E-dLLM: Energy-Efficient Diffusion LLM Inference via Phase-Aware DVFS

Mona Moghadampanah, Xiangchen Li, Farhana Amin, Dimitrios S. Nikolopoulos

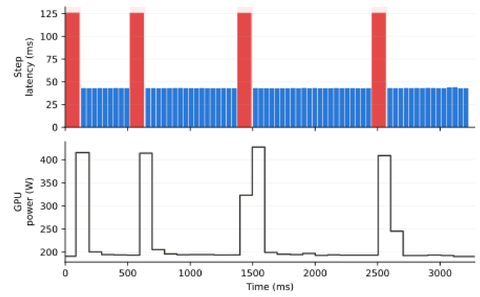
{monamp, lixiangchen, afarhana, dsn}@vt.edu , Computer Science Dept., Virginia Tech

Introduction

Diffusion LLMs (dLLMs) [1] have distinct decoding phases and configuration choices that strongly affect energy use. We characterize these behaviors and design **E-dLLM**, an energy-aware serving approach combining configuration selection with phase-aware DVFS. On LLaDA-8B, reactive DVFS **reduces energy by 17% with <1% latency overhead**, while fixed-frequency operation achieves **up to 30% energy savings with <10% latency overhead.**

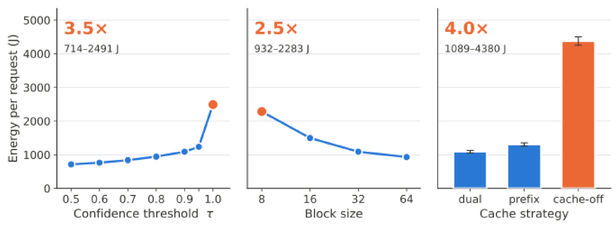
Motivation

- **Energy gap:** Energy-aware serving has been explored for autoregressive LLMs [4], while existing dLLM serving systems mainly optimize latency and throughput [2,3].



Per-step latency and GPU power within a DualCache decoding block.

- **Phase heterogeneity**: Cache-refresh and cache-reuse have very different compute and power behavior, creating an opportunity for phase-aware frequency control.



Energy per request under confidence-threshold, block-size, and cache-strategy sweeps.

- **Decoding choices:** Threshold, block size, steps, and cache strategy can change inference energy by up to **4.0×**.

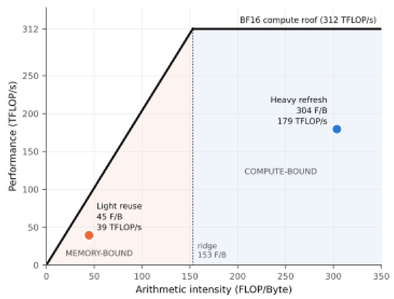
Research Questions

1. How do decoding configurations trade off accuracy, energy, and latency?
2. Can phase-aware DVFS reduce energy with minimal latency overhead?

Experimental Setup

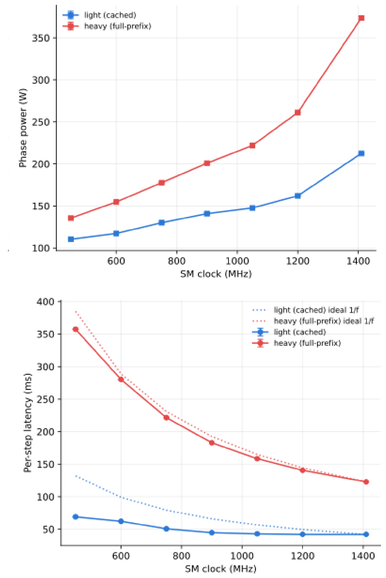
- **Model:** LLaDA-8B-Instruct, BF16, DualCache
- **GPU:** NVIDIA A100 80GB
- **Benchmark:** GSM8K, batch size 1
- **Energy measurement:** Zeus/NVML
- **DVFS:** SM clock range of 450–1410 MHz
- **Metrics:** accuracy, energy/request, latency/request, power
- **Repetitions**: 3 runs/config; energy CV <1%.

Characterization: Phase Behavior



Roofline analysis of cache-refresh and cache-reuse phases using Nsight profiling.

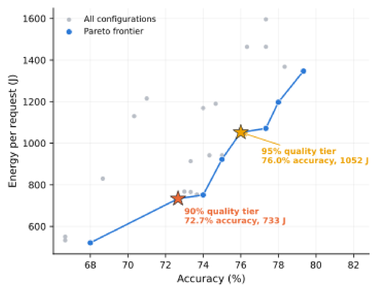
- **Roofline:** Refresh has **6.8× higher arithmetic intensity** and higher throughput , separating compute-bound refresh from memory-bound reuse.



Per-phase latency and power across fixed SM frequencies for cache refresh and reuse.

- **Phase latency & power:** Cache-refresh steps are ~3× slower and draw ~1.6× more power than cache-reuse steps, revealing strong within-block heterogeneity

Characterization: Configuration Trade-offs



Accuracy–energy Pareto frontier across valid decoding configurations on GSM8K.

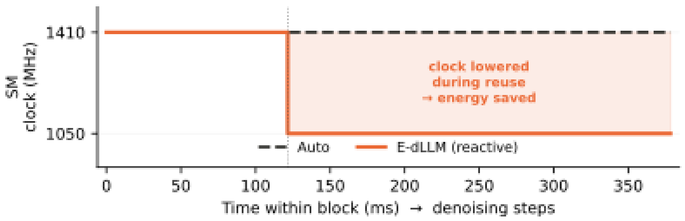
- **Configuration trade-offs:** Valid configurations span 66.7–79.3% accuracy and 521–1596 J/request, revealing a wide quality–energy design space.

E-dLLM Design



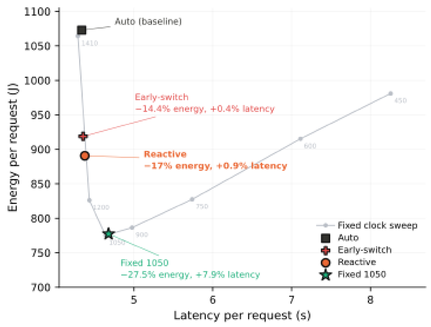
E-dLLM two-stage control flow for configuration and frequency selection.

- **Offline profiling:** Measure accuracy, latency, and energy for each decoding configuration on the target GPU.
- **Configuration selection:** From the offline profile, choose the lowest-energy configuration meeting the accuracy target and latency SLO.
- **Frequency selection:** Use offline frequency profiles to choose the lowest-energy DVFS policy meeting the latency SLO.
- **Phase-aware DVFS:** Keep refresh at a higher clock and lower the reuse clock based on its measured latency–energy trade-off.



Reactive phase-aware DVFS lowers the SM clock during cache reuse while keeping the refresh phase at higher frequency.

Results



Energy–latency trade-off across DVFS policies.

- At baseline, fixed 1050 MHz reduces energy by 27.5% but increases latency by 7.9%.
- Reactive phase-aware DVFS saves 17% energy with only 0.9% latency overhead.

Table:

- DVFS provides additional energy savings across all tested decode configurations.
- Reactive DVFS preserves near-baseline latency, while fixed 1050 MHz trades more latency for greater energy savings.

Config (τ / steps / block)	Accuracy	Auto Energy (J)	Auto Latency (s)	Fixed-1050 ΔE	Fixed-1050 ΔLat	Reactive ΔE	Reactive ΔLat
τ0.9 / s256 / b32(baseline)	**77.3%**	**1073**	**4.34**	**−27.5%**	**+7.9%**	**−17.0%**	**+0.9%**
τ0.6 / s256 / b64	66.7%	555	2.07	**−29.9%**	**+8.3%**	**−18.9%**	**+1.1%**
τ0.6 / s64 / b32	72.7%	738	2.44	**−30.7%**	**+12.3%**	**−13.7%**	**+0.9%**
τ0.9 / s128 / b64	73.3%	891	4.00	**−26.1%**	**+4.6%**	**−20.7%**	**+0.7%**
τ0.8 / s128 / b32	75.0%	947	3.52	**−30.0%**	**+8.8%**	**−17.0%**	**+0.7%**
τ0.9 / s128 / b32	76.0%	1014	4.28	**−26.3%**	**+7.2%**	**−16.1%**	**+0.4%**
τ0.8 / s128 / b16	79.3%	1273	4.34	**−29.0%**	**+13.4%**	**−11.2%**	**+0.4%**

DVFS energy and latency changes across decoding configurations.

17% lower energy
at only +0.9% latency (Reactive)

27.5% lower energy
at +7.9% latency (Fixed 1050 MHz)

Future Work

- **Full-system controller:** Integrate configuration selection and phase-aware DVFS.
- **Serving-scale evaluation:** Study larger batches, throughput, and p95 latency.
- **Generalization:** Evaluate more dLLMs, GPUs, and DVFS settings.

References

[1] Nie et al., *Large Language Diffusion Models*, 2025.
[2] Wu et al., *Fast-dLLM v2: Efficient Block-Diffusion LLM*, 2025.
[3] Chang et al., *DiLaServe: High SLO Attainment Serving for Diffusion Language Models*, 2026.
[4] Stojkovic et al., *DynamoLLM*, HPCA 2025.

Supported by NSF Grant No. 2315851, a Sony Faculty Innovation Award, an Amazon Research Award, and the Virginia Tech College of Engineering.