International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 22-28, May-2026

https://doi.org/10.47001/IRJIET/2026.105004

Does "Be Concise" Save Water? Measuring the Effect of Prompt Design on the Energy and Water Footprint of Open-Weight LLM Inference

[1]Isha Gautam Sontakke, [2]Unnati Nitin Shrivastava, [3]Sahiba Kamal Siddiqui, [4]Shrunkhal Moreshwar Supale, [5]Pushpa Tandekar

[1,2,3,4]Student, Computer Science and Engineering, Shri Sai College of Engineering and Technology, Bhadrawati, Chandrapur, India

[5]Professor, Computer Science and Engineering, Shri Sai College of Engineering and Technology, Bhadrawati, Chandrapur, India

Abstract **- Every text query sent to a modern large language model evaporates a small but measurable quantity of fresh water, directly through data-centre cooling and indirectly through power generation. At the scale of a popular consumer API, this aggregates to volumes equivalent to the household water consumption of a small town. We ask a narrow, practical question: by how much can a user reduce that footprint simply by changing how the prompt is written? We run a controlled experiment across four open-weight models served by OpenRouter, twenty standardised prompts spanning factual recall, reasoning, summarisation, and coding, and three prompting conditions, for a total of 266 controlled inferences. We separate direct (on-site cooling) from indirect (grid electricity) water, an accounting distinction the academic literature treats as essential [2, 3] but corporate sustainability disclosures routinely collapse [6]. On a fully sampled 20-billion-parameter model, prompts that ask for shorter answers reduce output tokens by 62-65% and water by 54-56% relative to an unconstrained baseline, with no measurable quality loss across all four task categories. Two cross-model findings sharpen the picture. On a 1.2-billion-parameter edge model, the same instruction reduces tokens but causes a quality cliff under one phrasing and not the other. On a 30-billion-parameter reasoning-tuned MoE model, an instruction to "answer in under 50 words" increases output tokens by 24%, the model interprets the instruction as a request for more careful reasoning rather than for shorter output. Prompt design is a real, immediately deployable user-side lever for AI sustainability; it is also an architecturally fragile one whose effect must be characterised per model class rather than assumed.**

Keywords: Large Language Model Inference, Prompt Engineering, Water Footprint, Energy Efficiency, Sustainable AI, Token Reduction, Open-Weight Models, OpenRouter, Water Usage Effectiveness, Green Computing.

I. INTRODUCTION

A short answer from a chatbot evaporates roughly half a drop of water [4]. A photorealistic image takes a small bottle of it [9]. A thirty-second AI video takes a bathtub. The asymmetry between how invisible these costs are and how rapidly they are scaling is the central environmental story of generative AI in 2026.

The total figures, when they enter public discussion at all, are large enough to demand attention. Global data centres consumed an estimated 560 billion litres of water in 2023 [5, 16]. Of this, roughly two thirds was indirect, water evaporated at fossil-fuel and nuclear power stations to generate the electricity the data centres then drew, and only one third was the direct on-site cooling water that hyperscalers report in their sustainability disclosures [5]. A single hyperscale facility in northern Virginia or Aragon, Spain, can draw the daily water of a small city; many such facilities are sited in regions already projected to face severe water stress by 2050 [12]. Against this backdrop, the framing of AI's environmental story almost exclusively in terms of carbon, with water as a footnote, is a measurable distortion of the underlying physics.

The water angle is also the angle on which the user agency is largest. End users have no influence over chip design, cooling architecture, grid composition, or model training compute. They have substantial influence over one variable: how many tokens the model generates in response to their query. Inference now accounts for the majority of an LLM's lifecycle energy consumption, with some bottom-up estimates placing the figure as high as 90% [1]. Inference scales linearly with adoption, and adoption is exploding. Per-query consumption multiplied by request volume is therefore the dominant term in the modern AI footprint, and per-query consumption scales linearly with output tokens. A user who learns to ask for shorter answers measurably shrinks the footprint of every query they send.

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 22-28, May-2026
https://doi.org/10.47001/IRJIET/2026.105004

This paper quantifies the size of that lever. Our contribution is threefold. First, we publish a fully reproducible measurement pipeline, code, prompt dataset, and raw API responses, that any student can re-run on a free OpenRouter account at zero cost. Second, we adopt the direct/indirect water decomposition explicitly in our reported numbers, rather than collapsing them into the single figure that consumer-facing materials typically present. Third, we report two empirical findings on prompt design that, to our knowledge, are not present in the prior literature: that the same length-constraining instruction can produce opposite token-count effects on different open-weight models, and that on summarisation tasks shorter-prompt instructions can mechanically improve answer fidelity rather than degrade it.

II. RELATED WORK

2.1 Estimating the water footprint of AI

The methodological foundation for AI water-footprint estimation is the work of Li et al. [4], who established that training GPT-3 evaporated approximately 700,000 litres of fresh water in Microsoft's US data centres and proposed a per-query estimation framework based on Power Usage Effectiveness and Water Usage Effectiveness coefficients. Subsequent work has split into a top-down school, which extrapolates from accelerator shipment data and assumed utilisation [10], and a bottom-up school, which benchmarks actual model inferences against measured node-level power [1][11]. The two schools disagree by roughly an order of magnitude on absolute volumes; the top-down approach yields the higher numbers because it assumes peak utilisation. We adopt the bottom-up approach here because it is more conservative and because the coefficients required are publicly documented.

A separate strand of work catalogues the geographic concentration of data-centre water demand, particularly the fraction of facilities sited in regions projected to face water stress by 2050 [12]. We do not contribute new evidence in this strand but reference it in our discussion of why the per-query numbers we report, although individually small, accumulate consequentially in specific watersheds.

2.2 Inference as the dominant lifecycle phase

Jegham et al. [1] introduced an infrastructure-aware framework for benchmarking the energy footprint of LLM inference, evaluating thirty state-of-the-art models. Two findings from that work are directly relevant to the design of our experiment. First, energy per token varies by more than a factor of seventy between optimised small models and reasoning models that perform extensive test-time computation. Second, even a single short query on a frontier model, when scaled to the daily traffic of a popular consumer API, yields aggregate water consumption equivalent to the annual drinking-water needs of a small city. The authors propose that future sustainability disclosures should report per-query footprints together with their underlying assumptions about hardware, batching, and parallelism. Our methodology adopts this principle.

2.3 Prompt-side efficiency

Most existing work on prompt-side efficiency targets engineering goals other than sustainability, latency reduction, cost minimisation, context-window pressure. Sprout [13] proposes routing simpler queries to smaller models and reserving expensive test-time compute for complex deductions. Recent EuroMLSys work [14] generalises this with energy-per-token routing curves. These are operator-side interventions; the user has no influence over them. We are not aware of prior empirical work that quantifies how much the user typing the prompt can change the energy or water footprint of their query through purely textual instructions to the model.

III. METHODOLOGY

3.1 Experimental design

We test four open-weight models, four task categories, three prompting conditions, with sample sizes designed to support robust per-cell comparisons on our primary subject and broader cross-model variance evidence on the others. The four models were chosen for public recognisability and to span the range of size buckets currently deployed: a 1.2B-parameter edge model (liquid/lfm-2.5-1.2b-instruct), two 20–30B-parameter models in different architectural families (openai/gpt-oss-20b; nvidia/nemotron-3-nano-30b-a3b, the latter a Mixture-of-Experts model with 3B active parameters per token), and a 1T-parameter sparse MoE model (inclusionai/ling-2.6-1t). All four are accessible through OpenRouter's free tier as of May 2026.

We treat gpt-oss-20b as our primary experimental subject (60 runs per condition; 15 per category-condition cell) and the other three models as supplementary cross-model variance evidence (22 to 37 runs each). Cell counts are reported alongside every figure and table so that readers can weigh findings appropriately.

The twenty prompts cover four categories with five prompts each: factual recall, reasoning, summarisation (a 5-sentence reference passage), and coding (a function specification with executable unit tests). The full prompt set, including ground-truth keywords and unit tests, is published with the code.

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 22-28, May-2026
https://doi.org/10.47001/IRJIET/2026.105004

The three prompting conditions are constructed by prepending a constraint to the base question: baseline (the question alone); terse (prefixed with "Answer in under 50 words. Be precise."); and minimal (prefixed with "Output only the answer. No explanation, no preamble, no caveats."). Run order is randomised with a fixed seed (42); temperature is fixed at zero throughout.

3.2 The estimation formula

We do not measure energy or water directly; this is impossible from outside a hyperscaler's data centre. We estimate from token counts using a transparent two-step formula:

$$energy\_per\_query\ (kWh) = (input\_tokens \times r \times E + output\_tokens \times E) / 3.6 \times 10^6$$

$$water\_per\_query\ (L) = energy\_per\_query \times (WUE\_direct + WUE\_indirect)$$

where E is the energy per output token in joules, r is the prefill-to-decode cost ratio (we use r = 0.20), WUE_direct is the on-site cooling water per kWh of IT load, and WUE_indirect is the off-site grid water per kWh of IT load delivered. The decomposition of total water into direct + indirect is the single most important methodological choice in this kind of estimation. Corporate sustainability disclosures consistently report the direct term alone, which is the smaller of the two by a ratio of roughly three to ten in the United States [5].

3.3 Energy and water coefficients

We bucket models into three size classes (small / medium / large) and assign each class low/mid/high estimates of joules per output token, drawn from published bottom-up benchmarks [1][11]. For water, we use WUE_direct = 0.18 L/kWh (top-tier hyperscaler average) and WUE_indirect = 0.60 L/kWh (US-blended grid). Together these give a total of 0.78 L/kWh, with an indirect-to-direct ratio of 3.33×. We report all numbers with low/mid/high bands so that readers can see the swing introduced by these coefficient choices.

3.4 Quality scoring

We score response quality using task-appropriate, deterministic metrics rather than subjective Likert ratings. Factual and reasoning prompts are scored by case-insensitive substring match against ground-truth keywords. Summarisation responses are scored by ROUGE-L F1 against a reference summary. Coding responses are scored by extracting the first Python code block from the model's reply, executing it inside a five-second timeout, and recording the fraction of unit tests that pass.

IV. RESULTS AND DISCUSSIONS

4.1 Primary finding (gpt-oss-20b)

Table 1 reports the headline numbers for the most heavily sampled model.

Table 1: Per-condition headline numbers for openai/gpt-oss-20b (N = 60 per cell)

Cond.	Output tokens	Wh	Water mL	Quality
baseline	182.1±162.5	0.171	0.134	0.774
terse	69.9±39.2	0.079	0.062	0.772
minimal	64.6±33.6	0.076	0.059	0.799

Both of the constrained conditions cut output tokens by roughly two-thirds against the baseline (terse: -61.6%, minimal: -64.6%), with corresponding reductions in estimated energy (-54%, -56%) and estimated water (-54%, -56%). Quality is preserved: the terse condition produces an essentially zero quality delta (Δ = -0.003), and the minimal condition produces a small but positive delta (Δ = +0.024). This combination, a 60% token reduction with no measurable quality cost, is the central finding of the paper. Under the unconstrained baseline, the model frequently produces extensive preamble, scaffolding, and post-hoc explanation that is not required by our quality metric and arguably not required by the user.

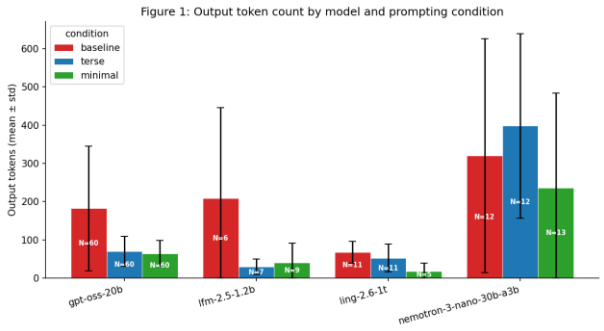

Figure 1: Output token count by model and prompting condition. Error bars show standard deviation; cell counts (N) shown inside each bar

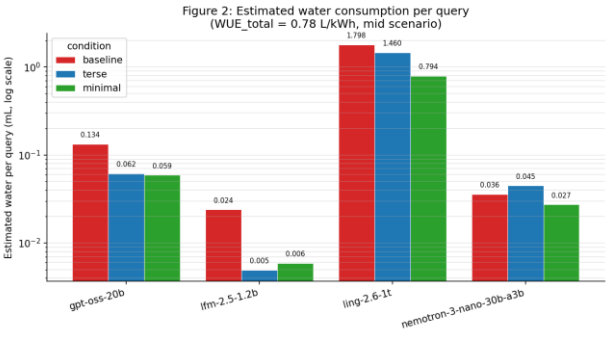

Figure 2: Estimated water consumption per query, by model and condition (log scale, mid scenario)

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 22-28, May-2026

https://doi.org/10.47001/IRJIET/2026.105004

4.2 Per-category breakdown

The headline finding is itself an average across categories, and the per-category breakdown is more informative for practitioners deciding when to use these techniques.

Table 2: Per-category breakdown for gpt-oss-20b (tokens / quality, N = 15 per cell)

Category	baseline	terse	minimal
factual	42.7 / .93	27.5 / .80	23.1 / .87
reasoning	212 / .80	93.7 / .80	74.0 / .80
summarisation	96.7 / .36	71.3 / .49	73.5 / .53
coding	377 / 1.00	87.0 / 1.00	87.6 / 1.00

Three observations. First, the largest absolute token saving comes from coding tasks, where baseline outputs run to nearly four hundred tokens of explanation around a function definition that is itself less than ninety tokens long. Unit-test pass rate is unaffected (1.00 across all three conditions). Second, reasoning tasks show a 65% token reduction with quality completely unchanged (0.80 across all three conditions); the model is happy to produce the correct answer without showing its work when asked not to. Third, and most surprisingly, summarisation quality improves under the constrained conditions (ROUGE-L: 0.36 → 0.49 → 0.53). We attribute this to a measurement artifact: under the baseline, the model returns the summary plus framing ("Here's a summary..."), which dilutes the ROUGE-L overlap with our reference; under the constrained conditions it returns just the summary itself.

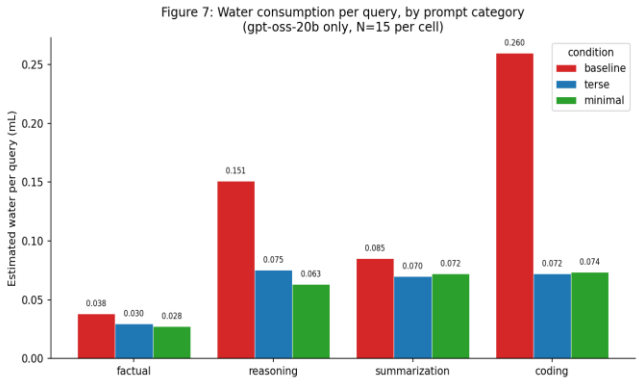

Figure 3: Water consumption per query by prompt category, gpt-oss-20b only (N = 15 per cell)

4.3 Cross-model variance

The other three models in our sample tell less clean stories. Table 3 reports their reduction percentages versus their respective baselines.

Table 3: Token, water, and quality changes versus each model's own baseline

Model	Condition	N	Token %	Water %	Quality Δ
lfm-1.2b	terse	7	-85.5%	-79.5%	-0.279
lfm-1.2b	min.	9	-80.6%	-75.4%	-0.018
nemotron -30b	terse	12	+24.4%	+25.2%	+0.148
nemotron -30b	min.	13	-26.1%	-23.7%	+0.073
ling-1t	terse	11	-22.6%	-18.8%	±0.000
ling-1t	min.	5	-72.6%	-55.8%	-0.076

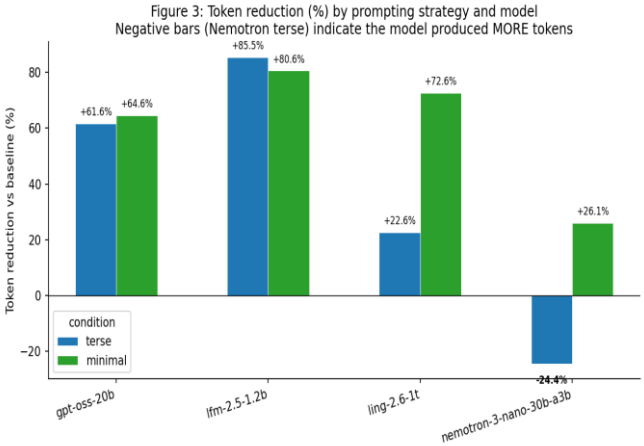

Figure 4: Token reduction (%) by prompting strategy and model. Negative bars indicate the model produced more tokens

4.4 The Nemotron exception

nvidia/nemotron-3-nano-30b-a3b is the only model in our sample where the terse condition produced more output tokens than the baseline (+24.4%), and the only model where that condition simultaneously improved quality (Δ = +0.148). The minimal condition does reduce tokens for this model (-26.1%) but by far less than the 60-80% range we see elsewhere. Nemotron 3 Nano is positioned as a reasoning-tuned model with configurable reasoning depth. The string "Be precise" in the terse condition appears to be interpreted not as an instruction to produce short output, but as a request for higher-quality reasoning, which manifests as longer chains of thought. The implication is that prompt-efficiency tactics cannot be assumed to generalise across model architectures: a prompt that saves 60% of tokens on a vanilla instruction-tuned model may save little or nothing, or, as here, actively waste tokens, on a reasoning-tuned model.

4.5 The lfm-2.5 quality cliff

The 1.2-billion-parameter Liquid model exhibits the inverse pathology: enormous token reductions (-81 to -86%) but a sharp quality cliff under one of the two phrasings. The

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 22-28, May-2026

https://doi.org/10.47001/IRJIET/2026.105004

terse condition drops average quality by 0.279, while the minimal condition under the same model produces nearly the same token reduction with effectively no quality loss (Δ = -0.018). A plausible mechanism is that "Answer in under 50 words. Be precise." encodes a length constraint that small models struggle to comply with while still producing complete reasoning; "Output only the answer." is a structurally simpler instruction that small models can follow cleanly. The asymmetry has practical consequences for production deployments.

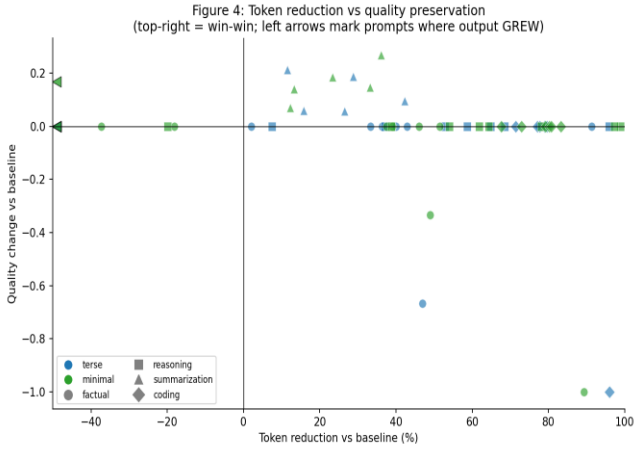

Figure 5: Token reduction vs. quality preservation. The top-right quadrant marks the win-win region

4.6 Direct vs indirect water

Across all 266 successful runs, the cumulative estimated water consumption is 56.6 mL. Of this, 13.1 mL (23%) is direct on-site cooling water and 43.6 mL (77%) is indirect grid water, a ratio of 3.33×. A sustainability disclosure that quotes only direct water captures less than a quarter of the actual hydrological cost in our model.

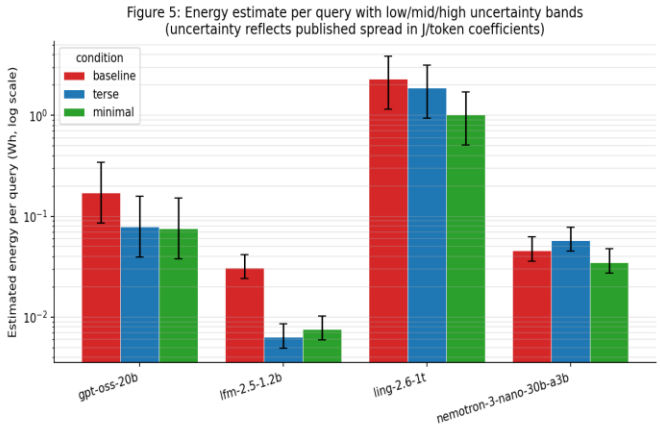

Figure 6: Energy estimate per query with low/mid/high uncertainty bands (log scale)

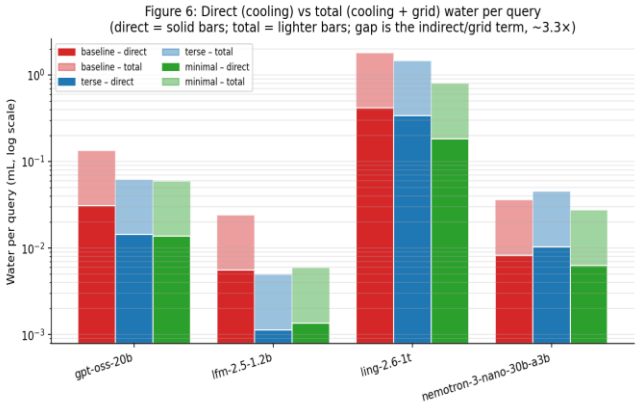

Figure 7: Direct (cooling) vs. total (cooling + grid) water per query. Solid bars = direct; lighter bars = total

4.7 What this looks like at scale

The per-query numbers above are small enough to feel inconsequential. Multiplied by realistic deployment volumes, they are not. Figure 8 extrapolates the per-query water cost for gpt-oss-20b across daily query volumes ranging from a small startup (10^4) to a single major commercial endpoint (10^9). At ChatGPT-scale traffic (10^9 queries/day), the unconstrained baseline corresponds to approximately 134 m³ of water per day. The minimal-prompt condition reduces this to 59 m³ per day, a saving of 74,000 litres daily, or 27 million litres annually, equivalent to the household water consumption of approximately 550 people in India (at 135 L per person per day). From a single textual instruction, applied to a single deployment, on a single mid-sized model. The per-query effect size is unspectacular; the aggregate is the story.

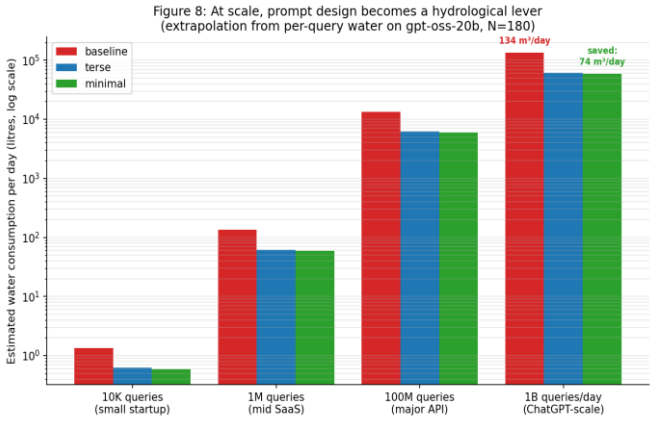

Figure 8: At scale, prompt design becomes a hydrological lever. Daily water consumption at four deployment scales, extrapolated from per-query water on gpt-oss-20b (N = 180)

4.8 Discussion

Two-thirds of output tokens, on a fully sampled mid-sized open-weight model, with no measurable quality cost, is an unusually clean result for a behavioural intervention in this

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 22-28, May-2026
https://doi.org/10.47001/IRJIET/2026.105004

field. By comparison, the headline efficiency improvement Google reported for its Gemini family between May 2024 and May 2025 was approximately 33× per prompt [15]. That figure was achieved through model-side optimisation requiring extensive infrastructure investment, was implemented over the course of a year, and is available to no one outside the company. Our 60% reduction is achieved through a thirteen-word string prepended to the user's prompt. The two are complements, not substitutes, but the size of the user-side lever has not yet been quantified in the literature.

It would be misleading to present prompt-side efficiency as a sufficient response to AI's environmental footprint. Per-query energy is small in absolute terms, and absolute volumes only become significant when aggregated across very large user bases, exactly the regime in which Jevons-paradox dynamics dominate. Per-query gains, no matter how large, do not by themselves bend the system-level curve; they must be paired with capacity discipline, which is a structural and political question rather than a technical one. We make no claim that prompt design solves the broader problem; we claim only that it is a real, quantifiable, user-actionable lever inside the broader problem.

4.9 Social impact

A growing fraction of new hyperscale data-centre construction is sited in regions already under hydrological stress. In the United States, this concentration is most visible in central Iowa, the Phoenix metropolitan area, and northern Virginia. In Europe, the cluster around Aragon in Spain has drawn community opposition specifically on water grounds. In Asia, India's largest data-centre build-out is concentrated in Maharashtra, with announced facilities in Pune, Mumbai, Navi Mumbai, and increasingly in central Maharashtra cities such as Chandrapur and Nagpur, regions whose summer water tables were already stressed before any data-centre demand was added [12]. At facility-scale water draws of 10^4 to 10^5 m^3 per day per hyperscale building, a 50% reduction in average per-query water consumption is the difference between one new data centre's worth of water demand and half of one.

The decision to send a prompt is made by an individual user in a high-income country, whose marginal cost is zero. The water consumed is drawn from an aquifer or river in a region that may be three continents away and may already be in seasonal deficit. The user has no visibility into this externality; the affected community has no agency over it. This is a textbook case of separated incentives, and one of the strongest arguments for in-product sustainability indicators: showing the user, in the moment, the resource cost of the query they just sent. Text inference is also the cheapest part of generative AI per output. Image generation has been documented at roughly four orders of magnitude more energy per output than text [9]. Short video generation has been documented at roughly four orders of magnitude above image. A complete social-impact account of generative AI in 2026 cannot stop at chatbots; the visible per-query numbers in the chatbot regime are an order of magnitude underestimate of the per-query numbers in the multimodal regime.

4.10 Limitations

We outline four categories of limitation. Routing opacity: OpenRouter routes different models to different upstream providers and physical regions; we have no way to inspect which data centre actually served any given query, so absolute water numbers should be read as estimates assuming our coefficient choices, while relative comparisons across conditions within a model are unaffected. Coefficient transfer: our energy-per-token coefficients are drawn from published benchmarks of similar models on similar hardware, not from telemetry on the actual servers OpenRouter dispatched to. Embodied water: we have not attempted to estimate Scope 3 embodied water from semiconductor fabrication and server manufacturing. Quality scoring: each response is scored by exactly one deterministic function rather than by an ensemble of references or human evaluation, so aggregate quality numbers are proxies for what a careful human reader would conclude. We have spot-checked the Nemotron observation by manually inspecting response texts to confirm the model is producing visibly longer reasoning chains under the terse condition; we are confident the +24% effect is not a measurement artifact, but additional samples and an analogous reasoning model from a different vendor would strengthen the claim.

V. CONCLUSION

Inference is now the dominant phase of an LLM's lifecycle resource consumption, and prompt design is a practical, measurable, user-controlled lever for reducing that consumption. On the most heavily sampled model in our experiment, simple instructions to constrain answer length cut output tokens by roughly two-thirds and water by roughly half, with no measurable quality cost, large enough at deployment scale to translate into 27 million litres of water saved annually from a single textual instruction applied to a single deployment. The effect is not uniform across model architectures: a 1.2-billion-parameter edge model collapses on quality under one of the two phrasings tested; a reasoning-tuned 30-billion-parameter MoE produces more output, not less, under an instruction explicitly asking for brevity. Sustainability claims that flatten across architectures will mislead. Sustainability disclosures that report only direct on-site cooling water will continue to capture less than a quarter

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 22-28, May-2026
https://doi.org/10.47001/IRJIET/2026.105004

of the actual hydrological cost. The methodological apparatus needed to measure this honestly is straightforward, can be assembled on a free-tier API, and does not require any privileged access to internal corporate telemetry; the only thing missing has been the empirical work itself, of which this paper is a small contribution.

ACKNOWLEDGEMENT

The authors thanks the OpenRouter platform for free-tier API access that made this work possible at zero cost, and the open-weight model providers (Liquid AI, OpenAI, NVIDIA, InclusionAI) whose released weights enabled reproducible evaluation. All code, prompts, and raw data are released at the repository linked above to enable replication.

REFERENCES

[1] N. Jegham et al., "How Hungry is AI? Benchmarking the Energy and Water Footprint of LLM Inference Across 30 Models," arXiv preprint, 2025.
[2] S. Ren et al., "Making AI Less Thirsty: A Methodological Critique of Top-Down Water Footprint Estimates," Communications of the ACM, 2024.
[3] A.de Vries, "The Hidden Resource Cost of Generative AI Infrastructure," Joule / ScienceDirect Patterns, 2025.
[4] P. Li, J. Yang, M. A. Islam, and S. Ren, "Making AI Less "Thirsty": Uncovering and Addressing the Secret Water Footprint of AI Models," arXiv:2304.03271, 2023 (updated 2024).
[5] Lawrence Berkeley National Laboratory, "2024 United States Data Center Energy Usage Report," prepared for the U.S. Department of Energy, 2024.
[6] Microsoft, "Environmental Sustainability Report FY2025"; Google, "Environmental Report 2024"; Meta, "Sustainability Report 2024"; Amazon, "Sustainability Report 2024" (aggregated and discussed in [5]).
[7] NVIDIA, architectural disclosures for the H100, B200, and Rubin-class accelerators, 2024–2025.
[8] A.H. Khalaj and S. K. Halgamuge, "A Review of Cooling Technologies for High-Density Data Centres," 2025.
[9] S. Luccioni, Y. Jernite, and E. Strubell, "Power Hungry Processing: Watts Driving the Cost of AI Deployment?" in Proc. ACM FAccT, 2024.
[10] A.de Vries, "The Growing Energy Footprint of Artificial Intelligence," Joule, vol. 7, no. 10, pp. 2191–2194, 2023.
[11] "TokenPowerBench: Node-Level Energy Profiling of Large Language Model Inference," 2025.
[12] Morgan Stanley Research, "AI Infrastructure and Water Stress: A Geospatial Analysis," 2025; MSCI ESG Research, "Data Center Asset-Level Climate Risk Assessment," 2025.
[13] "Sprout: Carbon-Aware Token Routing for LLM Inference," 2024.
[14] J. Stojkovic et al., "Energy-per-Token Routing in Production LLM Serving," in Proc. EuroMLSys, 2025.
[15] Google, "Methodology for Estimating Per-Prompt Energy and Water Consumption of Gemini Models," 2025.
[16] International Energy Agency, "Electricity 2024" and "World Energy Outlook 2025."

Citation of this Article:

Isha Gautam Sontakke, Unnati Nitin Shrivastava, Sahiba Kamal Siddiqui, Shrunkhal Moreshwar Supale, & Pushpa Tandekar. (2026). Does "Be Concise" Save Water? Measuring the Effect of Prompt Design on the Energy and Water Footprint of Open-Weight LLM Inference. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(5), 22-28. Article DOI https://doi.org/10.47001/IRJIET/2026.105004

****

