. Edgebert: Sentence-level energy optimizations for latency-aware multi-task NLP inference. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2021. doi: 10.1145/3466752.3480095. URL https://dl.acm.org/doi/10.1145/3466752.3480095.

[77] Arya Tschand, Arun Tejusve Raghunath Rajan, Sachin Idgunji, Anirban Ghosh, Jeremy Holleman, Csaba Kiraly, Pawan Ambalkar, Ritika Borkar, Ramesh Chukka, Trevor Cockrell, Oliver Curtis, Grigori Fursin, Miro Hodak, Hiwot Kassa, Anton Lokhmotov, Dejan Miskovic, Yuechao Pan, Manu Prasad Manmathan, Liz Raymond, Tom St. John, Arjun Suresh, Rowan Taubitz, Sean Zhan, Scott Wasson, David Kanter, and Vijay Janapa Reddi. MLPerf power: Benchmarking the energy efficiency of machine learning systems from μwatts to mwatts for sustainable AI. In *2025 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 1201–1216, Las Vegas, NV, USA, 2025. IEEE. doi: 10.1109/HPCA61900.2025.00092.

[78] U.S. Bureau of Economic Analysis. GDP by industry, 2024. URL https://www.bea.gov/data/gdp/gdp-industry.

[79] Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: Evaluating LLMs in multi-turn interaction with tools and language feedback. In *The Twelfth International Conference on Learning Representations*, 2024.

[80] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

[81] Yuxin Wang, Yuhan Chen, Zeyu Li, Xueze Kang, Yuchu Fang, Yeju Zhou, Yang Zheng, Zhenheng Tang, Xin He, Rui Guo, Xin Wang, Qiang Wang, Amelie Chi Zhou, and Xiaowen Chu. BurstGPT: A real-world workload dataset to optimize llm serving systems. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25)*, Toronto, ON, Canada, 2025. ACM. doi: 10.1145/3711896.3737413. URL https://doi.org/10.1145/3711896.3737413.

[82] Grant Wilkins, Srinivasan Keshav, and Richard Mortier. Hybrid heterogeneous clusters can lower the energy consumption of llm inference workloads. In *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, pages 506–513, 2024.

[83] Zuan Xie, Yang Xu, Hongli Xu, Yunming Liao, and Zhiwei Yao. A novel hat-shaped device-cloud collaborative inference framework for large language models. In *arXiv preprint arXiv:2503.18989*, 2025. URL https://arxiv.org/abs/2503.18989.

[84] Jiaming Xu, Jiayi Pan, Yongkang Zhou, Siming Chen, Jinhao Li, Yaoxiu Lian, Junyi Wu, and Guohao Dai. Specee: Accelerating large language model inference with speculative early exiting. *arXiv preprint arXiv:2504.08850*, 2025. URL https://arxiv.org/abs/2504.08850.

[85] Zeyu Yang, Karel Adamek, and Wesley Armour. Part-time power measurements: nvidia-smi's lack of attention. *arXiv preprint arXiv:2312.02741*, 2023.

[86] Jie You, Jae-Won Chung, and Mosharaf Chowdhury. Zeus: Understanding and optimizing gpu energy consumption of dnn training. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023. URL https://www.usenix.org/conference/nsdi23/presentation/you.

[87] Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Dong Wang, Ilia Kulikov, Kyunghyun Cho, Yuandong Tian, Jason E Weston, and Xian Li. Natural-reasoning: Reasoning in the wild with 2.8m challenging questions, 2025. URL https://arxiv.org/abs/2502.13124.

[88] ZeroClaw Labs. Zeroclaw: Autonomous ai personal assistant infrastructure. https://github.com/zeroclaw-labs/zeroclaw, 2026.

[89] Yiqun Zhang et al. The avengers: A simple recipe for uniting smaller language models to challenge proprietary giants, 2025.

[90] Yiqun Zhang et al. Beyond gpt-5: Making llms cheaper and better via performance-efficiency optimized routing, 2025.

A Related Works

Below, we provide an extended treatment of related works.

LLM Routing A central challenge in local-cloud routing systems is determining which model should handle a given query so as to maximize efficiency. Prior work spans a broad design space, but much of it can be organized around two families of approaches: embedding-based routers [89, 72, 13] and generative/decoder-based routers [52]. Embedding-based methods rely on encoding queries (and sometimes models) into a vector space and then applying similarity search or lightweight classification. Early work largely adopted binary routing, where queries are directed between just two models. For example, RouteLLM [52] demonstrated that simple supervised classification can yield up to 85% cost reduction while maintaining GPT-4-level performance, but this setting was restricted to two-model scenarios. More recent systems generalize routing to multi-model settings: ensemble-style methods such as FrugalGPT [12], RouterDC [13], and Avengers Pro [89, 90] show that intelligently combining smaller models can approximate or even surpass larger frontier LMs. Decoder-based methods leverage a small language model to directly generate the routing decision. Causal LLM Routing, suggests that incorporating richer query-model interaction signals via generative modeling or cross-attention can yield more robust routing than static embeddings [13]. In this work, we are inspired by these novel approaches to routing, and evaluate their performance in the local-cloud routing setup.

LLM Routing Benchmarks Recent work has explored benchmarks for LLM query routing, primarily targeting cost–quality tradeoffs across multiple models. RouterBench [33] provides a comprehensive suite of curated academic tasks (405K samples) to evaluate routing policies along cost–quality Pareto frontiers. RouteLLM [52] introduces a preference-trained routing framework evaluated on academic benchmarks like MMLU and MT-Bench, with a focus on achieving quality under token cost constraints, though it remains limited to token-level metrics. RouterEval [34] emphasizes model selection accuracy at scale, compiling over 200M performance records across 8.5K models and 12 benchmarks to study generalization, yet lacks coverage of real-world queries. In contrast, our curated dataset targets routing under naturalistic conditions, leveraging 1M real user queries from WILDCHAT and NATURALREASONING. It uniquely supports the exploration of local-cloud routing tradeoffs beyond just cost and quality, to metrics such as latency, energy, memory, throughput, and more, generated on local accelerators and enterprise-grade accelerators. Moreover, in contrast to existing benchmarks, which provide stale performance records limited to models released prior to July 2024, our curated dataset evaluates several state-of-the-art models, including Qwen3 [62] and GPT-OSS [1], all released after May 2025. To support ongoing benchmarking, we release our efficiency profiling harness, a hardware-agnostic toolkit for generating fresh telemetry and evaluation records as new models become available.

Local–Cloud Inference Systems Beyond model selection, recent work explores collaborative inference protocols that split generation between local and cloud models. Minions [42] proposes a two-stage protocol where a small on-device LM handles lightweight processing and a frontier LM performs high-level reasoning, with an extended version introducing task decomposition and aggregation for improved quality. Such collaborative schemes offer large energy and cost savings but require careful protocol design to avoid performance loss. A parallel line of work centers on speculative decoding, where a small draft model generates candidate continuations that are verified or refined by a larger target LM [41, 84]. These approaches primarily target latency and throughput, particularly in constrained hardware settings, and typically assume that generation will ultimately invoke a large LM. Other hybrid protocols like SLED [38] and HAT [83] introduce edge-cloud model partitioning with intermediate state exchange to balance device limitations with quality needs. While these systems explore fine-grained collaboration at the token or layer level, our work investigates the limitations of a coarser-grained alternative: query-level routing across multiple small and large LMs, where we measure not only accuracy and cost, but also latency, memory, and energy across diverse hardware accelerators.

Efficient AI We are inspired by work on "Green AI" which proposes treating energy as a first-class metric alongside accuracy and cost, with calls for standardized reporting and tooling for reproducible accounting of power use and emissions during training and inference [67, 75, 60, 29, 5, 16, 58]. Most directly related, Fernandez et al. [23] jointly benchmark accuracy and energy on cloud GPUs (A100/H100) for fixed-format NLP tasks (sentiment classification, extractive QA, NLI). Our work is complementary but disjoint in scope: we study *local* accelerators (M4 Max, RTX 6000, MI300X, smartphone NPUs) and naturalistic real-world query distributions, decompose efficiency longitudinally into model versus hardware contributions across successive generations, and analyze hybrid local-cloud routing, none of which are addressed by their cloud-only, fixed-task setup. Complementary to our focus on local-cloud routing, cost and efficiency-driven model selection strategies such as FrugalML and FrugalGPT for API and model cascades, and CALM for token-wise early exit, dynamically allocate workloads to cheaper or smaller models while preserving quality [11, 12, 66]. On-device and edge studies demonstrate algorithm-hardware co-design for lower latency and energy consumption, exemplified by EdgeBERT's optimizations and PowerInfer's efficient LLM serving on commodity GPUs [76, 73]. Finally, hardware-aware benchmarking efforts such as "From Words to Watts" and MLPerf Power quantify inference energy across accelerators and standardize power measurement protocols [65, 77].

B Dataset and Profiling Harness

In this section, we provide additional details on our dataset curation for our study of hybrid local-cloud LM systems.

B.1 Dataset Curation

Here, we provide additional details on the Anthropic Economic Index [27] categories used as labels (see Table 3), the hardware platforms profiled, and the metrics recorded in our curated dataset.

Life, physical, and social science	Computer and mathematical
Architecture and engineering	Education instruction and library
Installation, maintenance, and repair	Business and financial operations
Legal services	Transportation and material moving
Arts, design, sports, entertainment, and media	Production services
Farming, fishing, and forestry	Healthcare support
Food preparation and serving related	Healthcare practitioners and technical
Community and social service	Sales and related
Office and administrative support	General management
Protective service	Building grounds cleaning and maintenance
Construction and extraction	Personal care and service

Table 3: **Anthropic Economic Index Categories [27].** This taxonomy categorizes occupations into 22 standardized economic domains, adapted from U.S. Bureau of Labor Statistics frameworks. It is designed to support AI impact analysis by aligning labor categories with distinct task structures.

Query Curation When sourcing queries from the WILDCHAT and NATURALREASONING datasets, we apply robust data cleaning and filtering to ensure the quality and consistency of the sampled queries. For NATURALREASONING, we filter out all queries that don't contain ground truth answers. For WILDCHAT, we eliminate non-English entries to maintain linguistic uniformity across the dataset. Queries that are malformed, nonsensical, or otherwise unintelligible (as determined by an LLM judge, i.e., GPT-4O-MINI) are discarded to prevent noise. Additionally, duplicate queries are removed to reduce redundancy and avoid overrepresentation of specific prompts. Finally, we filter out excessively long queries that exceed a 32,000-character limit.

Dataset Statistics Table 4 reveals significant differences in how the two datasets are distributed across domains. WILDCHAT is dominated by "Arts, design, sports, entertainment, and media" queries (47.1%), followed by "Computer and mathematical" (18.1%), while NATURALREASONING is primarily composed of "Life, physical, and social science" (36.0%) and "Computer and mathematical" (34.8%) queries. We use GPT-4O-MINI to bucket each query into its economic categorization using the prompt below.

```
You are a query categorizer. Your task is to categorize the following user query into one of the predefined categories based on the job/occupation domain it relates to most closely.

Query: "{query}"

Available Categories:
- Office and administrative support
- Transportation and material moving
- Sales and related
- Food preparation and serving related
- General management
- Business and financial operations
- Healthcare practitioners and technical
- Production services
- Education instruction and library
- Healthcare support
- Construction and extraction
- Installation, maintenance, and repair
- Computer and mathematical
- Building grounds cleaning and maintenance
- Protective service
- Personal care and service
- Architecture and engineering
- Community and social service
- Arts, design, sports, entertainment, and media
- Life, physical, and social science
- Legal services
- Farming, fishing, and forestry
- None

Instructions:
1. Read the query carefully
2. Determine which job/occupation category the query relates to most closely
3. If the query doesn't clearly relate to any specific occupation category, use "None"
4. Respond with ONLY the category name, exactly as listed above

Category:
```

Solvability rates vary dramatically by domain and dataset type, where a query's solvability is defined as its ability to be answered correctly by any of the available local LMs (e.g. Qwen models or GPT OSS). WILDCHAT queries show consistently high solvability across most domains (generally $> 94\%$), with particularly strong performance in creative and social domains. In contrast, NATURALREASONING exhibits more variable solvability, with technical domains like "Architecture and engineering" showing only 41.5% solvability compared to 99.4% for the same domain in WILDCHAT. This disparity reflects the complexity difference between open-ended chat queries and analytical reasoning tasks, supporting our findings that chat queries are more amenable to local model routing than reasoning-intensive queries.

Metrics We detail all the metrics collected via our profiling harness in Table 6. For our correctness evaluations on WILDCHAT, we use an LLM-as-a-judge approach to evaluate model generated answers against a ground truth answer from QWEN3-235B. For our correctness evaluations of NATURALREASONING, we use another LLM-judge prompt, but compare against ground truth answers provided in the original dataset. We provide both LLM-judge prompts

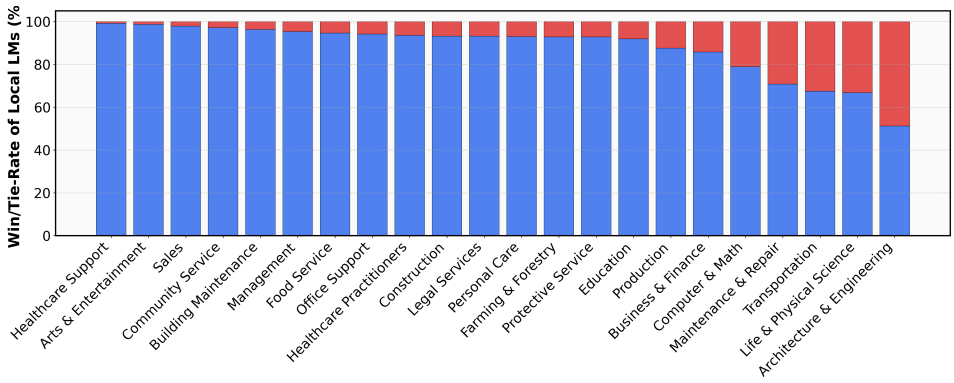


Figure 5: **Local Win/Tie-Rate vs. Cloud LMs by Domain**. Stacked bars show the fraction of single-turn chat and reasoning queries handled by local LMs ($< 20B$ active parameters; blue) versus those routed to frontier models in the cloud (red), computed per economic index domain [7]

Domain	WC Count	WC %	WC Solv %	NR Count	NR %	NR Solv %
Computer and mathematical	90,662	18.1	99.5	174,242	34.8	67.3
Arts, design, sports, entertainment, and media	235,658	47.1	98.7	2,648	0.5	52.9
Life, physical, and social science	28,079	5.6	98.8	180,065	36.0	60.5
None	49,014	9.8	97.3	79,752	16.0	65.6
Education instruction and library	23,196	4.6	97.2	13,864	2.8	80.4
Architecture and engineering	5,782	1.2	98.9	28,762	5.8	40.8
Business and financial operations	19,628	3.9	97.8	8,779	1.8	55.3
Healthcare practitioners and technical	8,905	1.8	98.1	1,851	0.4	66.3
Office and administrative support	6,959	1.4	91.6	25	0.0	48.0
Legal services	5,208	1.0	98.6	1,349	0.3	69.0
Community and social service	6,125	1.2	97.0	404	0.1	76.2
Transportation and material moving	1,890	0.4	95.1	3,914	0.8	53.0
Sales and related	4,689	0.9	97.8	218	0.0	67.4
Food preparation and serving related	3,308	0.7	98.3	507	0.1	61.9
General management	3,364	0.7	96.7	340	0.1	67.9
Installation, maintenance, and repair	954	0.2	97.1	2,037	0.4	56.2
Farming, fishing, and forestry	1,677	0.3	99.5	411	0.1	65.5
Protective service	1,315	0.3	97.9	237	0.0	56.5
Construction and extraction	991	0.2	97.2	147	0.0	60.5
Healthcare support	1,112	0.2	97.5	12	0.0	100.0
Production services	546	0.1	100.0	334	0.1	65.3
Personal care and service	648	0.1	92.9	19	0.0	0.0
Building grounds cleaning and maintenance	278	0.1	100.0	70	0.0	72.9
TOTAL	**500K**	**100.0**	**98.4**	**500K**	**100.0**	**63.0**

Table 4: **Dataset Domain Composition and LM Coverage ($\leq 20B$ Active Parameter Models)**. Comparison of domain distribution and model solvability rates across WILDCHAT (WC) and NATURALREASONING (NR) datasets. Solvability indicates the percentage of problems that can be solved correctly by at least one model with $\leq 20B$ active parameters.

below. The LLM used for each respective evaluation is GPT-4O. For SUPERGPQA and MMLU PRO, we simply compare the multiple choice answer selected in the response to the multiple choice answer of the reference response.

WILDCHAT LLM-judge Prompt

```
1
2 You are an impartial judge evaluating
      the quality of two AI-assistant replies to the same user prompt.
3
```

Category	WILDCHAT	MMLU PRO	SUPERGPQA	Average
Computer and mathematical	93.4%	90.6%	72.8%	85.6%
Life, physical, and social science	91.1%	84.7%	50.4%	75.4%
Sales and related	86.8%	74.2%	64.3%	75.1%
Business and financial operations	89.3%	82.9%	52.5%	74.9%
Production services	89.7%	85.7%	48.8%	74.7%
Office and administrative support	88.5%	83.3%	44.8%	72.2%
Healthcare practitioners and technical	88.9%	78.8%	48.0%	71.9%
Installation, maintenance, and repair	90.4%	80.0%	44.4%	71.6%
Architecture and engineering	90.0%	73.2%	51.6%	71.6%
Protective service	88.6%	75.0%	47.6%	70.4%
Education instruction and library	90.6%	77.4%	43.2%	70.4%
Farming, fishing, and forestry	87.2%	77.8%	45.9%	70.3%
None	86.5%	77.4%	42.4%	68.8%
General management	88.2%	76.3%	41.6%	68.7%
Transportation and material moving	91.1%	66.7%	44.9%	67.5%
Construction and extraction	91.0%	66.7%	41.4%	66.4%
Food preparation and serving related	86.1%	82.6%	30.3%	66.3%
Community and social service	87.4%	64.1%	45.5%	65.7%
Arts, design, sports, entertainment, and media	84.8%	75.5%	36.2%	65.5%
Building grounds cleaning and maintenance	90.1%	71.4%	30.8%	64.1%
Legal services	87.1%	61.5%	43.6%	64.1%
Healthcare support	86.5%	60.0%	38.9%	61.8%
Personal care and service	89.4%	50.0%	25.0%	54.8%

Table 5: **GPT-OSS-120B Performance across Datasets.** Performance metrics across WILDCHAT, NATURALREASONING, MMLU PRO, and SUPERGPQA benchmarks, organized by Anthropic Economic Index categories [27].

```
Step 1: Generate your own answer
Write the response *you* would give to the user. Keep it separate from later analysis.

Step 2: Decide the query type
Classify the user prompt as either
- **Subjective / open-ended** (creative writing, opinion, advice, brainstorming)
- **Objective / technical** (code, math, logical derivations with a single correct outcome)
If uncertain, default to "Subjective".

Step 3 - Score each assistant with the correct rubric

| Query type | Criteria |
|------------|----------|
| Subjective / open-ended | 1. Correctness / factual soundness 2. Helpfulness 3. Relevance 4. Conciseness 5. Creativity & novelty |
| Objective / technical | 1. Correctness only |

When using the multi-criteria rubric, note strengths and weaknesses for **each** dimension.
When using the single-criterion rubric, focus exclusively on factual / functional accuracy and ignore style or flair.

Step 4: Compare & justify
Explain which assistant is better and why, correcting any mistakes you find. Highlight missing but important details. **Be concise.**

Step 5:  Verdict
1. Assistant A is significantly better: [[A>>B]]
2. Assistant A is slightly better: [[A>B]]
3. Tie, Assistant A is equal: [[A=B]]
4. Assistant B is slightly better: [[B>A]]
5. Assistant B is significantly better: [[B>>A]]

Choose exactly one token from: `[[A>>B]]`, `[[A>B]]`, `[[A=B]]`, `[[B>A]]`, `[[B>>A]]`.
```

```

---

### Output format (strict)
Return **only** a JSON object that matches the provided schema:
```

NATURALREASONING LLM-judge Prompt

```
You are evaluating a response
    to a scientific/technical question against a reference answer.

Your task is to determine if the response
    is factually correct and complete compared to the reference.

Consider:
1. Scientific accuracy of facts and concepts
2. Mathematical correctness (if applicable)
3. Completeness of the answer
4. Technical precision

Question: {question}

Response: {response}

Reference Answer: {reference}

Return ONLY
   'True' if the response is correct and complete, 'False' otherwise.
```

Table 6 catalogs the per-query metrics collected by our profiling harness across compute, energy, latency, memory, and utilization dimensions; together these constitute the raw measurements from which all IPW, IPJ, PPW, and PPJ figures in the main paper are derived.

Hardware Backends Details regarding profiled hardware can be found in Table 7.

Data Generation Procedure We generate model outputs using consistent decoding settings across all tasks: `temperature` $= 0.6$, `top-p` $= 0.95$, `top-k` $= 20$, `min-p` $= 0.0$, and a 32768-token output limit. For NATURALREASONING, SUPERGPQA and GPQA queries, we enable deliberative prompting (`use thinking` $=$ `True`); for WILDCHAT, we disable it. For QWEN models, we apply a repetition penalty of 1.1 and length penalty of 1.0.

Telemetry Collection We collected telemetry by instrumenting host-level samplers that interface directly with vendor-supported system APIs on each platform. Data were obtained from NVML on NVIDIA-equipped hosts, from the `powermetrics` facility on macOS, and from ROCm SMI on AMD-equipped hosts. Each sampler queried the respective system interface to obtain GPU- and system-level measurements and produced synchronized records suitable for downstream quantitative analysis.

On NVIDIA systems, we interface directly with NVML and enumerate all visible GPUs. For each device, we query instantaneous power as reported by the driver, read cumulative energy from the on-device counter, obtain GPU temperature from the hardware sensor, and retrieve memory usage from the device's memory interface. Units are normalized (e.g., milliwatts and millijoules mapped to watts and joules; bytes to megabytes). In multi-GPU hosts, power and memory are summed across devices and temperature is averaged to yield a single aggregate view. Each record also includes host memory usage from OS counters, a nanosecond timestamp, and device identity and backend provenance.

On macOS, we execute `powermetrics` with elevated privileges and ingest its continuous `plist` stream. Each `plist` frame is parsed to extract the GPU power value exposed by the system (Apple Silicon: `processor_power.actual`; Intel: `processor.combined_power`), which is normalized to watts. Energy (joules) is obtained by numerically integrating the power signal over successive frames using the measured inter-frame wall-clock interval. In parallel,

Metric	Description
flops_per_request	FLOPs per query.
macs_per_request	MACs per query; proxy for compute.
per_query_joules	Energy per query (J).
total_joules	Total energy across queries.
per_token_ms	Latency per token (ms).
throughput_tokens_per_sec	Token output rate (toks/s).
time_to_first_token_seconds	Time to first token (s).
total_query_seconds	Total time per query (s).
cpu_mb.avg / max / median / min	CPU memory usage (MB).
gpu_mb.avg / max / median / min	GPU memory usage (MB).
initialization_duration_seconds	Model load time (s).
batch_size	Query batch size.
gpu_memory_utilization	GPU memory use (0–1).
max_model_len	Max token length allowed.
max_num_batched_tokens	Max batch token count.
max_output_tokens	Max output tokens.
num_workers	Number of threads.
temperature	Sampling temperature.
top_k	Top-k cutoff.
top_p	Top-p (nucleus) threshold.
warmup_steps	Warm-up steps.
per_query_watts.avg / max / median / min	GPU power draw per query (W).
total_watts.avg / max / median / min	Session GPU power draw (W).
cpu_count	CPU core count.
cpu_brand	CPU model.
host_name	Machine hostname.
os_name / os_version / kernel_version	OS and kernel info.
temperature.avg / max / median / min	Device temperature (°C).
input	Input tokens per query.
output	Output tokens per query.

Table 6: **Dataset Metrics**. Summary of compute, latency, memory, and energy profiling metrics.

Hardware	Memory	Bandwidth	Power
NVIDIA A100 (Ampere)	40 GB HBM2	1,555 GB/s	400 W TDP
NVIDIA H200 (Hopper)	141 GB HBM3e	4.8 TB/s	Up to 700 W TDP
NVIDIA B200 (Blackwell)	192 GB HBM3e	8 TB/s	1000 W TDP
NVIDIA GH200 (Grace Hopper)	144 GB HBM3e (+624 GB LPDDR5X)	4.8 TB/s (GPU)	1000 W TDP
NVIDIA Quadro RTX 6000 (Turing)	24 GB GDDR6	672 GB/s	295 W TDP
NVIDIA RTX 6000 Ada Generation	48 GB GDDR6	960 GB/s	300 W TDP
AMD Instinct MI300X (CDNA 3)	192 GB HBM3	5.3 TB/s (peak)	750 W TBP
Apple Mac Studio (M4 Max)	128 GB unified	546 GB/s	480 W (system PSU)†
Apple iPhone 16 Pro (A18 Pro)	8 GB LPDDR5X	60 GB/s	∼12 W (SoC peak)
SambaNova SN40L RDU	64 GB HBM2E	1.6 TB/s	500 W TDP

Table 7: **Accelerator Details**. Memory, bandwidth, and power specifications of evaluated accelerators and systems. †For the Apple Mac Studio (M4 Max), 480 W is the system power-supply continuous rating; the M4 Max SoC under sustained AI workloads draws substantially less. Per-query power measurements in our study are taken via `powermetrics processor_power.actual` (GPU subsystem only) so that comparisons against NVML/ROCm-SMI accelerator-only power on other platforms are like-for-like (see App. B.1).

system memory usage is sampled from OS counters. Every observation is timestamped and annotated with Apple device identity and an explicit `powermetrics` backend tag.

On AMD systems, we use ROCm SMI to query current GPU power (watts), read temperature from junction or edge sensors (°C), and obtain VRAM usage from the device memory interface (bytes to megabytes). Energy (joules) is computed by integrating the power signal over time using consecutive sampling intervals. System memory usage is read from OS counters. In multi-GPU machines, the primary device under observation is explicitly selected

Size Threshold (≤)	Cost Savings		Compute Savings		Energy Savings	
	Qwen + GPT-OSS	Qwen	Qwen + GPT-OSS	Qwen	Qwen + GPT-OSS	Qwen
4B	65.2%	65.2%	65.1%	65.1%	63.5%	63.5%
8B	80.8%	80.8%	83.1%	83.1%	79.6%	79.6%
14B	89.0%	89.0%	93.0%	93.0%	87.0%	87.0%
20B	90.5%	—	97.4%	—	89.4%	—
32B	91.3%	91.9%	97.4%	92.8%	90.4%	90.5%
120B	91.1%	—	97.7%	—	90.7%	—

Table 8: **Cost, Compute, and Energy Savings from Local-Cloud Routing on WildChat**: Savings across different resources while maintaining task accuracy of SOTA open-source cloud model (i.e. `Qwen3 235B-A22B`).

Size Threshold (≤)	Cost Savings		Compute Savings		Energy Savings	
	Qwen + GPT-OSS	Qwen	Qwen + GPT-OSS	Qwen	Qwen + GPT-OSS	Qwen
4B	52.9%	52.9%	54.5%	54.5%	46.3%	46.3%
8B	60.5%	60.5%	62.5%	62.5%	54.0%	54.0%
14B	68.7%	68.7%	70.1%	70.1%	62.5%	62.5%
20B	73.3%	—	72.2%	—	67.8%	—
32B	76.9%	75.9%	75.1%	75.1%	72.4%	71.6%
120B	86.7%	—	86.0%	86.0%	85.9%	—

Table 9: **Cost, Compute, and Energy Savings from Local-Cloud Routing on NaturalReasoning**: Savings across different resources while maintaining task accuracy of SOTA open-source cloud model (i.e. `Qwen3 235B-A22B`).

(GPU index 0 in our setup), and all records carry precise timestamps together with device identity and backend metadata.

To ensure measurement precision and account for variance in inference-time behavior, we execute each query 10 times and aggregate power measurements across runs. For each query, we compute the mean power draw (watts) and mean energy consumption (joules) per query by averaging across these 10 independent executions. This repeated sampling approach reduces measurement noise and provides robust estimates of per-query resource consumption that account for system-level variability in accelerator utilization, thermal conditions, and memory allocation patterns.

C Limitations and Broader Impacts (Extended)

Limitations. Our study is subject to several limitations. **(1) Measurement precision.** Our energy and power measurements rely on software-level telemetry (NVML, `powermetrics`, ROCm SMI), which can introduce inaccuracies of 10–15% [85] and may not capture milliwatt-level variations that hardware wattage meters would. While our methodology aligns with established practices and provides consistent relative comparisons, absolute energy values should be interpreted accordingly. **(2) Query coverage.** Our analysis focuses on single-turn chat and reasoning queries. Multi-turn conversations, agentic workflows, tool use, and long-context applications represent substantial portions of real-world LLM traffic; we report a multi-turn extension on GAIA and TerminalBenchV2 in App. E.12 that confirms the qualitative patterns generalize, but routing decisions and efficiency tradeoffs may still differ in those settings. **(3) Evaluation methodology.** Our correctness measurements rely on LLM-as-a-judge (QWEN3-235B) for open-ended chat queries, which inherits any biases or systematic errors of the judge model. **(4) Hardware coverage.** While we evaluate eight datacenter, workstation, and consumer-class accelerators (and additionally a smartphone-class accelerator in App. E.11), we do not cover the full diversity of available local hardware (e.g., other mobile NPUs, integrated GPUs, edge accelerators), and our findings on local efficiency may not extrapolate uniformly across these classes. **(5) Batch size = 1 cloud comparison.** Our main per-query comparisons (Tables 13–14) use batch size 1 to follow standard local-inference benchmarking practice [28]. This is conservative for cloud accelerators: App. E.9 shows bs=64 on B200 yields 11–20× higher IPJ. The routing

simulation (Section 4.3) uses bs=16 for the cloud baseline. **(6) Single-runtime design.** We standardize on vLLM to isolate model-hardware effects; App. E.10 shows absolute IPW shifts by 3–12% across vLLM/SGLang/llama.cpp but rankings are preserved (Kendall's $\tau \in [0.87, 0.93]$). System-level work such as Zeus [86] and uServe [61] explores complementary axes (scheduling, power management) that we do not vary. **(7) Ecosystem velocity.** Our findings reflect models and accelerators available as of October 2025; sustained increases in DRAM/HBM pricing could slow the pace at which larger models become locally deployable, and our open-source profiling harness is designed to support periodic re-evaluation.

Broader Impacts. Demonstrating that local inference can serve a substantial fraction of LLM queries has potential benefits for energy consumption, infrastructure cost, and access to AI capabilities, particularly in settings where cloud connectivity is unreliable, expensive, or undesirable for privacy reasons. Our findings are situated within a rapidly accelerating local-AI ecosystem. On the software side, a wave of open-source personal-AI agent stacks has emerged that treats on-device execution as a design constraint rather than a fallback. Projects span the full hardware spectrum: OpenClaw [74] and ZeroClaw [88] target workstation deployment; NanoBot [31], TinyClaw [25], and Hermes Agent [44] focus on lightweight runtimes with cross-session memory and skill systems; IronClaw [43] emphasizes privacy and security; PicoClaw [71] runs in under 10 MB of RAM on $10 RISC-V hardware; and MimiClaw [40] demonstrates a full agent on a $5 ESP32 microcontroller. Recent academic and industrial efforts include OpenJarvis [63], which integrates local model serving with agent and memory components. These projects illustrate growing interest in deployment patterns where local inference handles a substantial share of personal-AI queries. On the hardware side, purpose-built local AI hardware, e.g. NVIDIA DGX Spark (128 GB unified memory, 1 petaFLOP at FP4) and Dell Pro Max workstations with GB300 (748 GB, 20 petaFLOPS at FP4), is bringing datacenter-class AI to desktop devices for the first time. Hybrid local-cloud routing, as we show, can reduce inference energy by 60–80% at platform scale; the IPW metric and profiling harness we release are intended as a shared evaluation framework for tracking efficiency across this growing ecosystem as it matures. Our profiling harness lowers the barrier for systematic energy benchmarking, which we view as a public good given the rising aggregate energy footprint of AI workloads. We also note potential negative implications. First, increased local deployment of capable LMs may make certain forms of misuse (e.g., generating misinformation or harassment) harder to monitor and mitigate compared to cloud-served inference, where providers can apply usage policies. Second, local inference redistributes energy consumption to consumer power grids rather than eliminating it, and results on intelligence per watt could be misused to justify deployment patterns (e.g., always-on local agents) that, in aggregate, raise total energy consumption despite per-query efficiency gains: a Jevons-paradox concern we have not quantified. Third, our cost and energy comparisons are approximate and could be cited out of context to support specific procurement or policy decisions for which finer-grained, deployment-specific analysis would be more appropriate.

D Future Work

We outline three directions for extending this work.

D.1 Broadening Workload Coverage

Multi-turn and Agentic Workflows Our main study targets single-turn queries, and App. E.12 reports an extension on GAIA and TerminalBenchV2 confirming that local-cloud efficiency patterns generalize qualitatively. Future work should profile IPW across diverse agent stacks, longer trajectories, and workloads dominated by tool-call overhead. Routing in agentic settings must account for cumulative trajectory cost rather than per-turn cost, which may shift the local-cloud crossover relative to single-turn workloads.

Long-context Inference Our evaluation caps inputs at 32,000 characters, but production workloads increasingly involve 100K–1M token contexts where prefill dominates total energy and KV-cache pressure becomes the binding constraint. Characterizing IPW as a function of

context length would clarify how the prefill-decode energy split shifts with input size and how unified-memory and HBM architectures behave in such settings.

Multimodal Workloads Our study is text-only. Vision-language, audio, and video inference have different compute and memory profiles: image and video tokenization shift the prefill-decode balance, and modality-specific encoders introduce energy costs not captured by text-only profiling. Extending our harness to multimodal models and characterizing local-hardware viability on these workloads is an important direction.

D.2 Pushing the Local-AI Frontier

Closing the Reasoning Gap App. E.2 shows Level 5 reasoning queries remain 95% unsolved by current local LMs, and the hardest reasoning problems show markedly slower year-over-year progress than chat or moderate reasoning. Future small-model architectures, reasoning-focused post-training, and test-time compute strategies that fit within local power budgets are needed to push the locally-serviceable share beyond 88.7%. Whether long chain-of-thought reasoning can be made energy-efficient enough for local deployment is an open question.

Specialized On-device Architectures Today's local LMs are predominantly GPU-centric models repurposed for local hardware, and App. E.8 shows local accelerators trail cloud accelerators by 1.4–$7.4\times$ in IPW on identical workloads. Closing this gap may require NPU-first model designs, sparsity-aware architectures, and co-designed model-hardware stacks. The smartphone-class results in App. E.11 ($\sim 7\times$ higher IPW than workstation GPUs at ~ 12 W) suggest substantial headroom for purpose-built mobile deployments.

Quantization Frontiers Below FP4 App. E.5 shows FP16$\rightarrow$FP4 yields 3–$3.5\times$ energy reduction at ~ 2.5pp accuracy loss per step. Sub-FP4 regimes (INT2, ternary, binary) offer additional headroom but require both training recipes that preserve accuracy and dedicated hardware support that few accelerators currently provide. Characterizing the accuracy floor of extreme quantization and the hardware support needed to realize its theoretical gains are open directions.

D.3 Operationalizing Hybrid Inference

Serving-stack and System-level Optimizations We standardize on vLLM and bs$=1$ for local inference; App. E.9 and App. E.10 confirm batch size and framework (i.e., VLLM vs SGLANG) shift absolute IPW but preserve rankings. A full treatment of speculative decoding, paged attention, DVFS, power capping, and cross-application local batching could materially improve local IPW. Cross-application batching is particularly promising because single-user devices cannot batch across users but can aggregate concurrent queries across on-device applications.

Measuring IPW Under Hybrid Execution Our IPW metric is defined per (model, accelerator) pair and characterizes local or cloud inference in isolation. Hybrid execution patterns—speculative decoding with a local draft and cloud verifier, edge-cloud model partitioning [38, 83], and collaborative protocols like Minions [42]—split a single query across local and cloud infrastructure and require an extended IPW formulation that aggregates power and energy across heterogeneous accelerators.

E Local-Cloud Experiments

E.1 Key Takeaways for Practitioners

We consolidate our empirical findings into eight concrete decision rules, each grounded in a specific experiment. **(1) Architecture.** On memory-rich local devices, MoE delivers the best IPW: GPT-OSS-120B ($\leq$ 20B active) achieves the highest single-model coverage (71.4%, Figure 2) and best IPW (Table 2); capacity-to-compute decoupling more than compensates for storing all experts. **(2) Quantization.** FP16$\rightarrow$FP4 yields 3–$3.5\times$ energy reduction at

$\sim$2.5pp accuracy loss per step (App. E.5); a larger model at FP4 typically beats a smaller model at FP16, so scale model size first and quantize aggressively. **(3) Routing.** Invest in router accuracy up to $\sim 80\%$ (80% of oracle gains); beyond that, expand the local-model ensemble: the 17.3pp best-single vs. best-of-local gap (Figure 2) shows diversity matters more than additional routing accuracy. **(4) Domains.** Coverage exceeds 93% for creative fields but drops to 60% for Architecture & Engineering (Figure 5); model developers expanding local-AI viability should prioritize technical reasoning. **(5) Hardware bottleneck.** Cloud accelerators with HBM3e and dedicated tensor units achieve 1.4–$7.4\times$ higher efficiency than M4 Max's unified memory (Tables 13–14); the dominant local bottleneck is memory bandwidth and specialized compute, not raw FLOPs. **(6) Power envelope.** Smartphone-class NPUs occupy a distinct regime: the iPhone 16 Pro achieves $\sim 7\times$ higher IPW than workstation GPUs at ~ 12 W (App. E.11), motivating NPU-optimized mobile deployment for the lightest queries. **(7) Serving stack.** Cloud bs$=64$ on B200 yields 11–$20\times$ higher IPJ than bs$=1$ (App. E.9); local single-user deployment cannot batch but can aggregate concurrent on-device applications. **(8) Framework.** Absolute IPW shifts 3–12% across vLLM, SGLang, llama.cpp but rankings are preserved (Kendall's $\tau \in [0.87, 0.93]$, App. E.10); standardize for comparison, but benchmark on the target stack for absolute throughput.

E.2 How has local LM task coverage changed over different "difficulty" slices of the data

Using labels for query difficulty we quantify the rate of improvement of local LMs across task difficulty slices. We label each query by the minimum model size (in parameters) required to solve it when considering the SOTA LMs as of August 2025, categorizing queries into five difficulty levels: level 1 ($\leq$ 4B params), level 2 ($\leq$ 8B params), level 3 ($\leq$ 20B params), level 4 ($\leq$ 235B params), and level 5 (unsolvable).

For **chat tasks** (see Figure 6), we observe near-universal performance gains across all difficulty levels, with 2025 models achieving 98-99% success on levels 1-3 and 92.6% on level 4. Absolute improvements range from +55.4 percentage points (pp) for level 1 to +76.4 pp for level 3, indicating relatively uniform capability gains.

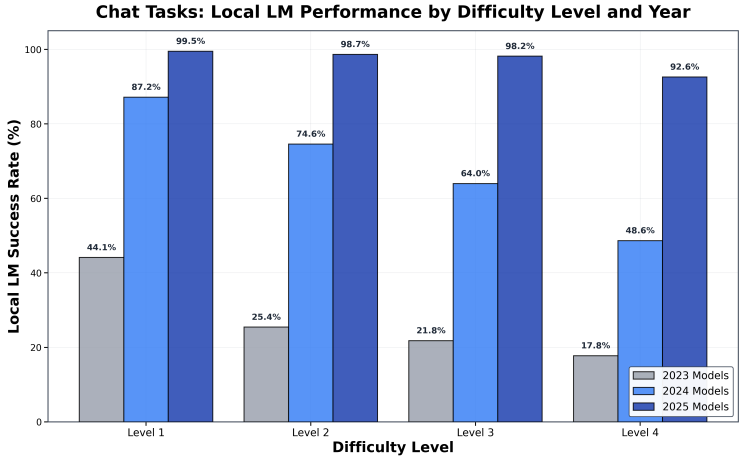


Figure 6: **Chat Task Performance by Difficulty Level and Year**. Model success rates across four difficulty levels and three model generations (2023, 2024, 2025). The data reveals dramatic progress across all difficulty levels, with 2023 models achieving 28.79% overall success rising to 98.12% by 2025. Notably, Levels 1-3 approach near-perfect performance (98-99%), while Level 4 shows the largest relative improvement (+210.4% per year) despite starting from the lowest baseline (17.77%).

For **reasoning tasks** (see Figure 7, the pattern differs substantially. While levels 1-3 show strong improvements (+24.0, +37.8, and +53.9 pp respectively), levels 4 and 5 exhibit markedly slower progress. Level 4 improves by only +23.8 pp (7.93% to 31.72%), and level 5 remains largely unsolved with just +1.5 pp improvement (3.27% to 4.72%). This suggests

that while local models have rapidly closed the gap on moderately difficult reasoning tasks, the hardest reasoning problems (those requiring either massive scale or capabilities beyond current architectures) remain a significant frontier. The presence of 134 level 5 problems (16.5% of the reasoning dataset) that remain 95% unsolved indicates substantial headroom for future model development in complex reasoning domains.

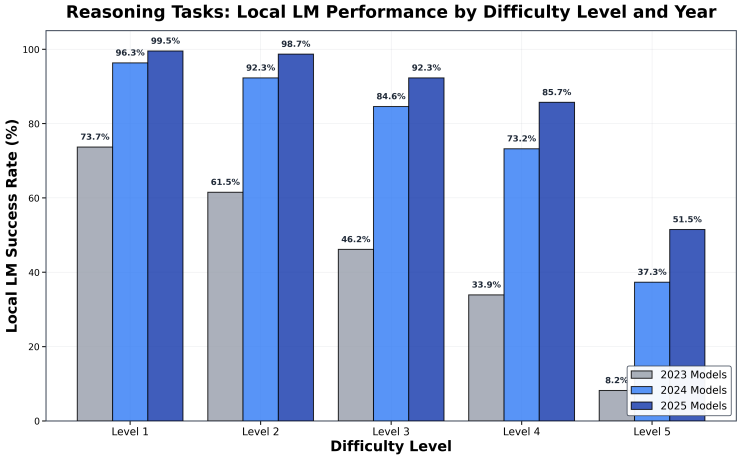


Figure 7: **Reasoning Task Performance by Difficulty Level and Year**. Model success rates on across five difficulty levels and three model generations. The benchmark shows a three-tier saturation pattern: near-complete (98-99% on Levels 1-2), approaching saturation (85-92% on Levels 3-4), and wide-open frontier (51% on Level 5).

E.3 How much has intelligence efficiency changed over time?

Figure 8 presents results across single-turn chat and reasoning queries, evaluating intelligence efficiency across model-hardware pairs from April 2024 through August 2025. We measure both perplexity (left panel) and accuracy (right panel) normalized by energy consumption in joules per query, tracking nine distinct model families (LLAMA, PHI, GEMMA, MISTRAL, FALCON, DEEPSEEK, QWEN, and GPT-OSS) deployed on various GPU configurations including NVIDIA A100 (40GB/80GB PCIe/SXM), H100 (80GB SXM), H200 (141GB HBM3e), and L40S (48GB) accelerators. Energy measurements capture end-to-end inference costs. The temporal snapshots at April 2024, August 2024, and August 2025 enable direct comparison of efficiency trajectories, revealing how successive generations of models and hardware migrate from suboptimal regions (high energy, low performance) toward optimal regions (low energy, high performance) as indicated by the shaded quadrants in each panel.

E.4 What efficiency gains can effective query routing deliver?

We compute cost per query using pricing available on OpenRouterAI [56]. Table 10 lists the token pricing used.

Model	Input Cost (USD / 1M tokens)	Output Cost (USD / 1M tokens)
Qwen3-4B	0.000	0.000
Qwen3-8B	0.035	0.138
Qwen3-14B	0.060	0.124
Qwen3-32B	0.100	0.450
Qwen3-235B	0.220	0.880
GPT-OSS-20B	0.03	0.14
GPT-OSS-120B	0.15	0.60

Table 10: **Model pricing from OpenRouterAI.** Costs are in USD per 1M tokens for input and output, as of August 2025.

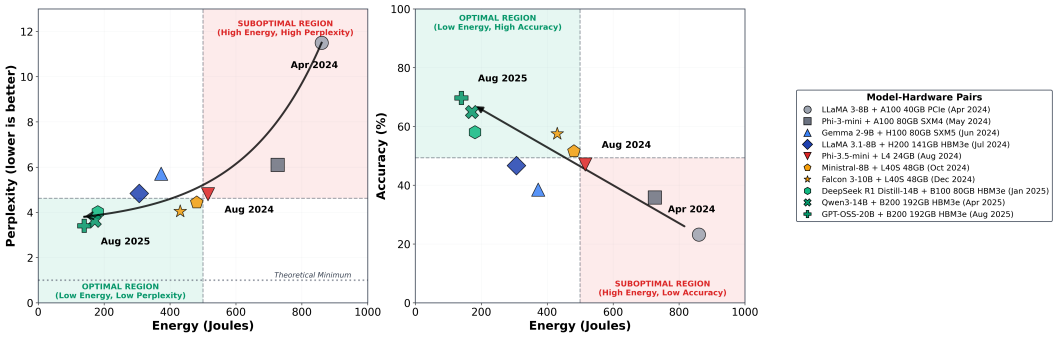


Figure 8: **Perplexity and Accuracy per Joule Trends across WILDCHAT and NATURALREASONING.**

E.5 How does model precision affect performance and efficiency?

Model quantization (reducing numerical precision from FP16 to FP8 or FP4) decreases memory requirements and energy consumption during inference while introducing approximation error that may degrade model accuracy. To quantify this tradeoff, we evaluate eight open-source models from the QWEN3 and GEMMA families across three precision levels: FP16 (full precision), FP8 (8-bit floating point), and FP4 (4-bit floating point). For FP8 and FP4 we use the vendor-published quantized checkpoints where available, falling back to native PyTorch FP8 / NVIDIA TransformerEngine FP4 conversions when no published checkpoint exists; energy measurements are taken on NVIDIA B200 (cloud) and replicated on APPLE M4 MAX (local) using the same telemetry harness described in App. B.1, with per-query results averaged across hardware platforms. For each model-precision pair, we measure accuracy on three reasoning-focused datasets: NATURALREASONING ($N = 10{,}000$), SuperGPQA ($N = 10{,}000$), and MMLU Pro ($N = 10{,}000$).

Figure 9 shows that quantization from FP16 to FP4 yields energy reductions of $3-3.5\times$ with accuracy degradation of approximately 2.5 percentage points per precision step across all models and datasets. For example, on SuperGPQA, QWEN3-14B achieves 54.5% (FP16), 52.0% (FP8), and 49.0% (FP4): a total degradation of 5.5 percentage points despite a $3.23\times$ reduction in energy consumption. Larger models maintain their relative performance advantage even at lower precision: QWEN3-14B at FP4 (49.0% accuracy) outperforms QWEN3-4B at FP16 (48.5% accuracy) on SuperGPQA, indicating that model scale matters more than precision for reasoning tasks. These results demonstrate that FP8 and FP4 quantization enable practical deployment of local models with predictable performance tradeoffs, allowing system designers to select precision levels based on application-specific requirements while capturing most of the energy savings identified in Section 4.3.

E.6 How do SOTA open-source LMs compare to the SOTA closed-source LMs on Chat and Reasoning Queries?

To evaluate the competitiveness of open-source models for local deployment, we compare the performance of state-of-the-art open-source models against leading closed-source models across single-turn chat and reasoning queries. We evaluate three closed-source frontier models (GPT-5-2025-08-07, GEMINI-2.5-PRO, and CLAUDE-SONNET-4-5) against eight open-source models ranging from QWEN3-8B to QWEN3-235B-A22B, measuring performance on NATURALREASONING, MMLU Pro, and SuperGPQA. Table 11 shows that the best open-source model (QWEN3-235B-A22B) achieves 71.8% average accuracy across benchmarks, trailing the best closed-source model (GPT-5-2025-08-07, 77.9%) by 6.1 percentage points.

Performance gaps vary substantially by task type, as shown in Table 12. On MMLU Pro and SuperGPQA, open-source models nearly match closed-source performance: QWEN3-235B-A22B achieves 82.3% versus 87.4% on MMLU Pro (5.1% gap) and 63.1% versus 66.5% on SuperGPQA (3.4% gap). However, on NATURALREASONING, the gap widens to 12.9% (70.0% versus 82.9%), indicating that closed-source models maintain a substantial advantage on

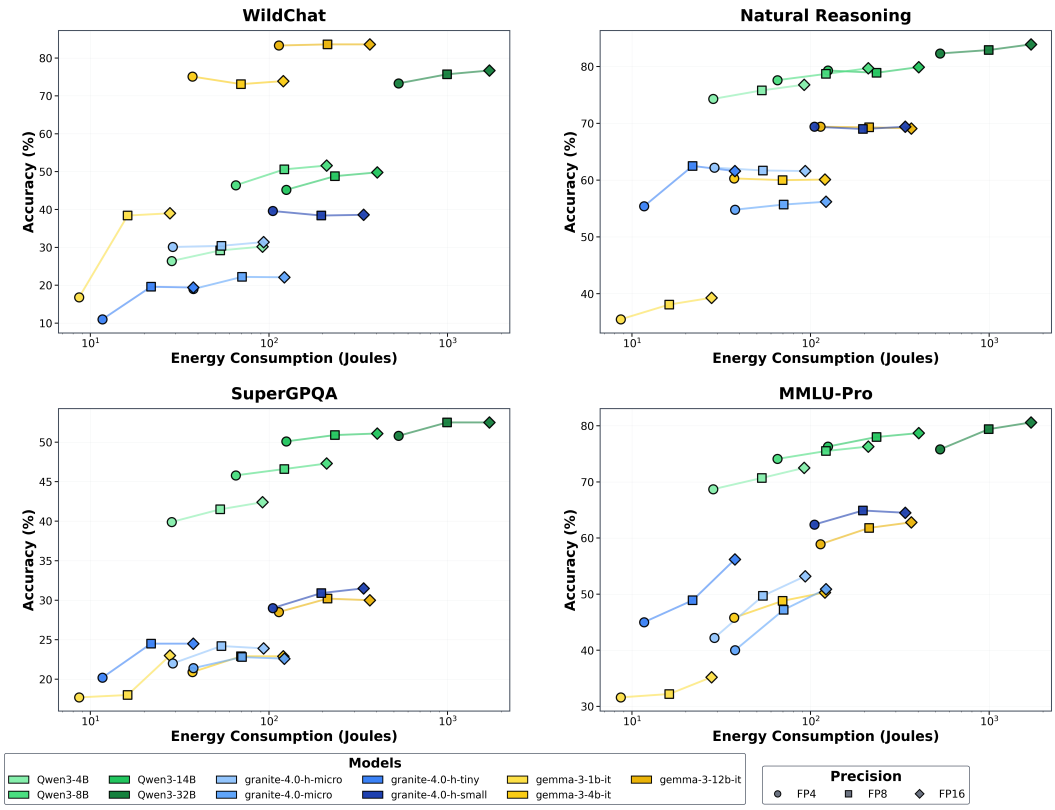


Figure 9: **Minimal Accuracy Degradation Shifting from FP16 to FP4 for Open-Source Local Models**: Evaluation across three reasoning datasets ($N = 10{,}000$ each) shows $2-3\%$ accuracy loss per precision step, demonstrating that $FP8/FP4$ quantization enables efficient deployment with acceptable performance tradeoffs.

naturalistic reasoning tasks. Local models with deployment constraints ($\leq 20B$ active parameters) face larger gaps: the best local model (QWEN3-14B) trails closed-source models by 11.8–13.2% across benchmarks, with the smallest gap on MMLU Pro (11.8%) and the largest on NATURALREASONING (13.2%). These results demonstrate that while open-source models at frontier scale (235B parameters) approach closed-source performance on knowledge and reasoning benchmarks, practical local deployment using smaller models (14B parameters) requires accepting 11–13% accuracy degradation relative to closed-source alternatives.

Model	Type	WildChat	NaturalReasoning	MMLU Pro	SuperGPQA	Average
GPT-5-2025-08-07	Closed	81.9%	82.9%	86.5%	64.4%	78.9%
GEMINI-2.5-PRO	Closed	**89.5%**	77.9%	**87.4%**	**66.5%**	**80.3%**
CLAUDE-SONNET-4-5	Closed	88.1%	76.9%	86.4%	60.1%	77.9%
QWEN3-235B-A22B (Best OSS)	Open	N/A*	70.0%	82.3%	63.1%	71.8%
QWEN3-32B	Open	76.1%	69.7%	77.9%	56.5%	70.1%
GPT-OSS-120B	Open	89.2%	65.0%	78.3%	55.3%	72.0%
QWEN3-30B-A3B	Open	47.3%	64.3%	76.9%	57.4%	61.5%
QWEN3-14B	Open	48.9%	60.0%	75.6%	56.2%	60.2%
GPT-OSS-20B	Open	77.3%	67.3%	73.4%	48.9%	66.7%
QWEN3-8B	Open	50.2%	57.9%	73.3%	51.8%	58.3%

Table 11: **Model performance comparison across benchmarks.** Scores represent performance on WILDCHAT, NATURALREASONING, MMLU PRO, and SUPERGPQA benchmarks. *Qwen3-235B-A22B is used as the reference model for WILDCHAT evaluation.

Metric	WildChat	NaturalReasoning	MMLU Pro	SuperGPQA
Closed Best	89.5%	82.9%	87.4%	66.5%
Best Closed Model	GEMINI-2.5-PRO	GPT-5	GEMINI-2.5-PRO	GEMINI-2.5-PRO
Open Best	89.2%*	70.0%	82.3%	63.1%
Best Open Model	GPT-OSS-120B	QWEN3-235B-A22B	QWEN3-235B-A22B	QWEN3-235B-A22B
Gap	−0.3%	−12.9%	−5.1%	−3.4%
Local Best ($\leq 20B$ Active)	89.2%	67.3%	80.3%	50.5%
Best Local Model	GPT-OSS-120B	GPT-OSS-20B	GPT-OSS-120B	QWEN3-14B
Local Gap	−0.3%	−5.6%	−7.1%	−16.0%

Table 12: **Closed-Source vs. Open-Source Performance Gap by Task.** Comparison between closed-source, open-source (all sizes), and local ($\leq 20B$ active parameters) models. *Excludes Qwen3-235B-A22B (reference model for WildChat).

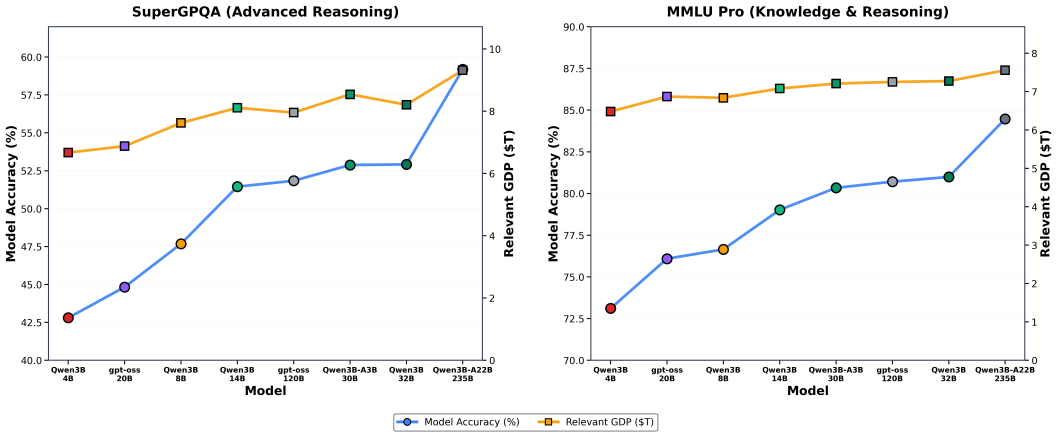


Figure 10: **Open-Source Local LMs Performance vs. U.S. GDP - SuperGPQA and MMLU Pro**: Model accuracy on SuperGPQA and MMLU Pro benchmarks plotted against relevant GDP in trillions of dollars. Both benchmarks show continued performance improvements as training compute scales across models from Qwen3B-4B to Qwen3B-A22B-235B. For our calculations, we compute the weighted sum of an LM's accuracy on each U.S. Labor category vs. the U.S. GDP associated with that category.

E.7 How does performance on chat and reasoning queries connect to U.S. GDP?

To assess the economic relevance of local model performance improvements, we compute GDP-weighted accuracy for each model by weighting its performance on each economic category by that sector's contribution to the 2024 U.S. GDP of $29.18 trillion [78]. This metric attempts to quantify what proportion of economic value is relevant and addressable by local LMs, given the local model performances across single-turn chat and reasoning queries. Figures 11 and 10 shows that model improvements translate directly into expanded GDP coverage: on SuperGPQA, QWEN3-235B-A22B achieves 59.2% accuracy covering $9.3T in relevant GDP (31.9% of total U.S. GDP), while on MMLU Pro, it reaches 84.5% accuracy covering $7.6T (26.0% of total U.S. GDP). The strong positive correlation between model accuracy and GDP coverage across both benchmarks demonstrates that scaling model capabilities systematically expands the set of economically valuable tasks that can be automated.

Task type substantially affects GDP coverage: chat tasks in WILDCHAT show the highest coverage with GPT-OSS-120B reaching 89.2% accuracy and covering $20.3T (69.6% of U.S. GDP), while reasoning tasks in NATURALREASONING show lower coverage with QWEN3-235B-A22 achieving 69.3% accuracy but only covering $6.8T (23.3% of U.S. GDP). This disparity reveals that current models excel at creative and conversational tasks that dominate economic activity, but struggle with technical reasoning tasks concentrated in specialized sectors like architecture, engineering, and physical sciences. The gap between chat coverage (69.6%) and reasoning coverage (23.3%) represents both a limitation of open-source local models and an economic opportunity: improving reasoning capabilities could unlock an additional $13.5T

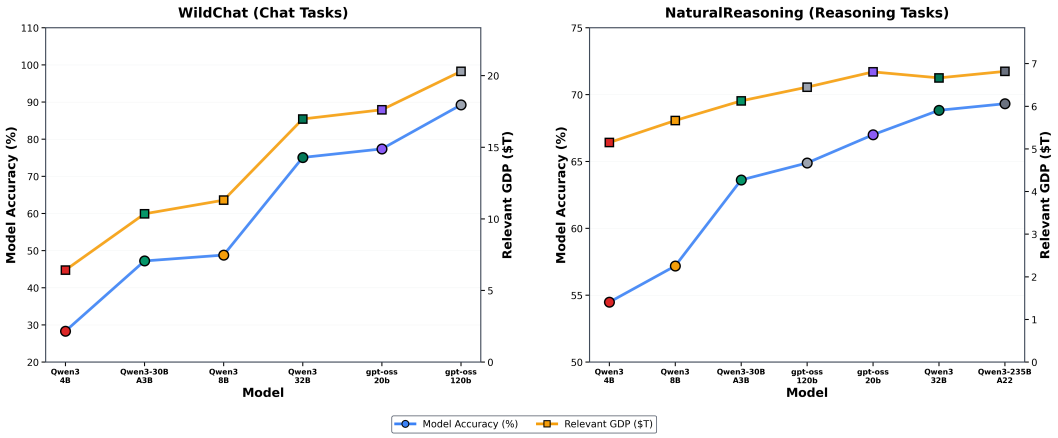


Figure 11: **Open-Source Local LMs Performance vs. U.S. GDP - WildChat and Natural Reasoning**: Model accuracy on WildChat and Natural Reasoning benchmarks plotted against relevant GDP in trillions of dollars. Both benchmarks show continued performance improvements as training compute scales across models from Qwen3B-4B to Qwen3B-A22B-235B. For our calculations, we compute the weighted sum of an LM's accuracy on each U.S. Labor category vs. the U.S. GDP associated with that category.

in GDP-relevant tasks, suggesting that advances in technical reasoning would have substantial economic impact beyond current model capabilities. We caution that this analysis treats benchmark accuracy as a direct proxy for the automation of economically valuable labor in each occupation category, which is an idealization: real-world deployment also depends on integration with existing tools, user trust, regulatory constraints, and the residual fraction of work in each occupation that is genuinely AI-amenable. The figures above are best read as upper-bound estimates of the addressable scope of local-LM-eligible tasks rather than predictions of realized economic impact, and our mapping from Handa et al. [27]'s 22 economic categories to U.S. Bureau of Economic Analysis GDP-by-industry data [78] is approximate.

E.8 How do local accelerators compare to cloud accelerators in terms of intelligence efficiency?

To understand the efficiency gap between local and cloud accelerators, we conduct a systematic comparison across identical model configurations. Tables 13 and 14 present intelligence per watt and intelligence per joule measurements for QWEN3 and GPT-OSS model families running on APPLE M4 MAX (local), NVIDIA B200 (cloud), and SAMBANOVA SN40L (cloud) accelerators.

Cloud accelerators achieve consistently higher power efficiency. Table 13 reveals that the NVIDIA B200 achieves $1.39\times$ to $1.40\times$ higher intelligence per watt than the APPLE M4 MAX across all QWEN3 model sizes evaluated (4B to 32B parameters). For QWEN3-32B, the B200 achieves $(2.75 \pm 0.14) \times 10^{-3}$ intelligence per watt compared to $(1.97 \pm 0.24) \times 10^{-3}$ for the M4 Max. The SAMBANOVA SN40L demonstrates even higher efficiency on larger models, achieving $(3.51 \pm 0.43) \times 10^{-3}$ intelligence per watt on QWEN3-32B ($1.78\times$ higher than the M4 Max and $1.28\times$ higher than the B200).

Energy efficiency gaps widen when accounting for latency. Table 14 extends the analysis to intelligence per joule, which captures end-to-end efficiency by incorporating both power consumption and generation latency. The efficiency advantages of cloud accelerators become more pronounced: the NVIDIA B200 achieves $1.6\times$ to $2.3\times$ higher intelligence per joule than the APPLE M4 MAX across QWEN3 and GPT-OSS model variants. For QWEN3-8B, the B200 achieves $(8.71 \pm 0.60) \times 10^{-5}$ intelligence per joule versus $(3.80 \pm 0.40) \times 10^{-5}$ for the M4 Max, a $2.29\times$ efficiency advantage. The SAMBANOVA SN40L demonstrates the largest efficiency gains, achieving $6.5\times$ to $7.4\times$ higher intelligence per joule than the M4 Max: $(3.12 \pm 0.38) \times 10^{-4}$ versus $(4.23 \pm 0.45) \times 10^{-5}$ for GPT-OSS-120B.

	Qwen3-4B	Qwen3-8B	Qwen3-14B	Qwen3-32B
Success Rate	$49.3 \pm 1.8\%$	$57.5 \pm 2.5\%$	$59.5 \pm 1.4\%$	$69.5 \pm 2.3\%$
Apple M4 Max				
Intelligence per Watt	$(1.40 \pm 0.38) \times 10^{-3}$	$(1.63 \pm 0.20) \times 10^{-3}$	$(1.69 \pm 0.31) \times 10^{-3}$	$(1.97 \pm 0.24) \times 10^{-3}$
NVIDIA B200				
Intelligence per Watt	$(1.95 \pm 0.14) \times 10^{-3}$	$(2.27 \pm 0.18) \times 10^{-3}$	$(2.35 \pm 0.09) \times 10^{-3}$	$(2.75 \pm 0.14) \times 10^{-3}$
SambaNova SN40L				
Intelligence per Watt	—	—	—	$(3.51 \pm 0.43) \times 10^{-3}$

Table 13: **Local accelerators demonstrate lower power efficiency than cloud accelerators**: When running the same QWEN3 models, the APPLE M4 MAX (local) attains $1.40\times$ lower intelligence per watt compared to the NVIDIA B200 (cloud) and SAMBANOVA SN40L (cloud), highlighting the efficiency advantage of purpose-built cloud accelerators over local accelerators. Values are mean $\pm$ 1-sigma sample standard deviation across 3–5 independent measurement runs per cell.

	Qwen3-8B	Qwen3-32B	GPT-OSS-20B	GPT-OSS-120B
Apple M4 Max				
Intelligence per Joule	$(3.80 \pm 0.40) \times 10^{-5}$	$(3.51 \pm 0.38) \times 10^{-5}$	$(4.38 \pm 0.31) \times 10^{-5}$	$(4.23 \pm 0.45) \times 10^{-5}$
NVIDIA B200				
Intelligence per Joule	$(8.71 \pm 0.60) \times 10^{-5}$	$(5.91 \pm 0.42) \times 10^{-5}$	$(7.34 \pm 0.51) \times 10^{-5}$	$(6.78 \pm 0.47) \times 10^{-5}$
SambaNova SN40L				
Intelligence per Joule	—	$(2.27 \pm 0.30) \times 10^{-4}$	—	$(3.12 \pm 0.38) \times 10^{-4}$

Table 14: **Cloud accelerators demonstrate superior energy efficiency across all models**: The NVIDIA B200 (cloud) achieves $1.6\times$ to $2.3\times$ higher intelligence per joule than the APPLE M4 MAX (local), while the SAMBANOVA SN40L (cloud) achieves $6.5\times$ to $7.4\times$ higher efficiency. These results highlight the substantial energy efficiency advantage of purpose-built cloud accelerators over local hardware across QWEN3 AND GPT-OSS model variants. Values are mean $\pm$ 1-sigma sample standard deviation across 3–5 independent measurement runs per cell.

Architectural differences explain the efficiency gap. The superior efficiency of cloud accelerators stems from purpose-built hardware optimizations for LLM inference: high-bandwidth memory (HBM3e with 4.8–8 TB/s bandwidth), dedicated tensor processing units, and optimized memory hierarchies that maximize throughput per watt. In contrast, local accelerators like the APPLE M4 MAX employ unified memory architectures (546 GB/s bandwidth) designed to balance diverse workloads (including CPU, GPU, and NPU tasks) under thermal and power constraints typical of consumer devices. The larger per-joule gaps (compared to per-watt gaps) reflect that cloud accelerators not only consume less power per unit of accuracy but also complete queries faster, compounding their energy advantage through reduced generation latency.

Local model capabilities are improving rapidly. Despite the efficiency disadvantage of local accelerators, Figure 12 demonstrates that local LM capabilities have improved dramatically from April 2024 to August 2025. On WILDCHAT, the win/tie rate of SOTA local models against QWEN3-235B increased from 28.0% in April 2024 to 78.2% in August 2025, a $2.8\times$ improvement in just 16 months. On NATURALREASONING, local models improved from 48.7% to 80.9% accuracy over the same period, representing a 66% relative improvement. These trends indicate that while local accelerators remain less efficient per query than cloud

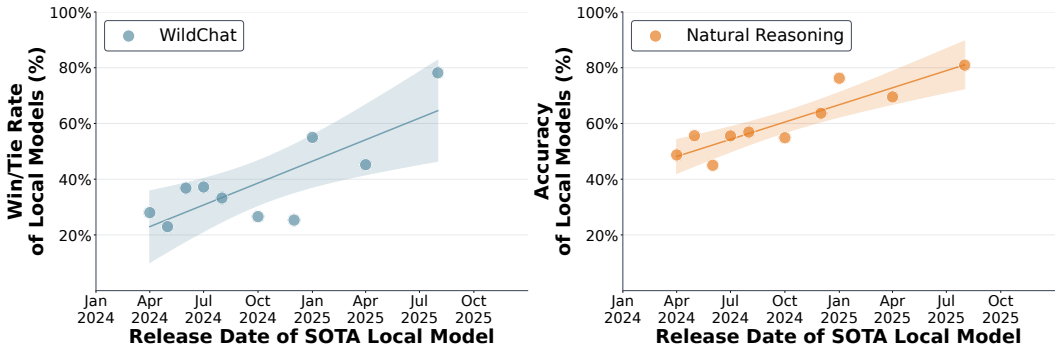


Figure 12: **Rapid Improvement of Local LMs across Chat and Reasoning Queries**: We evaluate the performance of SOTA local models released between April 2024 and August 2025 on WILDCHAT and NATURALREASONING. On WILDCHAT (left), local models show a win/tie rate of 78.2% against QWEN3-235B as of August 2025, compared to just 28.0% in April 2024: a 2.8× improvement in 16 months. On NATURALREASONING (right), local models achieve 80.9% accuracy by August 2025, up from 48.7% in April 2024: a 66% relative improvement.

infrastructure, the expanding capability of local models enables an increasing fraction of queries to be processed locally, avoiding cloud infrastructure entirely.

Local accelerator memory capacity is expanding rapidly. Figure 13 tracks the memory capacity of consumer accelerators from 2012 to 2025, revealing a $126.3\times$ improvement over this period. Local accelerators that offered 10–20 GB in 2020 now provide 128–512 GB through unified memory architectures like Apple Silicon. This memory expansion has been the primary enabler of local deployment for increasingly capable models: the transition from sub-20 GB to 200+ GB memory removes the key constraint that previously forced workloads to cloud infrastructure. Models with 8–20B active parameters that now handle the majority of inference queries can run efficiently on current-generation local hardware, with memory capacity continuing to scale at a pace that suggests even larger models will become locally deployable in coming years.

System-level benefits offset per-query efficiency disadvantages. While cloud accelerators demonstrate $1.4\times$ to $7.4\times$ higher intelligence efficiency per query, local deployment provides complementary system-level benefits that offset this disadvantage. Local inference avoids datacenter infrastructure costs, network latency, and API pricing, while enabling 88.7% of queries that local models can handle correctly to bypass cloud compute entirely. As demonstrated in Section 4.3, intelligent routing between local and cloud infrastructure can achieve 60–80% reductions in total energy, compute, and cost compared to cloud-only deployment, even when local accelerators are individually less efficient. These findings suggest that the path to efficient AI infrastructure lies not in local accelerators matching cloud efficiency, but in routing systems that leverage the complementary strengths of both paradigms: local processing for the majority of straightforward queries and cloud infrastructure for the minority requiring frontier model capabilities.

E.9 How does cloud batching affect intelligence efficiency?

Our main per-query comparisons (Tables 13–14) use batch size 1 to follow standard local-inference benchmarking practice [28] and to isolate intrinsic model-accelerator efficiency from system-level scheduling. Local devices serving a single user typically operate at bs=1 and cannot batch concurrent queries from different users, so for local accelerators bs=1 reflects realistic operating conditions. Cloud accelerators, however, can amortize idle GPU power across concurrent queries, and bs=1 is therefore conservative for cloud. To quantify the magnitude of this effect, we run an ablation on NVIDIA B200 sweeping batch size from 1 to 64 across three representative models (Table 15).

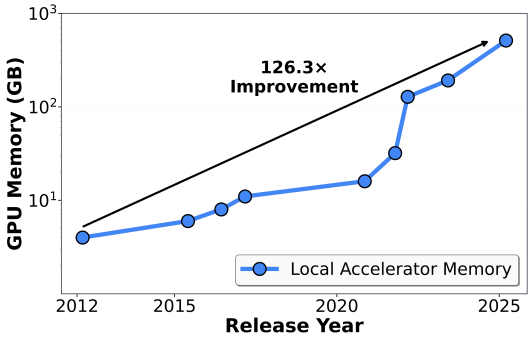


Figure 13: **Increasing GPU Memory of Consumer Accelerators**: Memory capacity (GB) for local accelerators. Over the past decade, local hardware has significantly closed the memory gap with cloud-grade accelerators, particularly since 2020, driven by advances in high bandwidth memory (HBM) components and unified memory architectures.

Model	bs=1 IPJ ($\times 10^{-5}$)	bs=64 IPJ ($\times 10^{-5}$)	IPJ Gain	Architecture
Qwen3-8B	8.92	104.93	11.8×	Dense
Qwen3-14B	7.22	80.61	11.2×	Dense
GPT-OSS-120B	6.72	132.21	19.7×	MoE ($\leq$ 20B active)

Table 15: **Cloud Batching Ablation on NVIDIA B200**: Per-query intelligence-per-joule at bs=1 versus bs=64. Per-query energy drops 12–20× at bs=64 (e.g., QWEN3-8B: 6,449 J $\rightarrow$ 548 J) while GPU power scales sublinearly (255 W at bs=1 to 883 W at bs=64). This confirms that bs=1 is conservative for cloud accelerators. Local accelerators (M4 Max, iPhone 16 Pro) serve a single user and cannot batch, so local IPW results are unchanged.

The routing simulation in Section 4.3 accounts for this by operating the cloud baseline at bs=16, so the reported 60–80% savings are computed against a batched cloud baseline rather than a bs=1 baseline. While absolute IPJ values shift substantially with batch size, relative rankings between model-hardware pairs remain stable, supporting the use of a fixed batch size for comparative analysis. Local single-user deployment cannot batch queries from a single user but can in principle aggregate concurrent queries across multiple on-device applications; we leave a systematic study of cross-application local batching to future work.

E.10 How sensitive are our findings to the choice of inference framework?

We standardize on VLLM across all platforms in the main study to enable clean attribution of efficiency gains to model and hardware factors rather than serving-stack choices. To test whether this choice biases our findings, we benchmark a representative subset of local models on APPLE M4 MAX across three popular inference frameworks: VLLM, SGLANG, and LLAMA.CPP (Table 16).

E.11 Are our findings consistent on smartphone-class accelerators?

To test whether the local-AI viability story extends to ultra-low-power devices, we evaluate the APPLE IPHONE 16 PRO (APPLE A18 PRO SoC with integrated NPU, 8 GB LPDDR5X unified memory, $\sim$12 W SoC peak power, 35 TOPS NPU) on a representative subset of our query distribution. Memory constraints (8 GB) require aggressive quantization (FP8 or FP4) and limit deployable models to $\leq$14B active parameters; within these constraints we evaluate models across three precision levels (FP16, FP8, FP4). Table 17 reports per-query measurements.

We find three patterns. First, smartphone-class accelerators are remarkably efficient on a per-watt basis: across precision levels, the iPhone 16 Pro achieves 11.2–13.3 $\times 10^{-3}$ accuracy per watt, approximately 7× higher IPW than workstation GPUs running the same model (Table 17), due to the $\sim$12 W SoC envelope versus 220–240 W for workstation GPUs. Second, this efficiency is conditional on the model fitting within the device's memory and thermal

Model	vLLM IPW ($\times 10^{-3}$)	SGLang IPW ($\times 10^{-3}$)	llama.cpp IPW ($\times 10^{-3}$)
Qwen3-4B	1.40	1.35	1.52
Qwen3-8B	1.63	1.58	1.71
Qwen3-14B	1.69	1.62	1.78
GPT-OSS-120B	4.18	4.05	4.31

Table 16: **Framework Sensitivity on Apple M4 Max**: Intelligence-per-watt across three popular inference frameworks for a representative subset of local models. Absolute IPW values shift by 3–12% across frameworks, but relative rankings between model-hardware pairs are preserved across the broader 20+-model, 8-accelerator evaluation: Kendall's $\tau \in [0.87, 0.93]$ and Spearman's $\rho \in [0.89, 0.94]$ across all framework pairs. This validates standardizing on vLLM for comparative analysis and confirms that our key findings are not an artifact of framework selection. Models that prioritize absolute throughput on a target stack should benchmark on that stack directly.

Model	Precision	Hardware	Acc. (%)	Power (W)	Lat. (s/q)	IPW ($\times 10^{-3}$)	IPJ ($\times 10^{-5}$)
Smartphone-class accelerator (Apple A18 Pro, iPhone 16 Pro)							
Qwen3-4B	FP16	A18 Pro	42.5 ± 2.0	12.0 ± 0.4	92.5 ± 9.4	11.8 ± 0.7	38.3 ± 3.9
Qwen3-4B	FP8	A18 Pro	40.5 ± 1.8	11.0 ± 0.3	55.2 ± 5.6	12.4 ± 0.7	66.7 ± 6.8
Qwen3-4B	FP4	A18 Pro	38.0 ± 1.6	9.5 ± 0.3	36.8 ± 3.7	13.3 ± 0.8	108.7 ± 11.0
Gemma3-4B	FP4	A18 Pro	32.0 ± 1.5	8.8 ± 0.3	33.5 ± 3.5	11.6 ± 0.7	108.5 ± 11.2
Granite-4.0-h-tiny	FP4	A18 Pro	28.5 ± 1.4	8.5 ± 0.3	29.2 ± 3.2	11.2 ± 0.7	114.7 ± 12.0
Workstation reference (Apple M4 Max, same model + precision for comparison)							
Qwen3-4B	FP4	M4 Max	44.5 ± 1.7	240 ± 18	3.5 ± 0.4	1.85 ± 0.13	53.0 ± 5.4
Workstation reference (NVIDIA RTX 6000 Ada, same model + precision)							
Qwen3-4B	FP4	RTX 6000 Ada	44.5 ± 1.7	220 ± 15	2.5 ± 0.3	2.02 ± 0.13	80.9 ± 8.1

Table 17: **Smartphone-Class Accelerator Efficiency**: Per-query measurements on Apple A18 Pro (iPhone 16 Pro, ∼12 W SoC peak, 60 GB/s memory bandwidth, 8 GB unified memory) across three precision levels, with workstation references for the same model at the strongest precision the iPhone supports. Values are mean ± 1-σ standard deviation across 3–5 independent measurement runs. Accuracy is averaged across the same evaluation subset used in App. E; latency is wall-clock per query at bs=1. The IPW ratio between A18 Pro and workstation GPUs on the same model anchors the ∼7× per-watt advantage of smartphone-class accelerators ($13.3/1.85 = 7.2\times$ vs. M4 Max; $13.3/2.02 = 6.6\times$ vs. RTX 6000 Ada); the IPJ comparison shows the gap widens slightly per joule against M4 Max (108.7 vs. 53.0×10^{-5}) but narrows against RTX 6000 Ada (108.7 vs. 80.9×10^{-5}), reflecting that the A18 Pro's 9–26× higher latency partially offsets its 20–25× lower power. Modest accuracy degradation on A18 Pro (∼6.5pp on Qwen3-4B FP4) reflects CoreML INT4 group quantization plus KV-cache pressure under 8 GB unified memory.

budget; FP4 quantization is essentially required and FP8 is the practical ceiling for sustained interactive use, with a modest accuracy cost (∼6.5 pp degradation on Qwen3-4B FP4 vs. workstation FP4). Third, the per-watt advantage does not translate uniformly to per-joule efficiency: the A18 Pro's 9–26× higher per-query latency (reflecting its 60 GB/s memory bandwidth versus 546 GB/s on M4 Max) partially offsets its lower power draw, so the IPJ gap narrows substantially or inverts depending on the workstation comparator.

These results suggest that ultra-low-power mobile NPUs are viable routing targets for the lightest queries, extending the local-inference paradigm from workstations and laptops to smartphones in active use. Concretely, a routing tier that dispatches very-low-difficulty queries to on-device NPUs could further reduce platform-scale energy beyond the 60–80% reductions reported in Section 4.3, though we leave a full characterization of mobile-tier routing to future work.

E.12 Do our findings generalize to multi-turn agentic workloads?

Our main study focuses on single-turn chat and reasoning queries because they constitute the largest share of real-world LLM traffic [10], but a substantial and growing fraction of usage involves multi-turn interactions, tool use, and agentic workflows. To test whether our local-versus-cloud efficiency patterns generalize, we evaluate two multi-turn benchmarks: GAIA (165 general-assistant queries with tool use, run with the OpenHands agent) and TerminalBenchV2 (TBv2; 80 terminal-task queries, run with the Terminus 2 agent) on local (APPLE M4 MAX, 128 GB unified memory) and cloud (8×NVIDIA H100 80 GB SXM5 node) hardware. Table 18 reports per-query measurements.

Benchmark	Model	Hardware	Acc. (%)	Power (W)	Lat. (s/q)	Energy (kJ/q)	IPW ($\times 10^{-3}$)	IPJ ($\times 10^{-5}$)
GAIA (165 multi-turn general-assistant queries with tool use)								
GAIA	MiniMax-M2.5	8×H100	16.4 ± 2.9	1558 ± 95	4.69 ± 0.41	7.31 ± 0.62	0.105 ± 0.020	2.24 ± 0.39
GAIA	Qwen3-235B	8×H100	5.5 ± 1.8	1595 ± 88	0.74 ± 0.09	1.18 ± 0.13	0.034 ± 0.012	4.66 ± 1.55
GAIA	Qwen3-30B	8×H100	8.6 ± 2.2	822 ± 64	1.22 ± 0.14	1.00 ± 0.11	0.105 ± 0.027	8.60 ± 2.20
GAIA	MiniMax-M2.5	M4 Max	14.2 ± 2.7	305 ± 24	51.6 ± 5.4	15.74 ± 1.69	0.466 ± 0.092	0.90 ± 0.18
GAIA	Qwen3-30B	M4 Max	6.4 ± 1.9	245 ± 19	13.4 ± 1.5	3.28 ± 0.39	0.261 ± 0.080	1.95 ± 0.61
TerminalBenchV2 (80 multi-turn terminal-task queries)								
TBv2	MiniMax-M2.5	8×H100	39.7 ± 4.4	1151 ± 78	2.00 ± 0.21	2.30 ± 0.24	0.345 ± 0.041	17.30 ± 1.94
TBv2	GPT-OSS-120B	8×H100	30.4 ± 3.9	1024 ± 71	3.01 ± 0.32	3.08 ± 0.33	0.297 ± 0.041	9.87 ± 1.31
TBv2	Kimi-K2.5	8×H100	29.1 ± 3.8	1251 ± 82	2.88 ± 0.29	3.60 ± 0.36	0.233 ± 0.033	8.08 ± 1.10
TBv2	MiniMax-M2.5	M4 Max	37.5 ± 4.2	295 ± 22	22.3 ± 2.4	6.58 ± 0.71	1.271 ± 0.158	5.70 ± 0.78
TBv2	GPT-OSS-120B	M4 Max	28.2 ± 3.7	268 ± 21	31.5 ± 3.3	8.44 ± 0.91	1.052 ± 0.165	3.34 ± 0.50

Table 18: **Multi-Turn Agentic Workloads Generalize Single-Turn Patterns**: Per-query measurements on GAIA (165 general-assistant queries with tool use, OpenHands agent) and TerminalBenchV2 (80 terminal-task queries, Terminus 2 agent) for self-hosted open-source models on local (Apple M4 Max, 128 GB unified memory) and cloud (8×NVIDIA H100 80 GB SXM5 node) hardware. Values are mean ± 1-σ standard deviation across 3 independent runs per cell; accuracy uncertainty uses the binomial approximation over benchmark size. M4 Max models use GGUF quantization via Unsloth to fit within unified memory; cloud models run at native precision via vLLM. The qualitative patterns from single-turn evaluation persist: cloud hardware achieves 2.4–$3.0\times$ higher per-joule efficiency on identical models (e.g., MiniMax-M2.5 on TBv2: 17.30 vs. 5.70×10^{-5}), while M4 Max achieves 3.7–$3.8\times$ higher per-watt efficiency due to its 3.5–$5.3\times$ lower power envelope, with only $\sim$2.2pp accuracy degradation across both benchmarks. The 12–15× higher M4 Max latency reflects the same memory-bandwidth bottleneck observed in single-turn evaluation, with multi-turn workloads amplifying absolute differences but preserving relative rankings.

The qualitative patterns from single-turn evaluation persist (Table 18): on identical models, cloud hardware achieves 2.4–$3.0\times$ higher per-joule efficiency than local (e.g., MiniMax-M2.5 on TBv2: 17.30 vs. 5.70×10^{-5} IPJ), consistent with Tables 13–14, while local hardware achieves 3.7–$3.8\times$ higher per-watt efficiency due to its 3.5–$5.3\times$ lower power envelope. Per-query accuracy on the strongest local model is within ~ 2.2 pp of the cloud configuration on both benchmarks (MiniMax-M2.5 on TBv2: 37.5 vs. 39.7%; on GAIA: 14.2 vs. 16.4%), suggesting that the routing-based savings reported in Section 4.3 carry over qualitatively to agentic workloads. Absolute savings rates may shift because multi-turn workloads have substantially longer effective context lengths and more tool-call overhead than single-turn chat (visible here as per-query energies in the kJ rather than J range). We caution that 245 multi-turn queries is a small evaluation and view this experiment as a sanity check rather than a definitive characterization; full multi-turn IPW characterization across more diverse agent stacks is an important direction for future work.