Call for Research Assistant: Efficient Artificial Intelligence

Aditya Desai

March 30, 2026

1 Project proposal

This is a broader proposal that would need more resources than those which are requested in the grant. This grant will be used to kick start this line of research

Long context processing is a critical capability in large language models because it enables them to operate effectively on real-world tasks that involve large, complex, and interdependent information. Many practical applications, such as analyzing legal contracts, understanding entire codebases, or performing deep research, require reasoning across documents that far exceed traditional context limits. With longer context windows, models can capture relationships between distant pieces of information, maintain coherence over extended interactions, and reduce the need for external retrieval systems that may miss relevant details. This improves both the accuracy and faithfulness of model outputs, as the model can ground its responses in a more complete view of the available data rather than relying on partial inputs. Additionally, long context processing aligns more closely with human workflows, where individuals naturally consider entire documents or histories when making decisions. It also supports long-form generation tasks such as writing detailed reports, drafting comprehensive articles, or generating extended narratives that remain coherent and consistent throughout. Overall, long context transforms language models from tools that process isolated fragments into systems capable of holistic understanding and deeper reasoning.

Another related (long context) problem is video generation, where models must produce temporally consistent sequences over many frames rather than isolated outputs. Unlike static image generation, video generation requires maintaining coherence in motion, object identity, lighting, and scene dynamics across time, effectively making it a long-context reasoning task. This capability is important for applications such as content creation, film and animation, simulation and gaming, virtual reality, autonomous systems training, and education, where it can be used to create interactive lessons, visual explanations, and immersive learning experiences. By leveraging long context, video generation models can ensure continuity and realism, enabling the synthesis of high-quality, structured visual narratives instead of disjointed clips.

Processing long contexts requires us to solve challenges on many fronts. I discuss them briefly below,

1. **Computational cost in *prefill*:** The prefill workload is when a sequence of tokens are processed for the first time. In auto regressive models, this populates the KV cache, or intermediate embeddings of tokens. In diffusion models used in diffusion LLMs and video generation, every diffusion step corresponds to a prefill workload. This workload is computationally heavy, primarily due to the $O(n^2)$ computational complexity of attention.

   A variety of approaches have been proposed to reduce the computational complexity of attention. Broadly, these include designing alternative architectures that replace standard attention mechanisms [1, 2]. However, to the best of my knowledge, even state of the art hybrid models still rely on at least a few full attention layers. Another line of work focuses on training models with sparse attention from the outset [3, 4, 5]. By restricting attention to a subset of tokens, these methods reduce the complexity to $O(nk)$, where each token attends to only k others. Complementary approaches apply sparsity at inference time to models originally trained with full attention [6, 7, 8]. Related ideas have also been explored in the context of video generation, where sparsity patterns tailored to spatiotemporal structure are used to accelerate prefill workloads [9, 10].

2. **Memory Movement in *decode*** The decode step is specific to autoregressive LLMs, where the model leverages the previously generated KV cache to predict the next token. This step has $O(n)$ computational complexity and is typically memory bound on modern hardware, as it requires transferring the entire KV cache from high bandwidth memory (or system RAM, if offloaded) to the compute units.

   To mitigate this bottleneck, several approaches have been proposed. Sparse attention methods restrict computation to a subset of tokens, thereby reducing memory traffic and improving decode latency [11, 12, 13, 14]. Another line of work focuses on compressing the KV cache itself, for example, through quantization, which lowers memory bandwidth requirements while preserving model performance [15].

3. **KV Cache compression** Closely related to the challenge of memory-bound decoding is the rapid growth in the memory footprint of the KV cache. As context length increases, the KV cache consumes a substantial fraction of available HBM, constraining the batch size during prompt processing and thereby reducing overall throughput. For example, in LLaMA-3.1-8B, a relatively small model, the KV cache footprint per token is approximately 0.25 MB. Consequently, long contexts can quickly exceed the memory capacity even of high-end GPUs.

   This has made KV cache compression a critical area of research. Most existing approaches rely on heuristic strategies to selectively retain or compress key-value pairs, balancing memory savings against impact on model quality [16, 17, 18, 19, 20].

4. **context window extension** No matter how long a context a model is trained on, real-world applications consistently demand longer horizons. For example, frontier coding platforms often need to summarize prior interactions to continue supporting extended user sessions. This underscores the importance of extending the effective context length beyond what the model was exposed to during training. However, preserving model quality under such extensions remains a significant challenge.

   Preliminary work in this area has explored techniques such as context repositioning [21] and sparsity-based methods [22]. While these approaches offer some promise, they often introduce substantial degradation in model quality, which cannot be overlooked. This highlights the need for more principled and robust solutions.

While each of the aforementioned problems has been studied to varying extents in the literature, existing approaches remain far from providing effective and reliable solutions. Below, I outline the key limitations of prior work and how our approach seeks to address them.

- **Over-reliance on heuristics:** A large fraction of existing methods are driven by heuristics or empirical observations. This lack of principled grounding not only leads to inconsistent gains / quality degradation across tasks and workloads, but also limits reliability and hinders real-world deployment. In contrast, my recent work has focused on developing *verified* approaches to efficiency [23, 24], where approximations are accompanied by explicit guarantees on the induced error. Building on this foundation, we will pursue verification-driven methodologies across all problem settings, ensuring both efficiency gains and robustness.

- **Narrow focus on conventional paradigms:** Prior research on efficiency has largely centered around sparsity and quantization, with some extensions to low-rank methods. While these approaches have achieved moderate success, they do not fully capture the design space of efficient computation. My previous work has demonstrated that fundamentally new paradigms can significantly shift the quality–efficiency trade-off frontier [25, 26, 27, 28]. Accordingly, we will not only refine existing techniques but also explore novel paradigms that are better aligned with the demands of these problems.

- **Fragmentation of evaluation and reproducibility:** The rapid growth of AI research has made it increasingly difficult to maintain a consistent and up-to-date understanding of the state of the art. This fragmentation leads to redundant efforts and the loss of insights that should propagate across works. As a result, progress on efficiency has often appeared "stalled" despite a large volume of published research.

To address this, we propose the development of a public leaderboard supported by centralized, modular codebases. This platform will implement state-of-the-art methods on standardized datasets under unified experimental protocols, enabling fair comparison, accelerating dissemination of results, and reducing the overhead required for rigorous evaluation.

- **Emergence of AI-driven research systems:** With the rise of AI-driven research (ADRS) [29] and automated discovery tools, the research process itself is undergoing a transformation. We plan to systematically explore ADRS methodologies across all problem domains considered in this proposal, leveraging them as copilots to accelerate hypothesis generation, experimentation, and analysis.

2 What is expected from the applicants

Following the skill rubric in Table 1(I have used examples from probability statistics / CUDA programming.), the candidate should be strong in atleast one of the following and fair at another

1. Probability and Statistics
2. Machine learning via pytorch / high level libraries
3. CUDA programming.

References

[1] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[2] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[3] Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

[4] Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23078–23097, 2025.

Strength Level	Examples
Poor	Can follow basic probability formulas (e.g., mean, variance) but struggles to apply them; limited understanding of linear algebra beyond definitions; writes simple sequential code; no experience with parallelism or GPU programming; cannot debug performance issues.
Fair	Understands core concepts like conditional probability, distributions, and basic linear algebra (matrices, eigenvalues); can implement standard ML algorithms; has basic exposure to parallel programming (e.g., multiprocessing, simple CUDA tutorials); limited ability to optimize or reason about efficiency.
Strong	Comfortable deriving and applying statistical results (e.g., bias-variance tradeoff, MSE analysis); solid grasp of linear algebra (SVD, projections) and optimization; writes efficient code; can implement and optimize parallel programs (multi-threading, vectorization); working knowledge of GPU programming (CUDA basics, memory hierarchy).
Very Strong	Can rigorously derive results and provide guarantees (e.g., concentration bounds, approximation error bounds); deep understanding of advanced math (randomized algorithms, numerical methods); designs new algorithms; expert in parallel and GPU programming (custom CUDA kernels, memory-bound vs compute-bound optimization, kernel fusion); able to reason end-to-end about system performance and scalability.

Table 1: Skill Matrix with Strength Levels and Examples

[5] Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chengxing Xie, Cunxiang Wang, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

[6] Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. *arXiv preprint arXiv:2502.20766*, 2025.

[7] Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, et al. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems*, 37:52481–52515, 2024.

[8] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[9] Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattention: Accurate and training-free sparse attention accelerating any model inference. *arXiv preprint arXiv:2502.18137*, 2025.

[10] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity, 2025.

[11] Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. *arXiv preprint arXiv:2406.10774*, 2024.

[12] Hailin Zhang, Xiaodong Ji, Yilin Chen, Fangcheng Fu, Xupeng Miao, Xiaonan Nie, Weipeng Chen, and Bin Cui. Pqcache: Product quantization-based kvcache for long context llm inference. *Proceedings of the ACM on Management of Data*, 3(3):1–30, 2025.

[13] Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Leon Bottou, Zhihao Jia, et al. Magicpig: Lsh sampling for efficient llm generation. *arXiv preprint arXiv:2410.16179*, 2024.

[14] Aditya Desai, Shuo Yang, Alejandro Cuadron, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Hashattention: Semantic sparsity for faster inference. In *Forty-second International Conference on Machine Learning*, 2025.

[15] Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun S Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024.

[16] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

[17] Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for llm kv cache compression at test time. *Advances in Neural Information Processing Systems*, 36, 2024.

[18] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*, 2024.

[19] Dmitry Akulov, Mohamed Sana, Antonio De Domenico, Tareq Si Salem, Nicola Piovesan, and Fadhel Ayed. Kvcompose: Efficient structured kv cache compression with composite tokens. *arXiv preprint arXiv:2509.05165*, 2025.

[20] Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

[21] Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. Training-free long-context scaling of large language models, 2024.

[22] Xiaoran Liu, Ruixiao Li, Qipeng Guo, Zhigeng Liu, Yuerong Song, Kai Lv, Hang Yan, Linlin Li, Qun Liu, and Xipeng Qiu. Reattention: Training-free infinite context with finite attention scope. *arXiv preprint arXiv:2407.15176*, 2024.

[23] Aditya Desai, Kumar Krishna Agrawal, Shuo Yang, Alejandro Cuadron, Luis Gaspar Schroeder, Matei Zaharia, Joseph E Gonzalez, and Ion Stoica. vattention: Verified sparse attention via sampling. In *The Fourteenth International Conference on Learning Representations*.

[24] Luis Gaspar Schroeder, Aditya Desai, Alejandro Cuadron, Kyle Chu, Shu Liu, Mark Zhao, Stephan Krusche, Alfons Kemper, Ion Stoica, Matei Zaharia, et al. vcache: Verified semantic prompt caching. *arXiv preprint arXiv:2502.03771*, 2025.

[25] Aditya Desai, Li Chou, and Anshumali Shrivastava. Random Offset Block Embedding (ROBE) for compressed embedding tables in deep learning recommendation systems. In D. Marculescu, Y. Chi, and C. Wu, editors, *Proceedings of Machine Learning and Systems*, volume 4, pages 762–778, 2022.

[26] Aditya Desai and Anshumali Shrivastava. The trade-offs of model size in large recommendation models: 100gb to 10mb criteo-tb dlrm model. *Advances in Neural Information Processing Systems*, 35:33961–33972, 2022.

[27] Aditya Desai, Keren Zhou, and Anshumali Shrivastava. Hardware-Aware Compression with Random Operation Access Specific Tile (ROAST) Hashing. In *International Conference on Machine Learning*, pages 7732–7749. PMLR, 2023.

[28] Aditya Desai and Anshumali Shrivastava. In defense of parameter sharing for model-compression. *arXiv preprint arXiv:2310.11611*, 2023.

[29] Audrey Cheng, Shu Liu, Melissa Pan, Zhifei Li, Bowen Wang, Alex Krentsel, Tian Xia, Mert Cemri, Jongseok Park, Shuo Yang, et al. Barbarians at the gate: How ai is upending systems research. *arXiv preprint arXiv:2510.06189*, 2025.