GenAI at the Edge: Comprehensive Survey on Empowering Edge Devices

Mozhgan Navardi[1], Romina Aalishah [1], Yuzhe Fu [2], Yueqian Lin [2], Hai Li [2], Yiran Chen [2], Tinoosh Mohsenin [1]

[1] Electrical and Computer Engineering, Johns Hopkins University, USA
[2] Electrical and Computer Engineering, Duke University, USA
{mnavard1, raalish1, tinoosh}@jhu.edu, {yuzhe.fu, yueqian.lin, hai.li, yiran.chen}@duke.edu

Abstract

Generative Artificial Intelligence (GenAI) applies models and algorithms such as Large Language Model (LLM) and Foundation Model (FM) to generate new data. GenAI, as a promising approach, enables advanced capabilities in various applications, including text generation and image processing. In current practice, GenAI algorithms run mainly on the cloud server, leading to high latency and raising security concerns. Consequently, these challenges encourage the deployment of GenAI algorithms directly on edge devices. However, the large size of such models and their significant computational resource requirements pose obstacles when deploying them in resource-constrained systems. This survey provides a comprehensive overview of recent proposed techniques that optimize GenAI for efficient deployment on resource-constrained edge devices. For this aim, this work highlights three main categories for bringing GenAI to the edge: software optimization, hardware optimization, and frameworks. The main takeaways for readers of this survey will be a clear roadmap to design, implement, and refine GenAI systems for real-world implementation on edge devices.

Introduction

Generative Artificial Intelligence (GenAI) has shown great promise in text generation, image synthesis, and multimodal content creation. These advancements rely on large-scale models, such as Large Language Models (LLMs), which achieve impressive performance but require substantial computational and memory resources. Traditionally executed on cloud servers, these models introduce latency, and privacy concerns. With the increasing demand for real-time applications and enhanced data security, there is a growing effort to integrate GenAI capabilities directly into edge devices (Nezami et al. 2024; Navardi et al. 2024).

However, implementing high-intensive models on the edge presents significant challenges (Pourmehrani et al. 2024; Kallakuri et al. 2024; Humes et al. 2023). Edge devices, including drones (Navardi et al. 2023), and autonomous systems (Manjunath et al. 2023) benefit significantly from the GenAI capabilities on devices. For instance, drones can generate real-time terrain analysis in remote areas, Autonomous systems can enhance decision-making through local models. Wearable health monitoring could generate personalized insights from biometric data while ensuring privacy through local data processing. To support these applications, specialized edge hardware such as NVIDIA Jetson, and Qualcomm AI Engine have been developed to handle the computational demands of GenAI while maintaining efficiency.

This situation calls for innovative approaches in software optimization including model compression, Neural Architecture Search (NAS). In parallel, hardware optimization including specialized accelerators, attention optimization, and dedicated frameworks address computational and energy constraints at the edge (Ali et al. 2024). These strategies not only reduce model size and inference latency but also address privacy concerns when deploying complex models on edge devices (Navardi et al. 2024). This paper aims to survey existing methods and provide extensive details on implemented GenAI techniques on edge devices. To the best of our knowledge, there is no dedicated survey on GenAI at the edge. By reviewing state-of-the-art techniques from top-tier conferences and journals, this work offers a roadmap for researchers seeking to apply GenAI in edge. The main category of the paper is organized as follows:

- **Software Optimization:** Discusses key strategies for adapting GenAI models to edge devices, including model compression methods (pruning, quantization, and knowledge distillation), NAS, and open-source GenAI models.
- **Hardware Optimization:** Explores hardware accelerators and attention optimization to highlight how they meet GenAI's computational demands while addressing power and resource constraints on edge devices.
- **Frameworks:** Reviews frameworks to improve inference latency, memory, and overall energy efficiency.

Software Optimization

Model Compression

The rapid advancement of GenAI models, while ushering in unprecedented capabilities, has also given rise to increasingly large model architectures that present significant deployment challenges (Guo et al. 2024a). Early attempts to address these challenges explored distributed mobile computing systems that could partition model computation across multiple devices (Mao et al. 2017b,a).

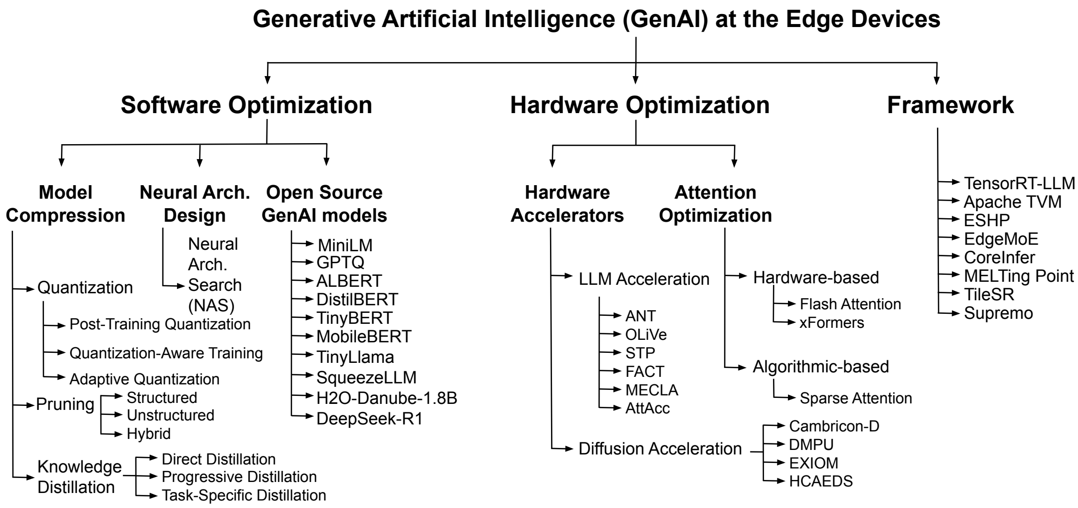

Figure 1: Illustration of the flow of GenAI at the edge

This challenge has since prompted extensive research in model compression techniques, which have evolved along three principal directions to enable broader deployment and accessibility. Firstly, quantization techniques have achieved remarkable efficiency through reduced precision representations, particularly through enhanced activation distribution handling and hardware-optimized strategies. Secondly, methodologies for pruning have advanced from rudimentary magnitude-based techniques to sophisticated hardware-aware structured approaches, enabling considerable model reduction while preserving architectural integrity. Thirdly, knowledge distillation has evolved to incorporate progressive frameworks and multi-teacher architectures, showing particular promise in task-specific applications. Contemporary research emphasizes hardware-aware compression strategies and architecture-specific solutions. While these advancements have enabled the deployment of foundation models with competitive performance metrics, the fundamental challenge persists in optimizing the compression-performance trade-off for edge deployment scenarios.

Quantization Model quantization has emerged as a critical technique for deploying large-scale GenAI models on resource-constrained edge devices. Quantization approaches are broadly categorized into post-training quantization (PTQ) and quantization-aware training (QAT). PTQ methods like OPTQ (Frantar et al. 2023) and AWQ (Lin et al. 2024a) directly convert trained model parameters to lower precision formats, while QAT approaches such as EdgeQAT (Shen et al. 2024) incorporate quantization effects during training. PTQ methods are generally preferred due to their computational efficiency, though recent advances in both approaches have enabled effective compression through sophisticated handling of weight and activation distributions. When applied to LLMs, unique challenges emerge from their heavy-tailed weight distribution. Methods like SmoothQuant (Xiao et al. 2023) address this through distribution smoothing and outlier handling. Mixed-precision approaches (Chen et al. 2024b) determine optimal bit widths for different components based on sensitivity. Recent work like OneBit (Xu et al. 2024) and BitNet (Wang et al. 2023) demonstrates viable 1-bit quantization through distribution-aware schemes. However, challenges remain in maintaining generation quality under extreme compression (Egiazarian et al. 2024).

Diffusion models present their own set of quantization challenges, particularly in handling varying activation distributions across diffusion steps. Approaches like Q-DM (Li et al. 2023c) and Q-Diffusion (Li et al. 2023a) tackle the challenge of varying activation distributions across diffusion steps through adaptive calibration and noise-aware quantization. Specialized temporal-aware quantization methods (Huang et al. 2024) have been developed to handle the unique challenges of the iterative denoising process. Current research focuses on effectively handling dynamic activation ranges and balancing compression ratios with generation quality for edge deployment (Yao et al. 2024).

Pruning Model pruning methods can be broadly categorized into structured and unstructured approaches, each with distinct trade-offs between compression efficiency and hardware compatibility. These techniques have shown particular promise in compressing large-scale generative models while maintaining performance for edge deployment. The field of LLM pruning has recently witnessed several novel approaches. Structured pruning methods like LLM-Pruner (Ma et al. 2023) and edge-optimized approaches (Khiabani et al. 2025) achieve 2× speedup with minimal performance degradation by removing entire structural components.Unstructured approaches like SparseGPT (Frantar and Alistarh 2023) enable up to 60% sparsity in large-scale models, while recent advances in modality-specific pruning techniques have shown promising results across speech, vision, and multimodal domains, with methods like SpeechPrune (Lin et al. 2024b) achieving up to 80% pruning rates while maintaining performance. Hardware-aware methods have become increasingly crucial, as exemplified by Flash-LLM (Xia et al. 2023), which achieves 3× inference speedup through unstructured sparsity-aware system optimization. Semi-structured pruning methods such as E-Sparse (Li et al. 2023b) further advance this direction by leveraging N:M sparsity patterns to maintain hardware compatibility while achieving high compression rates on edge devices.

In the context of diffusion models, methods like Diff-Pruning (Fang et al. 2023) achieve approximately 50% reduction in FLOPs by leveraging Taylor expansion over pruned timesteps while maintaining generative quality. Specialized approaches like LD-Pruner (Castells et al. 2024) implement task-agnostic pruning strategies for Latent Diffusion Models, while DiP-GO (Zhu et al. 2024) demonstrates $4.4\times$ speedup on Stable Diffusion without requiring retraining. Recent work combines gradient-based pruning for mask matrix continuity (Wan et al. 2025) with strategic data pruning (Briq et al. 2024), showing particular promise for edge deployment where both computational efficiency and generation quality are critical (Yan et al. 2024).

Knowledge Distillation. Knowledge Distillation (KD) has emerged as a crucial paradigm for deploying GenAI models on edge devices. The application of KD to language models has led to two main categories: white-box and black-box methods. White-box KD enables student models to match both final predictions and internal representations when the teacher model is open-source (e.g., LLaMA (Touvron et al. 2023b)), while black-box KD works with closed-source models (e.g., GPT-4 (OpenAI 2024)) through API calls (Liu et al. 2024a). Notable advances include MiniLLM (Gu et al. 2024), which introduces a reversed Kullback-Leibler divergence objective to stabilize student updates, and instruction-following distillation approaches that have produced efficient models like Vicuna (Chiang et al. 2023). Recent work in instruction-following KD has enabled compact yet capable models through supervised fine-tuning (Wu et al. 2024), while adaptive distillation methods dynamically adjust the process based on input complexity, focusing learning where improvement is most needed (Liang et al. 2024).

In the domain of diffusion models, KD primarily focuses on accelerating sampling speed to address high inference latency. Progressive distillation (Salimans and Ho 2022) iteratively halves sampling steps (e.g., from 1000 to 1), enabling efficient edge deployment while maintaining generation quality. Single-step approaches (Luhman et al. 2021) compress diffusion teachers into one-step generators, balancing efficiency and fidelity. Teacher-free acceleration methods like DPM-Solver (Lu et al. 2022) and consistency models (Song et al. 2023) reduce inference costs without retraining. Recent advances include two-stage distillation for text-conditional models (Meng et al. 2023) and score distillation sampling (Poole et al. 2023) for 3D generation, showcasing the versatility of distillation.

Neural Architecture Design

Efficient neural architecture design has emerged as a critical research direction to address the increasing complexity and resource demands of modern models, particularly for edge devices (Howard et al. 2017; Elsken et al. 2019). By automating the generation of network architectures while considering specific hardware and constraints, computational overhead, required memory, and power consumption have been improved, while maintaining model performance.

Neural Architecture Search (NAS). Neural Architecture Search (NAS) (Zoph 2016; Elsken et al. 2019) serves as a powerful framework to automate the design of optimal model topologies with strict latency, memory, or power budgets. By systematically exploring a predefined search space such as varying layer depth, width, or connection patterns. NAS algorithms can discover specialized architectures that outperform traditional solutions. In (Zoph 2016), they have proposed the first NAS using reinforcement learning (RL) to determine optimal Recurrent Neural Network (RNN) parameters. Subsequently, this idea was extended to Convonotional Neural Network (CNNs) in (Zoph et al. 2018), where the authors integrated a Sequential Model-Based Optimization (SMBO) approach with a reinforcement mechanism for cell-based searches to find the best configuration.

In the context of GenAI, where large models often dominate in tasks such as text generation or image synthesis, NAS-driven architectures present a promising route to achieve efficiency. There are a limited number of work on NAS in the field of transformers (Liu et al. 2024c). FL-NAS (Qin et al. 2024a) have proposed an approach which leverages LLM to find high-performance DNNs for resource-constrained systems. Moreover, work in (Benmeziane and Maghraoui 2024) proposed a LLM-based methodology for NAS technique in Edge devices. Puzzle (Bercovich et al. 2024) proposed an LLM optimized for inference using NAS under hardware constraints, achieving a 2.17x inference throughput speedup.

Open-Source GenAI Models

The recent advancements in reasoning capabilities of models such as DeepSeek-R1 (DeepSeek-AI et al. 2025) emphasize the power of open research development. One of the key contributions to the advancement in GenAI is open-source innovations, specifically for edge scenarios in which the resources are limited. In these cases, smaller model sizes and less latency besides not losing performance are the main considerations. Therefore, researchers explored various compression methods, leading to models like DistilBERT (Sanh et al. 2019), TinyBERT (Jiao et al. 2020), ALBERT (Lan et al. 2020), MobileBERT (Sun et al. 2020), MiniLM (Wang et al. 2020b), and MiniLMv2 (Wang et al. 2021) each using techniques such as knowledge distillation, parameter sharing, or factorization to make large models smaller while maintaining strong performance.

Beyond these compression-based strategies that are already covered in the previous sections, novelties in architecture further improved efficiency. Reformer (Kitaev et al. 2020) introduced locality-sensitive hashing for attention and reversible residual layers, enabling near-linear complexity for longer sequences. Meanwhile, GPT-NeoX-20B (Black et al. 2022), LLaMA (Touvron et al. 2023b), and LLaMA2 (Touvron et al. 2023a) showed how LLMs could be developed and released collaboratively, making it easier for edge-focused adaptations. Even smaller-scale of these projects such as TinyLlama (Zhang et al. 2024) and H2O-Danube-1.8B (Singer, Philipp others 2024) now offer compact language models tailored to edge constraints, continuing the trend of collaborative research. Similarly, research on instruction tuning (Chung et al. 2022), which trains models to handle various tasks by exposing them to different instructions, reinforced the importance of building flexible and open-source foundations for further innovation.

Researchers have further built on open releases to develop conversational systems, including Alpaca (Taori et al. 2023), Koala (Geng et al. 2023), and Vicuna (Chiang et al. 2023), each developed by fine-tuning LLaMA (Touvron et al. 2023b) on curated datasets, all demonstrating competitive performance against models like ChatGPT and Bard. These models have also served as benchmarks for edge-focused projects such as SqueezeLLM (Kim et al. 2024), which introduces a post-training quantization framework to compress LLMs for more efficient inference, focusing on reducing memory bandwidth, outperforming methods like GPTQ (Frantar et al. 2022), AWQ (Lin et al. 2024a), and SpQR(Dettmers et al. 2023). Taken together, several open-source LLMs have been developed, and some of them are compressed to reduce their size and improve efficiency. These include MPT-7B (MosaicML NLP Team 2023), which implements a 7B-parameter architecture designed for commercial applications; DLite (AI Squared 2023), which scales from 124M to 1.5B parameters; and RedPajama-INCITE (Computer 2023), which spans 3B to 7B parameters. Open-source models and innovations can be valuable for resource-constraint applications, and be fine-tuned for specific tasks to improve their performance.

Hardware Optimization

Hardware Accelerators

Hardware accelerators are typically designed through the software and hardware co-design for specific networks. Algorithmically, data sparsity is enhanced by pruning, and model compression, such as quantization, reduces network size. On the hardware side, specific architectures are designed to bypass sparse or redundant computations, increase data reuse, and minimize data movement, thus enabling energy-efficient acceleration on edge devices. Generative AI (GenAI) includes GAN, LLM, and Diffusion models. While extensive hardware work has focused on optimizing GAN models (Chen et al. 2018; Kang et al. 2021), recent trends have shifted toward LLM and Diffusion models. This section reviews recent efforts in optimizing hardware accelerator for LLM and Diffusion networks, with representative works summarized in Table 1.

LLM Acceleration LLM models have diverse distributions at the tensor or channel levels, numerous studies leverage customized data types to accommodate this challenge. For example, ANT (Guo et al. 2022) introduces a novel data type and employs an adaptive mechanism to determine the most appropriate type for each tensor. Expanding on ANT (Guo et al. 2022) , OliVe (Guo et al. 2023) proposes an outlier-victim pair approach, which provides a more precise representation of outlier distributions in LLM models. Some studies focus on reducing redundant computations in LLM models to improve the energy efficiency during inference. STP (Tambe et al. 2023) proposes a computation-skipping strategy and dynamic data path reconfiguration based on entropy, achieving high energy efficiency with minimal accuracy loss. Furthermore, it has been observed that linear projections contribute significantly to the memory footprint and latency in LLM models. FACT (Qin et al. 2023) introduces an eager prediction method with a leading-one detector and log-based inner-product estimation, reducing computations in both attention and linear projections. MECLA (Qin et al. 2024c) surpasses FACT by decomposing large matrices into smaller sub-matrices to minimize off-chip memory access and re-associating data on-chip for better reuse.

Recently, Computing-in-Memory (CIM) becomes a prominent approach for LLM acceleration. CIM accelerators offer significant energy efficiency gains for GEMM operations. Existing studies typically leverage CIM architectures to accelerate the attention mechanism, while relying on CPUs or GPUs to handle other operations. ASADI (Li et al. 2024b) introduces a sparse attention paradigm based on diagonal compression (DIA) format, enabling highly parallel computation on CIM processors. AttAcc (Park et al. 2024) accelerates batched LLM inference on CIM/NPU heterogeneous systems. Given these developments, it is expected that CIM-based accelerators for LLM models will become more prevalent in the future.

Diffusion Acceleration Diffusion networks have made significant progress recently in various GenAI tasks, with different network architecture from LLM models. These networks generate images or videos through multiple iterations of denoising operations, with highly similar images in consecutive iterations. Consequently, hardware optimizations often leverage inter- and intra-iteration similarity to accelerate Diffusion networks, typically through differential computing and skipping redundant computations.

Cambricon-D (Kong et al. 2024) introduces an approximate ReLU in the Stable Diffusion (SD) network, enabling differential computing for nonlinear functions and addressing the memory overhead associated with full-precision nonlinear calculations in traditional differential computing architectures. DMPU (Qin et al. 2024b) observes that many pixels exhibit minimal changes between consecutive time steps in Diffusion models, and thus proposes a semantic-segment sparse convolution along with a trivial attention exponent inheritance method to skip redundant computations in both the convolution and attention mechanisms, significantly enhancing the energy efficiency. EXION (Heo et al. 2025) presents an FFN-Reuse algorithm that can be applied across iterations, along with an improved eager prediction method for predicting attention scores, which reduces redundant computations and boosts throughput. HCAEDS (Guo et al. 2024b) is the first heterogeneous CIM chip designed for Diffusion models, incorporating a Sign-Magnitude radix-8 Booth CIM macro for integer data and a four-operand exponent CIM macro for floating-point data, achieving a high energy efficiency.

Numerous GenAI hardware studies (Kong et al. 2024; Yang et al. 2024; Wang et al. 2024c) have observed that nonlinear functions can introduce significant latency overhead during the hardware acceleration. These studies opti-

Accelerator	Year	Platform	Technology	Networks	Sparsity/Quantization	Peak Energy Efficiency (TOPS/W)
EXION (Heo et al. 2025)	2025	ASIC simulator	14nm	SD/DiT	✓/ ✓@INT12	11.53
HCAEDS (Guo et al. 2024b)	2024	CIM tapeout	28nm	SD	- / ✓@INT10/BF16	74.34
DMPU (Qin et al. 2024b)	2024	ASIC tapeout	22nm	DDPM	✓/ -	52.01
Cambricon-D (Kong et al. 2024)	2024	ASIC simulator	7nm	SD	✓/ ✓@INT3/FP16	13.34
AttAcc (Park et al. 2024)	2024	CIM simulator	7nm	LLaMA/GPT-3	- / -	2.67× DGX A100
ASADI (Li et al. 2024b)	2024	CIM simulator	28nm	GPT-2/BERT	✓/ -	-
MECLA (Qin et al. 2024c)	2024	ASIC simulator	28nm	LLaMA/BERT	- / ✓@INT8	7.09
STP (Tambe et al. 2023)	2023	ASIC tapeout	28nm	BERT	- / ✓@FP4	18.1
OliVe (Guo et al. 2023)	2023	ASIC simulator	22nm	GPT-2/OPT/BERT	- / ✓@Adaptive 4bit	4× GOBO (Zadeh et al. 2020)
FACT (Qin et al. 2023)	2023	ASIC simulator	28nm	BERT	✓/ ✓@INT8	4.39

Table 1: Hardware Accelerator for GenAI

mize nonlinear functions to enhance overall throughput. Additionally, some studies (Fu et al. 2024; Dong et al. 2024; Yan et al. 2019) have focused specifically on optimizing nonlinear functions and have designed specialized hardware to facilitate network inference. All of these studies indicate a potential research trend on optimizing nonlinear functions in GenAI networks. Combined with techniques such as eliminating redundant computations and data compression, these approaches can enhance hardware acceleration and improve energy efficiency for GenAI systems.

Attention Optimization

Transformers have become the backbone of many GenAI models, but their multi-head self-attention mechanism can dominate runtime and memory usage. Therefore, researchers have explored a range of strategies to optimize attention on *hardware* and *algorithmic* levels.

Hardware-based. FlashAttention (Dao et al. 2022) reorders attention operations to reduce the number of reads and writes between GPU high bandwidth memory (HBM) and on-chip static RAM (SRAM) by splitting queries, keys, and values into smaller blocks, recomputing attention on-chip during the backward pass, and fusing multiple GPU kernels into one. Built on this, FlashAttention-2 (Dao 2023) takes the foundation of memory efficiency and adds better parallelism and work distribution to further increase speed and GPU utilization, especially for longer sequences. Then, FlashAttention-3 (Shah et al. 2024) introduces asynchrony and low-precision computation to further optimize the attention mechanism for modern GPU architectures, which allows for even higher performance and efficiency, along with reduced error for low-precision (FP8) computing. Besides these, xFormers (Lefaudeux et al. 2022), a PyTorch-based library, provides a collection of optimized attention and Transformer blocks, including custom GPU kernels and memory-efficient attention implementations.

Algorithmic-based. Work on sparse attention reduces the quadratic complexity of self-attention by ignoring parts of the input that do not affect the result significantly. Child et al. (Child et al. 2019) pioneered this approach by limiting attention to strided patterns using sparse factorizations of the attention matrix to reduce computation cost while maintaining performance on sequence models. Subsequent techniques like Longformer (Beltagy et al. 2020) by using a combination of sliding window local attention and task-motivated global attention, Big Bird (Zaheer et al. 2020) by combining random, windowed, and global attention to create a sparse attention mechanism, and Linformer (Wang et al. 2020a) by decomposing attention with linear projections to achieve linear complexity introduced various structured sparsity patterns. Meanwhile, Choromanski et al. (Choromanski et al. 2021) developed performer, which uses random feature maps to approximate the softmax function, reducing its time complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

Frameworks

Deploying GenAI models on edge devices might bring challenges because of limited computational power, memory, and latency requirements. To address these constraints, researchers have explored various techniques that simplify computations at both the graph and operator levels. By fusing kernels, reducing redundant operations or parameters, and customizing algorithms to the hardware, these methods enable fast inference for tasks such as large language modeling, super-resolution, and more.

NVIDIA TensorRT and Apache TVM are pioneered compiler-based optimizations by combining graph-level fusion and quantization with lower latency. Likewise, Google's EdgeTPU and Coral stacks enable rapid deployment of compressed models through low-power hardware and software stack. TensorRT-LLM (NVIDIA Corporation 2025) is also a specialized toolkit for accelerating LLM inference on GPUs, including optimized CUDA kernels for attention computations, inflight batching, and quantization.

Beyond these compilers, researchers have developed frameworks customized for various GenAI workloads. For instance, EdgeMoE (Yi et al. 2023) is an engine specifically optimized for Mixture-of-Experts (MoE) language models. By using expert-wise bitwidth adaptation, it supports models with a large number of parameters on edge devices to reduce inference times substantially. Wang et al. introduced CoreInfer (Wang et al. 2024b), achieving over 10× speedup compared to the Huggingface implementation through semantic-based sparse activation that identifies, fixes, and maintains stable neuron activation patterns at the sentence level. MELTing point (Laskaridis et al. 2024) is a mobile benchmarking suite designed to evaluate LLM performance, focusing on energy usage and memory footprints, across smartphones and Jetson platforms. TinyChatEngine (MIT-HAN-Lab 2024) is also, an on-device LLM/VLM Inference Library that uses compression techniques to limit memory budgets while maintaining interactive response times on edge hardware.

In addition to language models, solutions target Super-Resolution (SR) and other vision-based generators. Chen et al. introduced TileSR (Chen et al. 2024a), which splits ultra-high-resolution images into tiles and selects the ones with the highest upscaling difficulty; these tiles are processed in parallel across multiple devices, reducing latency by up to 82% and improving the image quality up to 10% compared to other alternatives such as Supremo (Yi et al. 2022). ESHP (Wang et al. 2024a) combines a difficulty predictor with deep reinforcement learning to distribute SR tasks among CPUs, GPUs, and NPUs, speeding up SR processing without modifying the original architecture of the given SR model. Zhao et al. demonstrated a full-stack SR acceleration framework for embedded GPU devices, which outperformed standard TensorRT baselines in speed due to dictionary compression and operations optimization (Zhao et al. 2021).

FPGAs also provide a promising platform for runtime acceleration. Li et al. proposed a lookup-table (LUT)–based SR pipeline making sharper images while using much less energy without losing image quality (Li et al. 2024a). Other research has combined FFT-based processing with efficient multipliers (Malathi et al. 2024), designed heterogeneous CNN-SNN architectures (Choi et al. 2023), or combined FPGA and GPU via PCIe to achieve real-time SR in microscopic imaging (Gui et al. 2022). For video-specific scenarios, Kim et al. employed pipeline and memory optimizations to reach 60 fps on 4K UHD content (Kim et al. 2019), while Sun et al. developed RNN compression techniques to manage temporal correlations (Sun et al. 2022). On larger multi-core systems Georgis et al. attained speedups over CPU-only baselines via parallelization (Georgis et al. 2019), and Liu et al. achieved real-time 4K SR on edge FPGAs through a DSP-enhanced caching scheme (Liu et al. 2024b). Finally, several system-level revisions help further reduce overhead. Fan et al. (Fan et al. 2023) leveraged codec-side data to skip redundant decoding in video SR, improved performance by up to 9.4×. Deformable 3D convolutional networks, essential in video tasks, were accelerated through tile decoupling and memory optimization by Zhang et al. (Zhang et al. 2022). Even resource-limited devices like the Raspberry Pi can support real-time SR: Osorno-Ortiz et al. integrated 2D-DWT with parallel interpolation to handle HD images in a short time (Osorno-Ortiz et al. 2024).

Conclusion and Future Work

This survey explores the deployment of Generative AI (GenAI) on edge devices, highlighting its potential to reduce latency, enhance privacy, and enable real-time applications. It examines key enablers such as software optimization, hardware specialization, and on-device inference frameworks to overcome embedded system constraints. Despite progress, challenges remain in model personalization and security across distributed nodes. Addressing these issues alongside advancements in model design and hardware acceleration can drive more efficient, scalable, and privacy-preserving GenAI solutions at the edge.

References

AI Squared. 2023. DLite V2. https://huggingface.co/aisquared/dlite-v2-774m.

Ali, A. H.; et al. 2024. Energy-Aware FPGA Implementation of Spiking Neural Network with LIF Neurons. *arXiv:2411.01628*.

Beltagy, I.; et al. 2020. Longformer: The Long-Document Transformer. *arXiv preprint arXiv:2004.05150*.

Benmeziane, H.; and Maghraoui, K. E. 2024. Are Large Language Models Good Neural Architecture Generators for Edge? In *2024 IEEE EDGE*, 162–165.

Bercovich, A.; et al. 2024. Puzzle: Distillation-Based NAS for Inference-Optimized LLMs. *arXiv preprint arXiv:2411.19146*.

Black, S.; et al. 2022. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. *arXiv preprint arXiv:2204.06745*.

Briq, R.; et al. 2024. Data Pruning in Generative Diffusion Models. arXiv:2411.12523.

Castells, T.; et al. 2024. LD-Pruner: Efficient Pruning of Latent Diffusion Models using Task-Agnostic Insights. In *Proceedings of the IEEE/CVF Conference on CVPR Workshops*, 821–830.

Chen, F.; et al. 2018. ReGAN: A pipelined ReRAM-based accelerator for generative adversarial networks. In *23rd ASP-DAC*. IEEE.

Chen, N.; et al. 2024a. TileSR: Accelerate On-Device Super-Resolution with Parallel Offloading in Tile Granularity. In *IEEE Annual International Conference on Computer Communications*.

Chen, Z.; et al. 2024b. Channel-wise mixed-precision quantization for large language models. *arXiv preprint arXiv:2410.13056*.

Chiang, W.-L.; et al. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Child, R.; et al. 2019. Generating Long Sequences with Sparse Transformers. *arXiv preprint arXiv:1904.10509*.

Choi, J.; et al. 2023. A Resource-Efficient Super-Resolution FPGA Processor with Heterogeneous CNN and SNN Core Architecture. *IEEE Transactions on Computers*.

Choromanski, K.; et al. 2021. Rethinking Attention with Performers. In *ICLR*.

Chung, H. W.; et al. 2022. Scaling Instruction-Finetuned Language Models. *arXiv preprint arXiv:2210.11416*.

Computer, T. 2023. RedPajama: An Open Source Recipe to Reproduce LLaMA training dataset.

Dao, T. 2023. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. *arXiv preprint arXiv:2307.08691*.

Dao, T.; et al. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *arXiv:2205.14135*.

DeepSeek-AI; et al. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948*.

Dettmers, T.; et al. 2023. SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression. *arXiv preprint arXiv:2306.03078*. Extended Preprint.

Dong, P.; et al. 2024. Genetic Quantization-Aware Approximation for Non-Linear Operations in Transformers. In *ACM/IEEE DAC*.

Egiazarian, V.; et al. 2024. Extreme Compression of Large Language Models via Additive Quantization. arXiv:2401.06118.

Elsken, T.; et al. 2019. Neural architecture search: A survey. *JMLR*.

Fan, H.; et al. 2023. Co-ViSu: a Video Super-Resolution Accelerator Exploiting Codec Information Reuse. In *International Conference on Field-Programmable Logic and Applications (FPL)*.

Fang, G.; et al. 2023. Structural pruning for diffusion models. In *NeurIPS*.

Frantar, E.; and Alistarh, D. 2023. SparseGPT. arXiv:2307.00026.

Frantar, E.; et al. 2022. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. *arXiv:2210.17323*.

Frantar, E.; et al. 2023. OPTQ: Accurate Quantization for Generative Pre-trained Transformers. In *The Eleventh ICLR*.

Fu, Y.; et al. 2024. SoftAct: A High-Precision Softmax Architecture for Transformers Supporting Nonlinear Functions. *IEEE TCSVT*, 34(9).

Geng, X.; et al. 2023. Koala: A Dialogue Model for Academic Research. https://bair.berkeley.edu/blog/2023/04/03/koala/.

Georgis, G.; et al. 2019. Acceleration techniques and evaluation on multi-core CPU, GPU and FPGA for image processing and super-resolution. *Journal of Real-Time Image Processing*.

Gu, Y.; et al. 2024. MiniLLM: Knowledge Distillation of Large Language Models. In *The Twelfth ICLR*.

Gui, D.; et al. 2022. PCIe-based FPGA-GPU heterogeneous computation for real-time multi-emitter fitting in super-resolution localization microscopy. *Biomedical Optics Express*.

Guo, C.; et al. 2022. Ant: Exploiting adaptive numerical data type for low-bit deep neural network quantization. In *55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*.

Guo, C.; et al. 2023. Olive: Accelerating large language models via hardware-friendly outlier-victim pair quantization. In *Proceedings of the 50th Annual ISCA*, 1–15.

Guo, C.; et al. 2024a. A Survey: Collaborative Hardware and Software Design in the Era of Large Language Models. *arXiv:2410.07265*.

Guo, R.; et al. 2024b. 20.2 A 28nm 74.34TFLOPS/W BF16 Heterogenous CIM-Based Accelerator Exploiting Denoising-Similarity for Diffusion Models. In *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 67, 362–364.

Heo, J.; et al. 2025. EXION: Exploiting Inter-and Intra-Iteration Output Sparsity for Diffusion Models. *arXiv:2501.05680*.

Howard, A. G.; et al. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*.

Huang, Y.; et al. 2024. TFMQ-DM: Temporal Feature Maintenance Quantization for Diffusion Models. In *Proceedings of the IEEE/CVF Conference on CVPR*, 7362–7371.

Humes, E.; et al. 2023. Squeezed Edge YOLO: Onboard Object Detection on Edge Devices. *arXiv preprint arXiv:2312.11716*.

Jiao, X.; et al. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. *arXiv preprint arXiv:1909.10351*.

Kallakuri, U.; et al. 2024. Resource-Aware Saliency-Guided Differentiable Pruning for Deep Neural Networks. In *Proceedings of the Great Lakes Symposium on VLSI 2024*, 694–699.

Kang, S.; et al. 2021. GANPU: An Energy-Efficient Multi-DNN Training Processor for GANs With Speculative Dual-Sparsity Exploitation. *IEEE Journal of Solid-State Circuits*, 56(9): 2845–2857.

Khiabani, Y. S.; et al. 2025. Optimizing Small Language Models for In-Vehicle Function-Calling. *arXiv preprint arXiv:2501.02342*.

Kim, S.; et al. 2024. SqueezeLLM: Dense-and-Sparse Quantization. In *Proceedings of the 41st ICML*.

Kim, Y.; et al. 2019. A Real-Time Convolutional Neural Network for Super-Resolution on FPGA With Applications to 4K UHD 60 fps Video Services. *IEEE TCSVT*.

Kitaev, N.; et al. 2020. Reformer: The Efficient Transformer. In *International Conference on Learning Representations (ICLR)*.

Kong, W.; et al. 2024. Cambricon-D: Full-Network Differential Acceleration for Diffusion Models. In *ACM/IEEE ISCA*.

Lan, Z.; et al. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *ICLR*.

Laskaridis, S.; et al. 2024. MELTing point: Mobile Evaluation of Language Transformers. *arXiv preprint arXiv:2403.12844*.

Lefaudeux, B.; et al. 2022. xFormers: A modular and hackable Transformer modelling library. https://github.com/facebookresearch/xformers.

Li, H.; et al. 2024a. An Energy-Efficient Look-up Table Framework for Super Resolution on FPGA. *IEEE TCSVT*.

Li, H.; et al. 2024b. ASADI: Accelerating Sparse Attention Using Diagonal-based In-Situ Computing. In *IEEE HPCA*.

Li, X.; et al. 2023a. Q-Diffusion: Quantizing Diffusion Models. In *Proceedings of the IEEE/CVF ICCV*.

Li, Y.; et al. 2023b. E-sparse: Boosting the large language model inference through entropy-based n: M sparsity. arXiv:2310.12843.

Li, Y.; et al. 2023c. Q-DM: An Efficient Low-bit Quantized Diffusion Model. In *NeurIPS*, 76680–76691.

Liang, Z.; et al. 2024. Dynamic Self-adaptive Multiscale Distillation from Pre-trained Multimodal Large Model for Efficient Cross-modal Representation Learning. *arXiv preprint arXiv:2404.10838*.

Lin, J.; Tang, J.; Tang, H.; Yang, S.; Chen, W.-M.; Wang, W.-C.; Xiao, G.; Dang, X.; Gan, C.; and Han, S. 2024a. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*.

Lin, Y.; et al. 2024b. SpeechPrune: Context-aware Token Pruning for Speech Information Retrieval. *arXiv:2412.12009*.

Liu, C.; et al. 2024a. Evolving Knowledge Distillation with Large Language Models and Active Learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*.

Liu, H.; et al. 2024b. A High-Performance Accelerator for Real-Time Super-Resolution on Edge FPGAs. *ACM TODAES*.

Liu, Z.; et al. 2024c. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. *arXiv:2402.14905*.

Lu, C.; et al. 2022. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *NeurIPS*, 5775–5787. Curran Associates, Inc.

Luhman, E.; et al. 2021. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv:2101.02388*.

Ma, X.; et al. 2023. LLM-Pruner: On the Structural Pruning of Large Language Models. In *NeurIPS*.

Malathi, L.; et al. 2024. FPGA design of FFT-based intelligent accelerator with optimized Wallace tree multiplier for image super resolution and quality enhancement. *Biomedical Signal Processing and Control*.

Manjunath, T.; et al. 2023. Reprohrl: Towards multi-goal navigation in the real world using hierarchical agents. In *On 37th AAAI The 1st Reinforcement Learning Ready for Production workshop*.

Mao, J.; et al. 2017a. MeDNN: A distributed mobile system with enhanced partition and deployment for large-scale DNNs. In *2017 IEEE/ACM ICCAD*, 751–756. IEEE.

Mao, J.; et al. 2017b. MoDNN: Local distributed mobile computing system for Deep Neural Network. In *DATE, 2017*, 1396–1401.

Meng, C.; et al. 2023. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF CVPR*, 14297–14306.

MIT-HAN-Lab. 2024. TinyChatEngine: On-Device LLM Inference Library. https://github.com/mit-han-lab/TinyChatEngine.

MosaicML NLP Team. 2023. Introducing MPT-7B: A new standard for open-source, commercially usable LLMs. https://www.mosaicml.com/blog/mpt-7b.

Navardi, M.; et al. 2023. MLAE2: Metareasoning for latency-aware energy-efficient autonomous nano-drones. In *IEEE ISCAS*.

Navardi, M.; et al. 2024. MetaTinyML: End-to-End Metareasoning Framework for TinyML Platforms. *IEEE ESL*.

Nezami, Z.; et al. 2024. Generative AI on the Edge: Architecture and Performance Evaluation. *arXiv preprint arXiv:2411.17712*.

NVIDIA Corporation. 2025. TensorRT-LLM. https://github.com/NVIDIA/TensorRT-LLM.

OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Osorno-Ortiz, R. J.; et al. 2024. Implementation of the image super-resolution DWT based algorithm on Raspberry Pi platform for real-time applications. In *Proceedings of SPIE*.

Park, J.; et al. 2024. AttAcc! Unleashing the Power of PIM for Batched Transformer-based Generative Model Inference. In *Proceedings of the 29th ACM International Conference on ASPLOS*.

Poole, B.; et al. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *The Eleventh ICLR*.

Pourmehrani, H.; et al. 2024. FAT-RABBIT: Fault-Aware Training towards Robustness AgainstBit-flip Based Attacks in Deep Neural Networks. In *2024 IEEE ITC*, 106–110. IEEE.

Qin, R.; et al. 2024a. Fl-nas: Towards fairness of nas for resource constrained devices via large language models. In *ASP-DAC*. IEEE.

Qin, Y.; et al. 2023. FACT: FFN-Attention Co-optimized Transformer Architecture with Eager Correlation Prediction. In *Proceedings of Annual ISCA*. Association for Computing Machinery.

Qin, Y.; et al. 2024b. A 52.01 TFLOPS/W Diffusion Model Processor with Inter-Time-Step Convolution-Attention-Redundancy Elimination and Bipolar Floating-Point Multiplication. In *IEEE Symposium on VLSI Technology and Circuits*.

Qin, Y.; et al. 2024c. MECLA: Memory-Compute-Efficient LLM Accelerator with Scaling Sub-matrix Partition. In *IEEE ISCA*.

Salimans, T.; and Ho, J. 2022. Progressive Distillation for Fast Sampling of Diffusion Models. In *ICLR*.

Sanh, V.; et al. 2019. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv:1910.01108*.

Shah, J.; et al. 2024. FlashAttention-3: Fast and Accurate Attention with Asynchrony and Low-precision. *arXiv:2407.08608*.

Shen, X.; et al. 2024. EdgeQAT: Entropy and Distribution Guided Quantization-Aware Training for the Acceleration of Lightweight LLMs on the Edge. *arXiv preprint arXiv:2402.10787*.

Singer, Philipp others. 2024. H2O-Danube-1.8B Technical Report. *arXiv preprint arXiv:2401.16818*.

Song, Y.; et al. 2023. Consistency Models. In *ICML*.

Sun, K.; et al. 2022. An FPGA-Based Residual Recurrent Neural Network for Real-Time Video Super-Resolution. *IEEE TCSVT*.

Sun, Z.; et al. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. *arXiv:2004.02984*.

Tambe, T.; et al. 2023. 22.9 A 12nm 18.1 TFLOPs/W sparse transformer processor with entropy-based early exit, mixed-precision predication and fine-grained power management. In *IEEE ISSCC*.

Taori, R.; et al. 2023. Alpaca: A Strong, Replicable Instruction-Following Model. crfm.stanford.edu/2023/03/13/alpaca.html.

Touvron, H.; et al. 2023a. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Touvron, H.; et al. 2023b. LLaMA: Open and Efficient Foundation Language Models. In *arXiv preprint arXiv:2302.13971*.

Wan, B.; et al. 2025. Pruning for Sparse Diffusion Models based on Gradient Flow. arXiv:2501.01101.

Wang, H.; et al. 2023. Bitnet: Scaling 1-bit transformers for large language models. arXiv:2305.10403.

Wang, Q.; et al. 2024a. An Intelligent Co-Scheduling Framework for Efficient Super-Resolution on Edge Platforms With Heterogeneous Processors. *IEEE Internet of Things Journal*.

Wang, Q.; et al. 2024b. CoreInfer: Accelerating Large Language Model Inference with Semantics-Inspired Adaptive Sparse Activation. arXiv:2410.18311.

Wang, S.; et al. 2020a. Linformer: Self-Attention with Linear Complexity. *arXiv preprint arXiv:2006.04768*.

Wang, W.; et al. 2020b. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers.

Wang, W.; et al. 2021. MiniLMv2: Multi-Head Self-Attention Relation Distillation for Compressing Pretrained Transformers.

Wang, X.; et al. 2024c. DTrans: A Dataflow-transformation FPGA Accelerator with Nonlinear-operators fusion aiming for the Generative Model. In *2024 34th International Conference on FPL*. IEEE.

Wu, M.; et al. 2024. Lamini: Large Language Model Approaches for Generating Assistive Programs. In *EACL*.

Xia, H.; et al. 2023. Flash-LLM: Enabling Cost-Effective and Highly-Efficient Large Generative Model Inference with Unstructured Sparsity. arXiv:2308.04528.

Xiao, G.; et al. 2023. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In *ICML*.

Xu, Y.; et al. 2024. OneBit: Towards Extremely Low-bit Large Language Models. *arXiv preprint arXiv:2402.11295*.

Yan, B.; et al. 2019. RRAM-based spiking nonvolatile computing-in-memory processing engine with precision-configurable in situ nonlinear activation. In *Symposium on VLSI Technology*. IEEE.

Yan, C.; et al. 2024. Hybrid SD: Edge-Cloud Collaborative Inference for Stable Diffusion Models. arXiv:2410.02453.

Yang, G.; et al. 2024. SDA: Low-Bit Stable Diffusion Acceleration on Edge FPGAs. In *34th International Conference on FPL*. IEEE.

Yao, Y.; et al. 2024. Timestep-Aware Correction for Quantized Diffusion Models. In *European Conference on Computer Vision*.

Yi, J.; et al. 2022. Supremo: Cloud-assisted low-latency super-resolution in mobile devices. *IEEE TMC*.

Yi, R.; et al. 2023. EdgeMoE: Fast On-Device Inference of MoE-Based Large Language Models. *arXiv preprint arXiv:2308.14352*.

Zadeh, A. H.; et al. 2020. GOBO: Quantizing Attention-Based NLP Models for Low Latency and Energy Efficient Inference. In *IEEE/ACM International Symposium on Microarchitecture*.

Zaheer, M.; et al. 2020. Big Bird: Transformers for Longer Sequences. In *NeurIPS*.

Zhang, P.; et al. 2024. TinyLlama: An Open-Source Small Language Model. *arXiv preprint arXiv:2401.02385*.

Zhang, S.; et al. 2022. An Efficient Accelerator of Deformable 3D Convolutional Network for Video Super-Resolution. *IEEE TMM*.

Zhao, W.; et al. 2021. A High-Performance Accelerator for Super-Resolution Processing on Embedded GPU. *arXiv:2303.08999*.

Zhu, H.; et al. 2024. DiP-GO: A Diffusion Pruner via Few-step Gradient Optimization. *arXiv preprint arXiv:2410.16942*.

Zoph, B. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.

Zoph, B.; et al. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE CVPR*.