Transit-Layer Tensor Representation for Serverless Inference

Ben Mechels
University of Minnesota
Minneapolis, Minnesota, USA

Shu Shi (Advisor)
University of Chicago
Chicago, Illinois, USA

Wenyi Wang (Advisor)
University of Chicago
Chicago, Illinois, USA

Kyle Chard (Advisor)
University of Chicago
Chicago, Illinois, USA

Abstract

The scale of frontier machine learning systems has led to an increasing need to efficiently store and transfer model weights. Existing methods for serverless inference employ scheduling policies to reduce cold start and model swapping latency, but have largely reached the limitations of hardware. In this paper, we introduce Condense, an ultra-fast method of compressing tensor weights across applications. We implement Condense in a serverless runtime as a transit-layer representation and evaluate the latency reduction. Across Llama models from 1B to 13B parameters, Condense reduces stored weight sizes by 2.27–2.38× and improves cold-start loading time by 1.92–2.17× over ServerlessLLM while retaining competitive downstream accuracy. We demonstrate that compact tensor representations can be generated quickly to provide significant runtime reductions across machine learning applications.

Keywords

tensor compression, serverless inference, model loading, GPUs

1 Motivation

Modern machine learning systems move tensors between storage, host memory, accelerators, and networks. Yet tensor storage formats are usually chosen for numerical precision and kernel compatibility, not movement. Existing formats therefore spend scarce system bandwidth managing weight representations that still contain compressible structure. A compressed representation can reduce latency across applications over existing quantization techniques, critical for distributed training, federated learning, efficient model storage, serverless inference, and more.

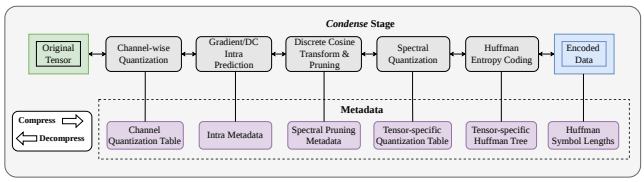


Figure 1: The Condense pipeline produces a densely compressed tensor package with ultra-high throughput.

2 Pipeline Design

Condense treats each transformer weight projection separately for compression. As Figure 1 shows, Condense does quantization plus specialized and throughput-focused compression techniques based on video codecs.

While stages like prediction and transforms are standard, we show that these methods can be tensor-specialized to increase both compression quality and throughput. Our pipeline design contributions are as follows:

- **Explicit prediction policies:** Each projection uses a globally selected prediction policy with explicit metadata grounding for block independent decoding.
- **Adaptive spectral pruning:** The least-magnitude coefficients are pruned independently, distributing error evenly within each transform window.
- **Tensor-specific quantization:** Outlier-aware quantization tables guard against resolution loss in spectral windows with high variance.
- **Low-overhead entropy coding:** A fast Symlen-style Huffman algorithm reduces coding and metadata overhead.
- **Metadata extraction emphasis:** Each stage captures specific structure as metadata, leading to large proportions of encoded data being metadata itself.

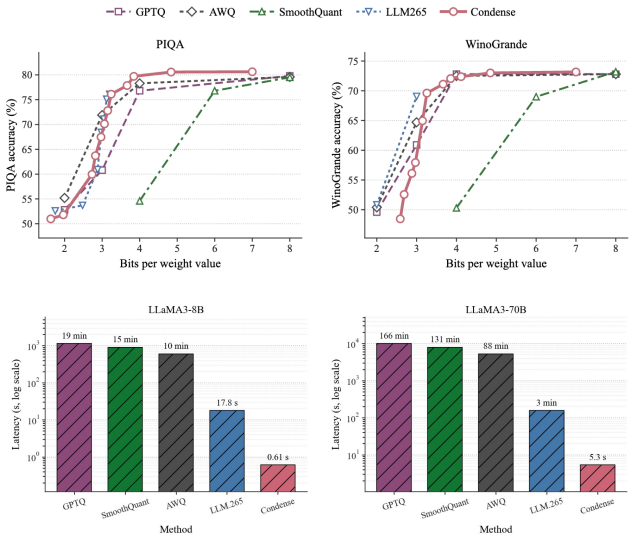


Figure 2: Condense is competitive with state-of-the-art quantization across bit rates (top) while compressing/decompressing models substantially faster (bottom).

3 Pipeline Evaluation

We evaluate quality on Llama-3-8B using PIQA [1] and WinoGrande [5] benchmarks. We compare against state-of-the-art approaches:

GPTQ [2], AWQ [4], SmoothQuant [6], and LLM.265 [7]. At 3.85 bits per value, Condense reaches 79.7% PIQA and 72.1% WinoGrande accuracy. Four-bit AWQ reaches 78.3% and 72.5% in the same comparison. Condense provides competitive, tunable performance near the forefront of low bitrate representations. It decompresses Llama-3-8B in 0.61 seconds and Llama-3-70B in 5.3 seconds, respectively, 29–34× faster than LLM.265, the next-fastest measured baseline. We report decompression rather than compression time because it is more relevant to most applications. However, runtime for Condense does not vary significantly between compression and decompression.

4 Runtime Design

We introduce serverless inference as a new application for tensor compression. In this runtime, when a request targets a model absent from its assigned GPU, billions of weights must arrive before the first token. ServerlessLLM optimizes scheduling and loading [3], but still transfers the full checkpoint leading to a clear bandwidth bottleneck. Condense instead stores and moves compact tensors, then reconstructs tensors directly in their final GPU addresses. Unlike post-training quantization such as AWQ [4], inference kernels never need to consume the compact format, though designing such kernels is possible as an additional runtime feature.

Checkpoint conversion records compressed and final raw tensor offsets. At load time, Condense allocates a compressed staging region and the final model region on GPU at the same time. Projection packages are then copied asynchronously from pinned host memory to staging memory. As each group becomes ready, GPU kernels decompress the representation back into the final model allocation while later groups are still in flight from disk or host. The CPU parses package structure, but the data path stays on the GPU.

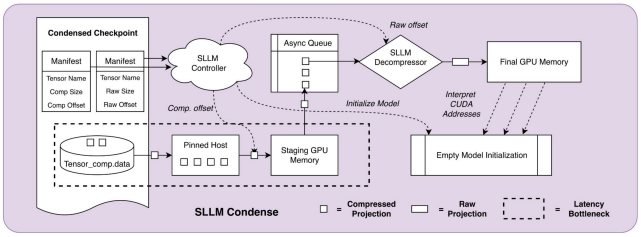


Figure 3: SLLM-Condense overlaps compressed checkpoint transfer with projection-level GPU reconstruction.

The compact representation shortens transfer, while the queue hides most reconstruction overhead behind later copies (Figure 3). The normal BF16 result requires no specialized inference kernels. This format can therefore be optimized as a hardware consideration without affecting the inference engine itself.

5 Runtime Evaluation

We measure complete cold-start model loading time for Llama-3.2-1B, Llama-3.2-3B, Llama-3-8B, and Llama-2-13B. Across four trials per model, Condense loads in 12.47, 30.14, 74.76, and 107.90 seconds. These times are 2.38–2.67× faster than SafeTensors and 1.92–2.17× faster than vanilla ServerlessLLM. The speedup follows the reduction in stored bytes, indicating that transfer remains the bottleneck and compact representation drives the performance gain.

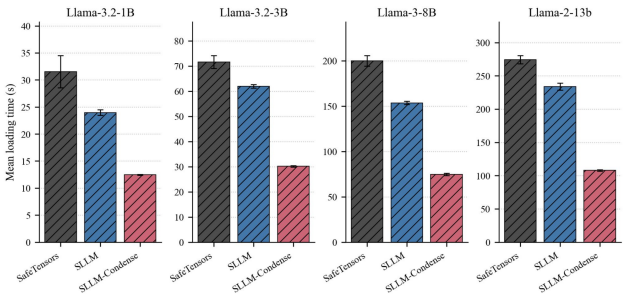


Figure 4: Condense reduces cold-start end-to-end loading time across model sizes. Bars report means over four trials.

6 Conclusion

Storage, transit, and inference are often limited by the immense size of model weights. Condense provides new tensor-specialized methods to achieve ultra-fast but competitive rate-distortion performance across ML domains. We evaluate Condense for serverless inference model management. It moves fewer bytes, directly alleviating expensive cold-start and swapping latencies.

Acknowledgments

This project was supported by the National Science Foundation (2447314) and the Data Science Institute at the University of Chicago.

References

[1] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. PIQA: Reasoning about Physical Commonsense in Natural Language. arXiv:1911.11641 [cs.CL] https://arxiv.org/abs/1911.11641

[2] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. *CoRR* abs/2210.17323 (2022). arXiv:2210.17323 doi:10.48550/ARXIV.2210.17323

[3] Yao Fu, Leyang Xue, Yeqi Huang, Andrei-Octavian Brabete, Dmitrii Ustiugov, Yuvraj Patel, and Luo Mai. 2024. ServerlessLLM: Low-Latency Serverless Inference for Large Language Models. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*. USENIX Association, Santa Clara, CA, 135–153. https://www.usenix.org/conference/osdi24/presentation/fu

[4] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration. In *Proceedings of Machine Learning and Systems*, P. Gibbons, G. Pekhimenko, and C. De Sa (Eds.), Vol. 6. 87–100.

[5] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: an adversarial winograd schema challenge at scale. *Commun. ACM* 64, 9 (Aug. 2021), 99–106. doi:10.1145/3474381

[6] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 38087–38099. https://proceedings.mlr.press/v202/xiao23c.html

[7] Ceyu Xu, Yongji Wu, Xinyu Yang, Beidi Chen, Matthew Lentz, Danyang Zhuo, and Lisa Wu Wills. 2025. LLM.265: Video Codecs are Secretly Tensor Codecs. In *Proceedings of the 58th IEEE/ACM International Symposium on Microarchitecture (MICRO '25)*. Association for Computing Machinery, New York, NY, USA, 445–460. doi:10.1145/3725843.3756078