Research Statement

Jinuk Kim

Machine learning (ML) is transforming how people work and create, and the pace of change is accelerating. In two decades, capabilities have progressed from recognizing handwritten digits to building simple conversational chatbots; in just the past five years, tool-using agents that browse the web and execute code can build software from high-level instructions. Because returns to intelligence *compound* (i.e., new capabilities enable faster subsequent gains), the societal impact will broaden even more rapidly. My goal is to democratize these benefits by making AI systems more efficient: reducing memory, computation, latency, and energy so state-of-the-art models can be trained and deployed under real-world constraints.

My research centers on compression and efficiency across ML algorithms and systems. As ML expands into new domains, heterogeneous architectures, pipelines, and data types introduce distinct constraints; I formalize these design questions as discrete or continuous optimization problems, identify the underlying problem class, and develop tractable, principled solutions. I pair this with engineering efforts to turn ideas into practical artifacts and measurable gains. The remainder of this statement summarizes my contributions across diverse ML settings, distills key lessons, and outlines future directions aimed at addressing efficiency bottlenecks in emerging fields where machine learning systems will play an increasingly important role.

Research Progress

GuidedQuant [1]. The rapid scaling of large language models (LLMs) has unlocked powerful capabilities but also introduced severe challenges in memory consumption and inference latency. Post-training quantization (PTQ) offers a practical path to compression without costly retraining, yet existing methods suffer from a key limitation: some neglect the varying importance of hidden features to the end-task loss, while others capture loss sensitivity but ignore dependencies between weights. In GuidedQuant, we addressed this gap with a unified PTQ framework that incorporates gradient signals from the end-task loss directly into the quantization objective while preserving cross-weight interactions. This design bridges two previously disjoint PTQ paradigms and consistently improves performance across *weight-only scalar, weight-only vector, and weight-and-activation* quantization schemes, enabling drop-in enhancements to state-of-the-art quantizers. Beyond this unifying framework, we further proposed a novel non-uniform scalar quantizer that provably monotonically reduces the quantization objective and empirically outperforms prior scalar methods. Overall, GuidedQuant delivers a principled approach to advancing PTQ, enabling a more accurate compression of LLMs across diverse quantization formats.

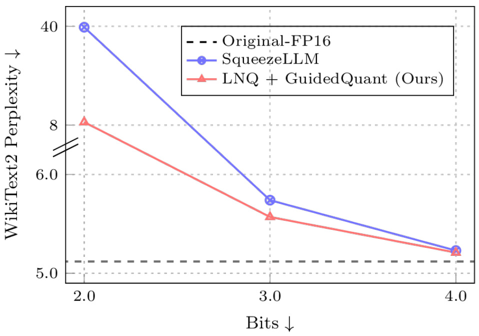


Figure 1: GuidedQuant is a versatile plug-in framework that improves PTQ methods under various quantization schemes.

LayerMerge [2]. In convolutional neural networks, reducing the number of layers can significantly accelerate inference. *Depth compression* approaches remove redundant activation functions and merge consecutive convolution layers, while *layer pruning* directly eliminates convolution layers. However, each modality alone has limitations: depth compression often results in larger kernel sizes that offset latency gains, and layer pruning can degrade representational capacity. To address this, we introduced LayerMerge, a unified framework that jointly prunes convolution layers and activation functions, leveraging their complementary strengths. In LayerMerge, we formulate the joint layer-selection problem as a surrogate optimization and solve it efficiently with dynamic programming. Experiments show that LayerMerge consistently outperforms both standalone depth compression and layer pruning methods, and importantly, the framework applies seamlessly to both classification models (MobileNet-V2) and generative models (DDPM). In summary, LayerMerge offers a general and effective framework for layer-level compression, achieving superior efficiency with minimal performance loss.

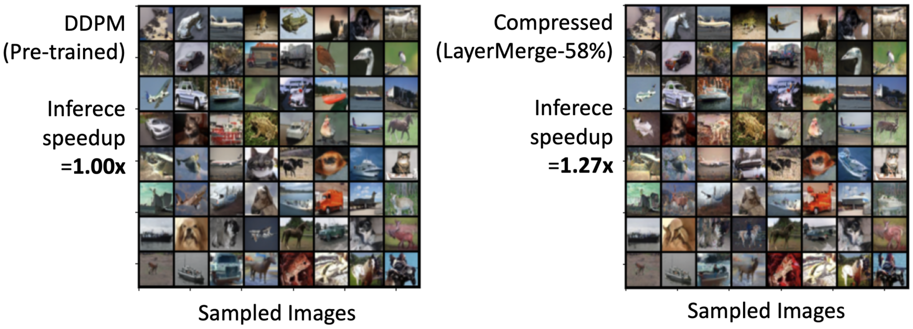


Figure 2: LayerMerge jointly prunes convolution and non-linear activation layers, accelerating both classification and generative models.

Neural Network Depth Compression [3]**.** Prior to LayerMerge, we introduced a principled depth compression framework for convolutional neural network acceleration that optimizes which non-linear activation layers to remove and subsequently merges consecutive convolution layers to accelerate inference. Unlike earlier depth compression methods that relied on heuristic design and restricted search spaces, we formulated the problem as selecting a subset of activation layers to replace with identity functions under a latency constraint. We then proposed a surrogate optimization problem with linear objectives and constraints and developed a dynamic programming algorithm that solves it exactly. This was the first work to frame depth compression in terms of linear objectives and constraints and to provide an exact dynamic programming solution. This foundation directly motivated LayerMerge, where we extended the framework to jointly optimize both convolution layers and activation functions, leveraging their complementary strengths for broader applicability across classification and generative models.

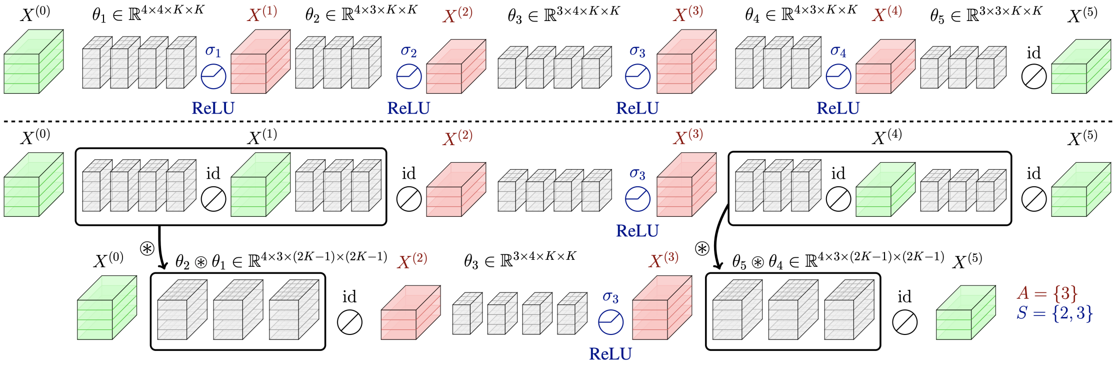

$X^{(0)}$
$\theta_1 \in \mathbb{R}^{4\times4\times K\times K}$
σ_1
ReLU
$X^{(1)}$
$\theta_2 \in \mathbb{R}^{4\times3\times K\times K}$
σ_2
ReLU
$X^{(2)}$
$\theta_3 \in \mathbb{R}^{3\times4\times K\times K}$
σ_3
ReLU
$X^{(3)}$
$\theta_4 \in \mathbb{R}^{4\times3\times K\times K}$
σ_4
ReLU
$X^{(4)}$
$\theta_5 \in \mathbb{R}^{3\times3\times K\times K}$
id
$X^{(5)}$
$X^{(0)}$
$X^{(1)}$
id
id
$X^{(2)}$
σ_3
ReLU
$X^{(3)}$
$X^{(4)}$
id
id
$X^{(5)}$
$\circledast$
$X^{(0)}$
$\theta_2 \circledast \theta_1 \in \mathbb{R}^{4\times3\times(2K-1)\times(2K-1)}$
id
$X^{(2)}$
$\theta_3 \in \mathbb{R}^{3\times4\times K\times K}$
σ_3
ReLU
$X^{(3)}$
$\circledast$
$\theta_5 \circledast \theta_4 \in \mathbb{R}^{4\times3\times(2K-1)\times(2K-1)}$
id
$X^{(5)}$
$A = \{3\}$
$S = \{2, 3\}$

Figure 3: We formulate a subset selection problem that optimizes which non-linear activations layer to remove, which we solve via dynamic programming algorithm.

KVzip [4]**.** Transformer LLMs cache past context as key–value (KV) pairs, making long-context inference memory- and latency-bound. In KVzip, we developed a query-agnostic KV eviction framework that scores each KV pair by how well the underlying model can reconstruct the original context from it and evicts low-importance pairs. This reconstruction-based criterion enables effective cache reuse across diverse prompts and multi-query settings, yielding 3-4× KV-size reduction and roughly 2× faster FlashAttention decoding with negligible accuracy loss, up to 170K-token contexts. On the engineering side, I contributed by implementing the method across different LLM architectures: the Gemma3 architecture, which employs sliding-window attention, and the LLaMA3 architecture under a quantization setting, where weights and activations are quantized to 8-bits and KV caches to 4-bits. KVzip unlocks new applications by producing reusable and accurately compressed KV caches suitable for multiple future queries, which is particularly valuable in real-world scenarios such as enterprise document retrieval–augmented systems or personalized user contexts.

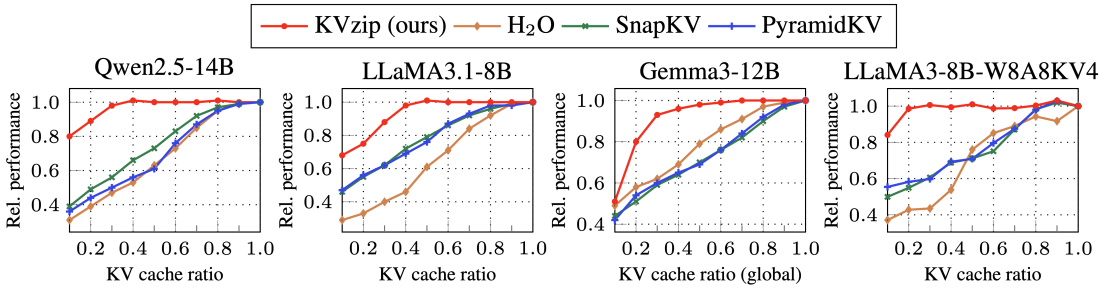
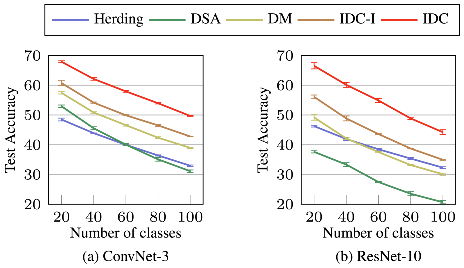


Figure 4: KVzip consistently outperforms query-aware baselines under multi-query workloads, including on LLMs using sliding-window attention (Gemma3-12B) or quantization (LLaMA3-8B-W8A8KV4).

Dataset Condensation [5]. Modern ML systems rely on massive datasets, making training and tuning prohibitively costly in compute and storage. Prior condensation methods synthesize small training sets but are fundamentally limited by optimization procedures that ignore inherent data regularities. We proposed a condensation framework that efficiently parameterizes synthetic examples by exploiting such regularities (e.g., nearby pixels often share similar colors), enabling the generation of multiple diverse samples under a fixed storage budget. On the engineering side, I extended the framework to continual learning by using condensed sets as exemplar memories for past classes; under the same memory budget, this achieved the highest accuracy compared to alternative condensation and coreset approaches. Overall, the method advances dataset condensation by leveraging data regularities and further demonstrates promise in continual learning scenarios.



(a) ConvNet-3

(b) ResNet-10

Figure 5: Leveraging our condensed data (IDC) as exemplars in continual learning yields superior performance.

Ongoing and Future Directions

Looking ahead, I aim to enhance the efficiency of machine learning systems, particularly in domains with the greatest societal and scientific impact. I envision ML technologies that interact with external environments, leverage external tools, discover new science, and integrate into everyday life as ambient systems that transform how we live and work. My goal is to make the benefits of emerging architectures and applications broadly accessible through principled compression research, coupled with proper engineering effort, for real-world impact. Below, I outline ongoing and future research directions that reflect this vision; the list is illustrative and not limited to those enumerated.

- *Efficient QAT for LLMs:* Develop efficient QAT pipelines that scale to trillion-parameter MoE models and remain practical for practitioners.
- *Compressing RL fine-tuned LLMs:* Create policy-aligned quantization and pruning that preserve RL objectives and reward/preference fidelity for agentic and reasoning tasks.

- *Compressing Diffusion LLMs:* Design decoding order-aware compression objectives that mirror inference-time decoding schedules to retain quality at low cost for diffusion LLMs.
- *Compressing Vision-Language-Action (VLA) models:* Build an evaluation suite that reflects end-to-end simulation throughput and develop task-adaptive compression to meet task-specific throughput targets.

References

[1] **Jinuk Kim**, Marwa El Halabi, Wonpyo Park, Clemens JS Schaefer, Deokjae Lee, Yeonhong Park, Jae W. Lee, and Hyun Oh Song. GuidedQuant: Large Language Model Quantization via Exploiting End Loss Guidance. In *International Conference on Machine Learning (ICML)*, 2025.

[2] **Jinuk Kim**, Marwa El Halabi, Mingi Ji, and Hyun Oh Song. LayerMerge: Neural Network Depth Compression through Layer Pruning and Merging. In *International Conference on Machine Learning (ICML)*, 2024.

[3] **Jinuk Kim***, Yeonwoo Jeong*, Deokjae Lee, and Hyun Oh Song (*: Equal contribution). Efficient Latency-Aware CNN Depth Compression via Two-Stage Dynamic Programming. In *International Conference on Machine Learning (ICML)*, 2023.

[4] Jang-Hyun Kim, **Jinuk Kim**, Sangwoo Kwon, Jae W. Lee, Sangdoo Yun, and Hyun Oh Song. KVzip: Query-Agnostic KV Cache Compression with Context Reconstruction. In *International Conference on Machine Learning (ICML) ES-FoMo-III Workshop*, 2025.

[5] Jang-Hyun Kim, **Jinuk Kim**, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset Condensation via Efficient Synthetic-Data Parameterization. In *International Conference on Machine Learning (ICML)*, 2022.

Last updated: August 29, 2025