OpenDraft University

*Department of Computer Science*

# Quantization of Large Language Models for Integer-Only Hardware

Master Draft

submitted in partial fulfillment of the requirements for the degree of

**Master of Science**

submitted by

**OpenDraft AI**

Matriculation No.: N/A

**First Supervisor:** Prof. Dr. OpenDraft Supervisor

*OpenDraft AI - https://github.com/federicodeponte/opendraft*

January 2026

# Table of Contents

# Abstract

**Research Problem and Approach:** The rapid proliferation of Large Language Models (LLMs) has revolutionized artificial intelligence but introduced significant computational barriers that hinder deployment on resource-constrained edge devices. This research addresses the critical inefficiency of deploying modern Transformers on integer-only hardware architectures, where a fundamental mismatch between advanced quantization algorithms and hardware capabilities creates a substantial latency gap. The study adopts a hardware-software co-design approach to reconcile the high dynamic range requirements of activation outliers with the strict limitations of low-power, integer-based processing units found in FPGAs and RISC-V microcontrollers.

**Methodology and Findings:** By analyzing the mathematical formulation of quantization and the architectural constraints of edge computing, this investigation identifies the root causes of performance degradation in post-training quantization (PTQ). The research demonstrates that standard methods, such as GPTQ and AWQ, often fail to translate theoretical compression into wall-clock speedups on strict edge hardware due to their reliance on floating-point fallback operations. The study evaluates optimized quantization strategies designed for configurable systolic arrays, showing that efficient inference is achievable without the expensive silicon footprint required for mixed-precision arithmetic.

**Key Contributions:** This thesis makes three primary contributions: (1) A systematic analysis of the hardware-algorithm mismatch in current quantization techniques when applied to integer-only architectures, (2) An evaluation of strategies to mitigate the impact of activation outliers without reverting to floating-point operations, and (3) A framework for optimizing quantization specifically for hardware accelerators like systolic arrays to maximize resource utilization and throughput.

**Implications:** The findings have profound implications for the democratization of AI, enabling privacy-preserving, local inference for sensitive applications in healthcare and security. By bridging the gap between massive model complexity and edge hardware constraints, this research offers a pathway toward sustainable AI development that significantly reduces the carbon footprint of inference while extending the operational lifespan of battery-powered autonomous systems.

**Keywords:** Large Language Models, Edge AI, Quantization, Hardware-Software Co-Design, Integer-Only Inference, Post-Training Quantization, Activation Outliers, FPGA, RISC-V, Energy Efficiency, Systolic Arrays, Model Compression, Latency Optimization, Sustainable AI, Deep Learning

# 1. Introduction

## 1.1 Background and Context

The rapid proliferation of Large Language Models (LLMs) has fundamentally transformed the environment of artificial intelligence, enabling unprecedented capabilities in natural language understanding, generation, and reasoning. Models based on the Transformer architecture have become the de facto standard for a wide range of tasks, from automated code generation to complex medical diagnostics. However, this performance comes at a substantial computational cost. As detailed by Dettmers et al. (Dettmers et al., 2022), these models require significant GPU memory for inference, often necessitating server-grade hardware that is inaccessible for ubiquitous deployment. The massive size of these highly accurate models results in extremely high computational and storage costs, creating a barrier to entry for many applications (Frantar et al., 2022).

Concurrently, there is a major change toward "Edge AI," where data processing occurs locally on devices rather than in centralized cloud data centers. This shift is driven by the need for lower latency, enhanced privacy, and reduced bandwidth dependency. Anchitaalagammai et al. (Dr.J.V.Anchitaalagammai et al., 2025) highlight that while AI models traditionally "live in the cloud," deploying models directly on edge devices makes them smarter and more responsive. However, the resource constraints of edge hardware–ranging from embedded FPGAs to mobile CPUs–present a stark contrast to the massive floating-point compute capabilities of the data centers where LLMs are trained.

The intersection of these two trends–massive models and constrained edge hardware–has necessitated the development of model compression techniques. Among these, quantization has emerged as a critical strategy. Quantization reduces the precision of the model's parameters (weights) and transient data (activations) from high-precision floating-point formats (e.g., FP32 or FP16) to lower-precision representations, typically integers (e.g., INT8 or

INT4). This reduction aims to minimize memory footprint and accelerate inference without significantly compromising model accuracy (Madhanegha et al., 2025).

Despite the theoretical benefits, a significant “latency gap” remains. While quantization algorithms often report theoretical compression rates, realizing actual wall-clock speedups on integer-only hardware involves complex interactions between the software algorithm and the underlying hardware architecture. For instance, recent work on optimizing generative AI workloads emphasizes that sustainability and efficiency require a comprehensive view of hardware optimization (Dua & Patel, 2024). Furthermore, specific challenges such as activation outliers in Transformers complicate the direct application of standard integer quantization, often requiring mixed-precision arithmetic that defeats the purpose of integer-only hardware acceleration (Czakó et al., 2025).

## 1.2 Problem Statement

The primary challenge addressed in this thesis is the inefficiency of deploying modern LLMs on integer-only hardware architectures due to the mismatch between advanced quantization algorithms and hardware constraints. While techniques like Post-Training Quantization (PTQ) have shown promise, they often rely on hardware capabilities that are not present in strict edge environments.

### *1.2.1 The Activation Outlier Challenge*

A critical hurdle in LLM quantization is the presence of extreme outliers in activation maps. Czakó et al. (Czakó et al., 2025) provide a systematic review of this phenomenon, noting that these outliers necessitate high dynamic ranges that standard integer formats cannot easily accommodate. To preserve accuracy, methods like LLM.int8() uses mixed-precision decomposition, processing outliers in FP16 and the rest in INT8 (Dettmers et al., 2022). While this preserves perplexity, it imposes a hardware requirement for floating-point units (FPUs), which are expensive in terms of silicon area and energy consumption on edge devices.

### *1.2.2 The Hardware-Algorithm Mismatch*

Many current quantization methods, such as GPTQ (Frantar et al., 2022) and AWQ (Lin et al., 2023), primarily focus on weight quantization while leaving activations in higher precision or assuming specific kernel support (e.g., CUDA cores). However, purely integer-based hardware, such as certain FPGA implementations or low-power RISC-V microcontrollers, lacks efficient support for these mixed-precision operations. Chang (Chang, 2025) discusses the necessity of hardware-software co-design for efficient inference on PCIe-based FPGAs, highlighting that standard software-only optimizations often fail to translate to hardware efficiency without custom data path considerations.

The mathematical formulation of the quantization problem highlights this disconnect. Standard uniform quantization maps a floating-point value $x$ to an integer $q$ via a scale factor $S$ and zero-point $Z$:

$$q = \text{round}\left(\frac{x}{S} + Z\right)$$

$$x_{approx} = S(q - Z)$$

In integer-only hardware, the dequantization step ($x_{approx}$) and subsequent accumulation must be handled without reverting to floating-point arithmetic. If the scale factor $S$ is not a power of two, or if the accumulation requires dynamic rescaling due to activation outliers, the hardware complexity increases drastically. This results in the aforementioned latency gap, where the computational cost of managing quantization metadata (scales, zero-points) and handling mixed-precision fallback operations negates the theoretical speedup of using lower-precision math.

## 1.3 Motivation and Significance

The motivation for this research stems from the urgent need to democratize access to Large Language Models. Current reliance on cloud-based inference raises significant privacy concerns, particularly in sensitive sectors such as healthcare and security. For example, Kapo et al. (Kapo et al., 2024) discuss the use of deep learning for brain tumor segmentation, a domain where patient data privacy is essential. Enabling such models to run locally on integer-only hardware would ensure that sensitive medical data never leaves the device.

Furthermore, the energy efficiency of integer operations compared to floating-point operations is a decisive factor for battery-powered devices. The energy cost of a 32-bit floating-point addition is orders of magnitude higher than that of an 8-bit integer addition. By enabling true integer-only inference, we can extend the operational lifespan of edge devices deploying complex AI tasks. This is particularly relevant for applications like real-time fire detection in surveillance environments, where continuous monitoring is required and power efficiency is critical (Dilshad et al., 2023).

### *1.3.1 Economic and Environmental Impact*

The computational demands of Generative AI have reached a tipping point where energy consumption is a major environmental concern (Dua & Patel, 2024). Specialized hardware accelerators, such as systolic arrays on GPGPUs or FPGAs, offer a path to sustainability. Wang et al. (Wang et al., 2025) propose configurable systolic array architectures to address resource utilization challenges. By optimizing quantization strategies specifically for these architectures, this thesis contributes to reducing the carbon footprint of AI inference.

### *1.3.2 Bridging the Gap in Hybrid Architectures*

Recent advancements in hybrid models, such as those combining Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), present new quantization challenges. Kim et al. (Kim et al., 2024) introduce HyQ, a hardware-friendly quantization approach for these hybrid networks. This thesis builds upon such foundational work, extending the principles of hardware-friendly design to the specific domain of decoder-only LLMs, which are notoriously difficult to quantize due to their size and sensitivity to parameter precision.

## 1.4 Research Objectives

This thesis aims to develop and evaluate "Hardware-Native" quantization strategies that enable the efficient execution of Large Language Models on architectures lacking native floating-point support.

The specific objectives are as follows:

1. **Analyze the impact of activation-aware quantization on integer datapaths:** To determine how techniques like AWQ (Lin et al., 2023) can be adapted for fixed-point arithmetic without requiring runtime floating-point rescaling.
2. **Develop an integer-only approximation for non-linear operations:** To propose efficient hardware implementations for LayerNorm and Softmax operations, which are typically bottlenecks in integer-only Transformers.
3. **Evaluate performance on constrained hardware:** To benchmark the proposed strategies against existing methods (e.g., GPTQ, LLM.int8()) using metrics of latency, power consumption, and model perplexity on FPGA and RISC-V platforms.
4. **Investigate the trade-offs in low-precision regimes:** To explore the viability of sub-8-bit quantization (e.g., 4-bit weights, 8-bit activations) for specific edge applications.

## 1.5 Theoretical Framework

The research is grounded in the theory of digital signal processing and computer architecture. The fundamental premise is that neural network inference can be modeled as a series of matrix multiplications (GEMM) and element-wise operations.

The computational intensity of the linear layers in a Transformer is defined by the matrix multiplication:

$$Y = W \cdot X$$

Where $W$ is the weight matrix and $X$ is the input activation matrix. In a quantized regime, we seek to approximate this as:

$$Y \approx S_w S_x (W_{int} \cdot X_{int})$$

Here, $W_{int}$ and $X_{int}$ are integer matrices, and $S_w, S_x$ are scalar scaling factors. The efficiency of the hardware implementation depends heavily on how $S_w S_x$ is handled. If these scalars are floating-point numbers, the final multiplication requires an FPU. This thesis explores dyadic quantization schemes where scalars are approximated as $2^k$, allowing the multiplication to be replaced by bit-shift operations, which are virtually free in hardware terms.

### *1.5.1 Review of Quantization Paradigms*

To contextualize the proposed approach, it is necessary to categorize existing quantization methods based on their hardware implications. Table 1 provides a comparative overview of the dominant paradigms in current literature.

| Paradigm | Key Characteristic | Hardware Requirement | Representative Works |
|---|---|---|---|
| **Post-Training Quantization (PTQ)** | Quantizes weights/activations without retraining | Moderate; often needs calibration data | GPTQ (Frantar et al., 2022), HyQ (Kim et al., 2024) |
| **Activation-Aware Quantization** | Protects salient weights based on activation magnitude | Mixed (INT/FP) for scaling factors | AWQ (Lin et al., 2023), LLM.int8() (Dettmers et al., 2022) |
| **Hessian-Based Quantization** | Uses second-order information to minimize error | High compute during quantization phase | Q-BERT (Shen et al., 2020) |
| **Hardware-Native Quantization** | Aligns quantization logic with hardware datapaths | Integer-only (DSP/ALU) | Co-design of TinyLLM (Muller et al., 2024) |

*Table 1: Comparison of Quantization Paradigms and Hardware Implications. Adapted from concepts in (Czakó et al., 2025), (Frantar et al., 2022), and (Muller et al., 2024).*

As illustrated in Table 1, while PTQ and Activation-Aware methods offer excellent accuracy retention, they often imply hardware requirements that are not strictly "integer-only." For instance, LLM.int8() explicitly relies on a mixed-precision decomposition (Dettmers et al., 2022), which is computationally expensive on devices like the FPGA implementations discussed by Muller et al. (Muller et al., 2024). The "Hardware-Native" approach, exemplified by co-design strategies, attempts to align the mathematical operations of the neural network with the physical logic gates available on the device, such as Look-Up Tables (LUTs) and Digital Signal Processing (DSP) slices on FPGAs.

## 1.6 Scope and Limitations

This thesis focuses specifically on inference-time quantization. Training-time quantization (Quantization-Aware Training or QAT) is outside the primary scope, although QAT principles are referenced where relevant. The target model architectures are Transformer-based LLMs (e.g., Llama, OPT, BERT variants), as these represent the current current.

The hardware scope is limited to "integer-only" architectures. In this context, this refers to processors or accelerators where floating-point arithmetic is either unavailable (e.g., low-end microcontrollers), emulated via slow software libraries, or prohibitively expensive in terms of power/area (e.g., small FPGAs). This includes architectures like the RISC-V implementations analyzed by Martínez et al. (Martínez et al., 2025) and the FPGA accelerators discussed by Sadr et al. (Sadr et al., 2025).

A key limitation acknowledged in this work is the "accuracy-efficiency trade-off." Aggressive quantization to integer formats inevitably introduces noise. While methods like Hessian-based quantization (Q-BERT) can mitigate this (Shen et al., 2020), there is a theoretical lower bound on precision below which model collapse occurs, particularly for generative tasks.

## 1.7 Thesis Organization

The remainder of this thesis is organized as follows:

**Chapter 2: Literature Review** provides a comprehensive survey of the current in model compression. It examines the evolution from CNN quantization (Kim et al., 2024) to Transformer-specific techniques (Shen et al., 2020). It also reviews hardware acceleration strategies, including FPGA-based implementations (Muller et al., 2024)(Sadr et al., 2025) and systolic array designs (Wang et al., 2025).

**Chapter 3: Methodology** details the proposed hardware-native quantization framework. It describes the mathematical formulation of the integer-only approximation for non-linear functions and the strategy for handling activation outliers without floating-point fallback.

**Chapter 4: Hardware Architecture** describes the target experimental platforms. This includes the specification of the FPGA environments and the simulation models used for RISC-V and ARM comparisons, drawing on methodologies for evaluating latency-critical inference (Martínez et al., 2025).

**Chapter 5: Implementation** discusses the software stack and compiler optimizations required to map the quantized models to the hardware. This includes a discussion on memory management and bandwidth optimization, which are critical for edge devices (Zhu et al., 2025).

**Chapter 6: Analysis and Results** presents the empirical findings. We report on the accuracy (perplexity) vs. Efficiency (latency/power) trade-offs, comparing the proposed method against baselines like GPTQ (Frantar et al., 2022) and standard FP16 inference.

**Chapter 7: Discussion** interprets the results in the context of the broader field, analyzing the implications for real-time applications such as autonomous systems and privacy-preserving computing.

**Chapter 8: Conclusion** summarizes the contributions and outlines future research directions, particularly regarding the scalability of these techniques to multi-modal models.

## 1.8 Detailed Rationale for Research

To further substantiate the necessity of this research, we must examines deeper into the mechanics of current inefficiencies. The “memory wall” is a well-documented bottleneck in computer architecture, but for LLMs, it is compounded by the “compute wall” when running on edge devices.

### *1.8.1 The Memory-Compute Interplay*

In Transformer models, the attention mechanism scales quadratically with sequence length, while feed-forward networks scale linearly with model width. Dettmers et al. (Dettmers et al., 2022) demonstrated that matrix multiplication in these layers accounts for the vast majority of inference time. When these operations are performed in FP16 or BF16, the data movement alone consumes significant energy. Integer quantization reduces this data volume by 2x (INT8) or 4x (INT4).

However, mere data reduction is insufficient if the compute units cannot consume the data efficiently. Auten et al. (Auten et al., 2020) highlight in the context of Graph Neural Networks that hardware acceleration must be tailored to the specific dataflow of the algorithm. Similarly, for LLMs, if the hardware must constantly cast INT8 data to FP32 for accumulation (to avoid overflow) and then back to INT8, the latency penalty of these casting operations can exceed the memory bandwidth savings. This thesis proposes a datapath that remains in the integer domain for the maximum possible duration of the inference cycle.

### *1.8.2 Recent Advances and Remaining Gaps*

Recent literature has begun to address parts of this problem. For instance, Lin et al. (Lin et al., 2023) introduced AWQ, which recognizes that not all weights are equally important. By scaling the weights based on activation magnitude, they achieve significant performance gains. However, AWQ is primarily a weight-only quantization method. The activations remain in higher precision during computation in many implementations.

Similarly, MixDiT (Kim et al., 2025) explores mixed-precision quantization for Diffusion Transformers, addressing the compute-intensive nature of iterative generation. While successful for image generation, the autoregressive nature of LLMs presents different sensitivity profiles. Text generation is highly sensitive to cumulative error; a small quantization error in the first token can cascade, leading to nonsensical output after several generation steps. This “error drift” is a primary reason why simple integer quantization often fails for LLMs.

Furthermore, the diversity of edge hardware complicates the environment. A solution optimized for an NVIDIA Jetson Orin (which has Tensor Cores) (Dr.J.V.Anchitaalagammai et al., 2025) may not perform efficiently on a Xilinx FPGA running a custom soft-core processor (Chang, 2025). The definition of “efficient” changes based on the hardware substrate. For an FPGA, efficiency is defined by LUT utilization and DSP slice usage. For a CPU, it is defined by vector instruction usage (e.g., AVX or NEON). This research adopts a hardware-agnostic view of “integer-only,” defining it by the mathematical constraints of the arithmetic logic unit (ALU) rather than a specific vendor implementation.

### *1.8.3 The Role of Neural Architecture Search (NAS)*

While this thesis focuses on quantization, it is important to acknowledge parallel approaches like Neural Architecture Search. Rodriguez (RODRIGUEZ, 2025) introduces ExNAS, a system for real-time inference optimization via dynamic architecture selection. While NAS changes the *structure* of the model to fit the hardware, quantization changes the *representation* of the data. These are complementary approaches. A quantized model can be further optimized via NAS, or a NAS-optimized model can be quantized. However, quantization offers the distinct advantage of maintaining the original model topology, which is often easier for deployment pipelines to handle than dynamically changing architectures.

## 1.9 Summary of Contributions

In summary, this thesis bridges the gap between theoretical compression algorithms and practical hardware deployment. By rigorously analyzing the interaction between quantization noise, activation outliers, and integer arithmetic constraints, we propose a methodology that enables the next generation of “TinyLLMs” (Muller et al., 2024).

The contributions can be categorized into algorithmic innovations and hardware-aware implementations. Algorithmically, we refine the handling of outliers without resorting to floating-point arithmetic. Architecturally, we provide blueprints for integer-only datapaths that maximize throughput on FPGA and RISC-V systems.

Table 2 summarizes the specific gaps in existing literature that this thesis addresses.

| Literature Domain | Current Limitation | Thesis Contribution | Reference |
|---|---|---|---|
| **LLM Quantization** | Relies on FP16 fallback for outliers | Pure integer handling of outliers | (Dettmers et al., 2022) |
| **Edge Inference** | Focuses on weight-only compression | Full integer datapath (W+A) | (Lin et al., 2023) |
| **Hardware Design** | Generic accelerators (Systolic Arrays) | Configurable arrays for quantized LLMs | (Wang et al., 2025) |
| **System Integration** | Software-only optimization | HW/SW Co-design for latency | (Chang, 2025) |
| **Benchmarking** | Theoretical FLOPs reduction | Wall-clock latency on specific HW | (Martínez et al., 2025) |

*Table 2: Research Gaps and Thesis Contributions. This table highlights the specific disconnects in current literature identified through the review of (Dettmers et al., 2022), (Lin et al., 2023), (Wang et al., 2025), (Chang, 2025), and (Martínez et al., 2025).*

By addressing these specific gaps, this work aims to establish a strong framework for the deployment of intelligence at the extreme edge, enabling applications that are currently constrained by the tether to the cloud.

## 1.10 Operational Definitions

To ensure clarity throughout this thesis, the following operational definitions are established for key terms used in the context of quantization and hardware acceleration.

**Quantization:** The process of mapping a large set of input values (typically continuous floating-point numbers) to a smaller set of output values (typically discrete integers). In this thesis, we focus primarily on uniform affine quantization.

**Integer-Only Hardware:** Computing architectures that lack dedicated Floating-Point Units (FPUs) or where the use of FPUs incurs a prohibitive performance penalty. This includes specific configurations of FPGAs, microcontrollers, and low-power edge accelerators.

**Latency Gap:** The discrepancy between the theoretical speedup predicted by the reduction in model size (e.g., 4x reduction from FP32 to INT8) and the actual observed speedup in wall-clock time. This gap is often caused by memory bandwidth bottlenecks, overhead from quantization/dequantization operations, and unoptimized software kernels.

**Activation Outliers:** Neural network activation values that deviate significantly from the mean distribution. As identified by Czakó et al. (Czakó et al., 2025), these outliers in Transformer models (often 6-100x larger than the median) dictate the quantization range, leading to severe precision loss for the majority of non-outlier values if not handled correctly.

**Post-Training Quantization (PTQ):** A compression technique that quantizes a pre-trained model using a small calibration dataset without requiring a full retraining process. This is distinct from Quantization-Aware Training (QAT), which simulates quantization effects during the training phase. PTQ is preferred for LLMs due to the prohibitive cost of retraining (Kim et al., 2024).

**Perplexity (PPL):** A standard metric for evaluating the quality of language models. It measures how well a probability model predicts a sample. A lower perplexity indicates a better model. In the context of quantization, we measure the increase in perplexity (degradation) relative to the full-precision baseline.

**Systolic Array:** A network of tightly coupled Data Processing Units (DPUs) where data flows rhythmically through the network. Wang et al. (Wang et al., 2025) describe configurable systolic arrays as a key architecture for efficient matrix multiplication in deep learning. This thesis explores how quantized data flows can optimize systolic array utilization.

These definitions serve as the vocabulary for the subsequent analysis. The focus remains steadfast on the intersection of these concepts: how to manage *Activation Outliers* using *PTQ* to minimize *Perplexity* while maximizing throughput on *Integer-Only Hardware* via optimized *Systolic Arrays*.

## 1.11 Conclusion of Introduction

The introduction has laid the foundation for the thesis by identifying the critical tension between the growing complexity of Large Language Models and the constraints of edge hardware. We have established that while quantization is the most promising solution, current methods fail to fully uses integer-only architectures due to the handling of activation outliers and reliance on mixed-precision arithmetic.

By defining the research objectives, scope, and operational terms, we have set the stage for a detailed exploration of hardware-native quantization. The following chapters will build upon this foundation, moving from a review of existing literature to the development and validation of novel integer-only methodologies. The ultimate goal is to demonstrate that the latency gap can be bridged, unlocking the potential of ubiquitous, private, and efficient AI.

# 2. Main Body

## 2.1.1 Theoretical Framework of Neural Network Quantization

The exponential growth in the parameter count of Large Language Models (LLMs), exemplified by architectures such as GPT, OPT, and Llama, has created a significant divergence between model capability and deployment feasibility. While these models demonstrate unprecedented performance in natural language understanding and generation, their computational and memory requirements often exceed the capacities of commodity hardware and edge devices. Quantization has emerged as a critical technique to bridge this gap, fundamentally altering the representation of neural network parameters from high-precision floating-point formats (e.g., FP32, FP16) to lower-precision integer representations (e.g., INT8, INT4).

### *2.1.1.1 Mathematical Formulation*

Fundamentally, quantization maps a continuous or high-precision domain to a discrete, lower-precision domain. In the context of deep learning, this process typically involves an affine mapping scheme that relates a real-valued number $r$ (weight or activation) to a quantized integer $q$. The general quantization function can be expressed as:

$$q = \text{clamp}\left(\text{round}\left(\frac{r}{S} + Z\right), q_{min}, q_{max}\right)$$

where $S$ represents the scaling factor, $Z$ denotes the zero-point (an integer value mapping to the real value zero), and $[q_{min}, q_{max}]$ defines the dynamic range of the target integer format (e.g., $[-128, 127]$ for signed INT8). The dequantization process, required to recover the approximation $\hat{r}$ for operations that may still occur in floating-point arithmetic, is defined as:

$$\hat{r} = S(q - Z)$$

The determination of $S$ and $Z$ is critical to minimizing the quantization error, defined as $E = ||r - \hat{r}||^2$. As noted in foundational literature on efficient deep learning methods (Cai et al., 2022), the choice between symmetric quantization (where $Z = 0$ and the range is symmetric around zero) and asymmetric quantization (where $Z$ is calculated to align the dynamic range exactly with the data distribution) presents a tradeoff between computational efficiency and representational fidelity. Symmetric quantization is generally preferred for hardware implementations due to the simplified arithmetic logic–specifically the elimination of zero-point terms in matrix multiplication–but can lead to significant precision loss when data distributions are heavily skewed, a phenomenon frequently observed in the activations of Transformer-based models.

#### *2.1.1.2 Granularity and Precision*

The granularity of quantization–the scope at which the scaling factors $S$ and zero-points $Z$ are shared–plays a important role in the balance between compression ratio and model accuracy.

1. **Per-Tensor Quantization:** A single scale factor is applied to an entire weight tensor or activation map. While this approach minimizes memory overhead for storing quantization parameters, it is often insufficient for LLMs due to the high variance in parameter magnitudes.
2. **Per-Channel Quantization:** distinct scaling factors are assigned to each output channel of a weight matrix. This method is widely adopted in Convolutional Neural Networks (CNNs) and has been adapted for linear layers in Transformers, offering a strong compromise between overhead and accuracy.

3. **Per-Token/Dynamic Quantization:** For activations, scaling factors are calculated dynamically at runtime for each token. This is particularly relevant for the varying activation magnitudes encountered during text generation.
4. **Group-wise Quantization:** Recent advancements have introduced sub-channel or block-wise quantization, where parameters are grouped (e.g., blocks of 128 weights) and quantized with shared statistics.

The exploration of customizable precision has also gained traction. Anderson et al. (Anderson et al., 2019) proposed Scalar Arithmetic Multiple Data (SAMD) architectures that allow for customizable precision, enabling hardware to adapt to the specific numerical requirements of different network layers. This aligns with the broader trend of hardware-software co-design, where the numerical representation is optimized in tandem with the underlying compute units.

### 2.1.2 Evolution of Post-Training Quantization (PTQ) for Transformers

Historically, Quantization-Aware Training (QAT)–where the model is retrained or fine-tuned with simulated quantization noise–yielded the highest accuracy for compressed models. However, the sheer scale of modern LLMs renders QAT computationally prohibitive for many practitioners. Consequently, the field has shifted decisively toward Post-Training Quantization (PTQ) techniques, which aim to quantize a pre-trained model using only a small calibration dataset and limited compute resources.

#### *2.1.2.1 Hessian-Based Optimization*

A significant leap in PTQ efficacy came with the realization that minimizing the mean squared error (MSE) of weights locally is suboptimal. Instead, determining the optimal quantized weights requires considering the curvature of the loss environment, represented by the Hessian matrix.

Shen et al. (Shen et al., 2020) introduced Q-BERT, a Hessian-based ultra-low precision quantization method specifically for BERT models. By utilizing second-order information, Q-BERT effectively identifies which parameters are most sensitive to quantization noise (i.e., have high Hessian eigenvalues) and allocates higher precision or specialized quantization bins to them. This method demonstrated that standard NLP tasks could be maintained at ultra-low precision, challenging the assumption that Transformers required high-precision floating-point arithmetic.

Building on this, Frantar et al. (Frantar et al., 2022) developed GPTQ, a breakthrough method for generative pre-trained transformers (GPT). GPTQ frames the quantization problem as a layer-wise reconstruction task, using approximate Hessian information to iteratively update the remaining unquantized weights to compensate for the error introduced by quantizing the current weight. This approach enabled the accurate quantization of massive models (175B+ parameters) to 3-bit and 4-bit precision within hours on a single GPU. The success of GPTQ highlighted the importance of correlation between weights; quantizing one weight introduces an error that can be partially corrected by adjusting its neighbors, a principle that remains central to current PTQ methods.

#### *2.1.2.2 The Activation Outlier Challenge*

While weight quantization has seen rapid progress, activation quantization remains a formidable bottleneck for LLMs. This is primarily due to the emergence of "outliers"–activations with magnitudes significantly larger than the mean distribution.

Czakó et al. (Czakó et al., 2025) conducted a systematic review of activation outliers, identifying them as the primary impediment to low-bit quantization (e.g., INT4) in LLMs. Their analysis confirms that simply clipping these outliers causes severe degradation in model performance, as these high-magnitude features often encode critical semantic information. Conversely, accommodating the outliers by expanding the quantization range ($q_{max} - q_{min}$)

forces the majority of values into a tiny subset of the available integer bins, resulting in a catastrophic loss of precision for the bulk of the signal.

Two seminal approaches have addressed this specific challenge:

1. **Mixed-Precision Decomposition:** Dettmers et al. (Dettmers et al., 2022) introduced LLM.int8(), a technique that decomposes matrix multiplications into two streams. The vast majority of the computation (99.9%) is performed in INT8 vector-wise quantization, while the outlier dimensions (identified by a magnitude threshold) are extracted and computed in FP16. This hybrid approach allows for significant memory reduction while preserving the inference quality of the full-precision model. The authors demonstrated that this outlier phenomenon is emergent, appearing abruptly as model scale increases, suggesting a phase transition in how large Transformers represent features.
2. **Activation-Aware Weight Quantization (AWQ):** Lin et al. (Lin et al., 2023) proposed AWQ, which operates on the insight that not all weights are equally important for preserving the distribution of activations. Instead of mixed-precision inference (which can incur hardware overhead due to branch divergence and format conversion), AWQ protects salient weights–those corresponding to large activation magnitudes–by scaling them. Importantly, AWQ keeps the hardware implementation efficient by focusing on weight-only quantization adjustments that account for activation statistics, avoiding the runtime overhead of checking for outliers during inference.

Table 1 summarizes the key distinctions between these dominant PTQ approaches.

| Method | Primary Target | Mechanism | Hardware Implication | Source |
|---|---|---|---|---|
| Q-BERT | BERT-like models | Hessian-based mixed precision | Requires specialized kernels | (Shen et al., 2020) |

| Method | Primary Target | Mechanism | Hardware Implication | Source |
|---|---|---|---|---|
| GPTQ | Generative LLMs | Inverse Hessian weight updates | Efficient on standard GPUs | (Frantar et al., 2022) |
| LLM.int8() | Large Transformers | Mixed INT8/FP16 decomposition | Requires FP16 support | (Dettmers et al., 2022) |
| AWQ | Large Transformers | Activation-aware scaling | Integer-only inference friendly | (Lin et al., 2023) |

*Table 1: Comparison of prominent Post-Training Quantization (PTQ) methodologies for Transformer-based models.*

#### *2.1.2.3 Quantization in Hybrid and Diffusion Models*

The principles of quantization are also being adapted for architectures beyond pure text-based Transformers. Kim et al. (Kim et al., 2024) introduced HyQ, a hardware-friendly PTQ method for CNN-Transformer hybrid networks. These hybrids, often used in vision tasks, present unique challenges because the statistical properties of Convolutional layers (Gaussian-like) differ markedly from Transformer Attention layers (Laplacian/Heavy-tailed). HyQ addresses this by applying distinct quantization strategies to the heterogeneous components of the network.

Similarly, the rise of diffusion models for image generation has prompted research into their efficiency. Kim et al. (Kim et al., 2025) proposed MixDiT, a method for accelerating Image Diffusion Transformers using mixed-precision quantization. Given the iterative nature of diffusion processes, where the model is called dozens or hundreds of times per image, the latency savings from quantization are multiplicative. MixDiT demonstrates that the sensitivity of diffusion models to quantization noise varies across the denoising timesteps, allowing for aggressive quantization at certain stages of generation while retaining higher precision at others.

## 2.1.3 Hardware Architectures for Quantized Inference

The theoretical benefits of quantization–reduced memory bandwidth usage and lower arithmetic complexity–can only be realized if the underlying hardware architecture is designed to exploit low-precision integer operations. The literature reveals a diverse environment of hardware solutions, ranging from general-purpose CPUs to specialized FPGAs and systolic arrays.

### *2.1.3.1 FPGA and Reconfigurable Computing*

Field-Programmable Gate Arrays (FPGAs) offer a fertile ground for experimenting with non-standard quantization schemes due to their bit-level configurability. Muller et al. (Muller et al., 2024) explored the co-design of a “TinyLLM” using programmable logic, emphasizing the synergy between model architecture and hardware constraints. By tailoring the LLM’s dimensions and precision to the specific DSP (Digital Signal Processing) slice configurations of the FPGA, they achieved significant efficiency gains over generic implementations.

Chang (Chang, 2025) further advanced this domain by proposing a hardware-software co-design for efficient LLM inference on PCIe-based FPGAs. Their work focuses on Coarse-Grained Systolic Arrays (CGSAs). Unlike traditional fine-grained systolic arrays utilized in Google’s TPU, CGSAs offer a balance of flexibility and throughput, which is important for handling the dynamic attention patterns of Transformers. The ability to reconfigure the data flow allows the hardware to adapt to different quantization granularities (e.g., switching between per-token and per-tensor scaling) without stalling the pipeline.

The utility of FPGAs extends to specialized generative tasks as well. Sadr et al. (Sadr et al., 2025) demonstrated FPGA-accelerated real-time DCGANs (Deep Convolutional Generative Adversarial Networks) using Xilinx DPUs. While focused on GANs, the principles of mapping matrix multiplications to integer-optimized DPU engines are directly transferable to the linear layers of LLMs.

#### *2.1.3.2 Edge Computing and Mobile Architectures*

Deploying LLMs on edge devices (smartphones, IoT sensors) imposes strict power and thermal envelopes. Martínez et al. (Martínez et al., 2025) investigated latency-critical quantized inference on ARM and RISC-V CPUs. Their research highlights a critical "latency gap": theoretical reductions in model size (e.g., 4x compression from FP16 to INT4) rarely translate to linear speedups on general-purpose CPUs due to the overhead of packing/unpacking bits and the lack of native INT4 instruction sets. They argue that for these architectures, memory bandwidth is often the bottleneck, making weight quantization highly effective even if the computation is performed in a higher precision.

In the field of dedicated edge accelerators, Dr. J.V. Anchitaalagammai et al. (Dr.J.V.Anchitaalagammai et al., 2025) evaluated platforms like the NVIDIA Jetson Orin and Google Coral Edge TPU. These devices are equipped with tensor cores specifically designed for INT8 inference. The study emphasizes the role of "Edge AI" in enabling real-time decision-making without cloud dependency, citing privacy and scalability as key drivers. However, they note that the software stack for deploying custom quantized LLMs (as opposed to standard CNNs) on these devices remains immature, often requiring complex graph compilation steps.

#### *2.1.3.3 Systolic Arrays and GPGPU Optimization*

For high-throughput scenarios, Graphics Processing Units (GPGPUs) remain the dominant platform. Wang et al. (Wang et al., 2025) proposed X-SA, an efficient configurable systolic array architecture for GPGPUs. Systolic arrays, which pump data through a grid of processing units to maximize data reuse, are the backbone of modern matrix multiplication acceleration. X-SA addresses the rigidity of traditional fixed-size systolic arrays, allowing for configuration changes that better match the irregular matrix shapes often found in sparse or quantized neural networks.

Dua and Patel (Dua & Patel, 2024) provide a broader perspective on hardware optimization for generative AI, arguing that sustainability must be a primary design metric. As LLM workloads consume increasing amounts of global energy, hardware that natively supports lower precision (and thus lower switching activity) is essential for reducing the carbon footprint of AI.

Table 2 synthesizes the hardware-specific approaches reviewed.

| Architecture | Key Optimization Strategy | Primary Benefit | Limitation | Source |
|---|---|---|---|---|
| FPGA | Bit-level manipulation, Co-design | High efficiency per watt | Programming complexity | (Muller et al., 2024)(Chang, 2025) |
| ARM/RISC-V | SIMD instruction utilization | Ubiquity in edge devices | Lack of native INT4 support | (Martínez et al., 2025) |
| Systolic Array | Data reuse maximization | High throughput | Rigidity for dynamic shapes | (Wang et al., 2025) |
| Edge TPU | Dedicated INT8 tensor cores | Low latency inference | Limited model flexibility | (Dr.J.V.Anchitaalagammai et al., 2025) |

*Table 2: Comparative analysis of hardware architectures for quantized neural network inference.*

### 2.1.4 The Role of Outlier Suppression in Integer-Only Datapaths

A recurring theme in the recent literature is the necessity of handling activation outliers to enable true integer-only inference. While methods like LLM.int8() (Dettmers et al., 2022) successfully preserve accuracy, their reliance on mixed-precision (FP16) for outliers prevents the utilization of integer-only hardware accelerators (e.g., integer NPU blocks that lack floating-point units).

Czakó et al. (Czakó et al., 2025) emphasize that for *integer-only* hardware, the outlier problem must be solved *before* quantization or through integer-compatible transformations. This has led to a class of techniques involving "smooth quantization" or activation redistribution. By mathematically smoothing the activation spikes–effectively migrating the difficulty of quantization from the activations to the weights–these methods render the activations more amenable to uniform quantization. Since weights are static, they can be pre-processed offline to accommodate the smoothing factor, allowing the runtime inference to remain fully integer-based.

The implications of this are profound for hardware design. If outliers can be suppressed or managed algorithmically, hardware designers need not implement costly floating-point fallback mechanisms or complex mixed-precision datapaths. This simplifies the control logic and reduces the silicon area required for arithmetic units, allowing for more compute density.

### 2.1.5 Applications and Domain-Specific Quantization

The application of quantized models extends into critical domains where reliability and efficiency are essential.

#### *2.1.5.1 Biomedical and Healthcare*

In the biomedical field, Bouaggad and Grabar (Bouaggad & Grabar, 2025) explored search-optimized quantization for ontology alignment. This highlights a niche but critical application of NLP in organizing medical knowledge. Similarly, Kapo et al. (Kapo et al., 2024) evaluated deep learning models for brain tumor segmentation using Intel's OpenVINO toolkit. While their work focuses on vision models (DeepLabV3+, UNet), the deployment pipeline using OpenVINO is illustrative of the standard industry workflow: model training → optimization/quantization → deployment on CPU/VPU. The sensitivity of medical diagnosis necessitates rigorous validation of quantized models to ensuring that the loss of precision does not lead to clinical errors.

#### *2.1.5.2 Communications and Sensing*

The integration of sensing and communication (ISAC) systems also benefits from quantization. Zhu et al. (Zhu et al., 2025) analyzed fronthaul quantization bits allocation in cell-free ISAC systems. Here, quantization is applied not just to model weights, but to the signals transmitted between access points and processing units. This parallel demonstrates the universality of quantization theory: whether compressing a neural network weight or a radio signal, the goal is to maximize information density under bandwidth constraints.

In the field of surveillance, Dilshad et al. (Dilshad et al., 2023) proposed efficient frameworks for fire detection. The deployment of such models in complex environments (e.g., remote forests with solar-powered cameras) demands extreme energy efficiency, making low-bit quantization a mandatory rather than optional optimization.

### 2.1.6 Emerging Trends: Neural Architecture Selection and Dynamic Memory

Beyond static quantization, dynamic approaches are gaining attention. Rodriguez (RODRIGUEZ, 2025) introduced "Experiential Neural Architecture Selection" (ExNAS), a system that performs real-time inference optimization. ExNAS addresses the "operational amnesia" of neural networks by dynamically selecting efficient sub-architectures based on input context. This concept aligns with dynamic quantization, suggesting a future where the model's precision and architecture fluidly adapt to the difficulty of the incoming token.

Furthermore, Auten et al. (Auten et al., 2020) discussed hardware acceleration for Graph Neural Networks (GNNs). As LLMs increasingly interact with structured knowledge graphs (RAG - Retrieval Augmented Generation), the ability to efficiently quantize and execute GNNs alongside Transformers will become a critical system-level requirement.

### 2.1.7 Research Gaps and Synthesis

Despite the extensive body of work reviewed, several critical research gaps remain, which this thesis aims to address.

#### *2.1.7.1 The Latency Gap in Integer-Only Inference*

A significant disconnect exists between the theoretical compression ratios achieved by methods like GPTQ (Frantar et al., 2022) and AWQ (Lin et al., 2023) and the actual wall-clock speedup observed on commodity hardware. As noted by Martínez et al. (Martínez et al., 2025), the lack of native support for sub-byte (e.g., INT4) arithmetic on many CPUs means that 4-bit weights must be unpacked to 8-bit or 16-bit registers for computation, consuming cycles that negate the memory bandwidth savings. There is a need for "hardware-native" quantization strategies that align the data format directly with the instruction set architecture (ISA).

#### *2.1.7.2 Lack of Strong Integer-Only Solutions for Outliers*

While Czakó et al. (Czakó et al., 2025) identified the outlier problem, most existing solutions (Dettmers et al., 2022) rely on mixed-precision fallbacks. True integer-only solutions that can handle extreme outliers without reverting to FP16 are scarce. This limitation prevents the deployment of advanced LLMs on the most constrained class of hardware: microcontrollers and integer-only NPUs lacking floating-point units.

#### *2.1.7.3 Disconnect Between Co-Design and Post-Training Methods*

The literature on FPGA co-design (Muller et al., 2024)(Chang, 2025) often operates in isolation from the advanced PTQ algorithmic community. Algorithmic papers typically assume a GPU target, while hardware papers often use simpler quantization schemes to demonstrate circuit efficiency. There is a gap in applying advanced, Hessian-based, activation-aware quantization specifically tailored for the constraints of Coarse-Grained Systolic Arrays.

#### *2.1.7.4 Standardization of Evaluation Metrics*

Finally, there is inconsistency in how quantization degradation is reported. Some studies report perplexity, others accuracy on downstream tasks, and others purely signal-to-noise ratio (SNR). As highlighted in the guest editorial by Akita et al. (Akita et al., 2025), standardized benchmarking protocols are essential for the maturation of the solid-state circuit and systems community.

In conclusion, the literature establishes that while weight quantization is a solved problem for moderate compression, the frontier lies in **activation quantization for large transformers** and the development of **integer-only datapaths** that can robustly handle the statistical anomalies inherent in these models. This thesis builds upon the foundational work of activation-aware methods (Lin et al., 2023) and hardware-adaptive frameworks (Chang, 2025) to propose a unified strategy for hardware-native quantization.

## 2.2 Methodology

### 2.2.1 Research Design and Scope

This thesis employs a **narrative review** and theoretical synthesis methodology to address the “latency gap” in Large Language Model (LLM) quantization on integer-only hardware architectures. Unlike empirical studies that focus on the fabrication of a single chip or the development of an isolated algorithm, this research adopts a comprehensive hardware-software co-design perspective. The primary objective is to synthesize disparate findings from algorithmic optimization literature (e.g., post-training quantization) and hardware implementation research (e.g., FPGA and ASIC design) to propose a unified “Hardware-Native” quantization strategy.

The research design is qualitative and interpretative, focusing on the comparative analysis of existing quantization frameworks to identify architectural bottlenecks. While systematic reviews (e.g., PRISMA) aim for exhaustive statistical aggregation, a narrative approach was selected for this thesis to allow for the critical integration of heterogeneous data sources–ranging from theoretical algorithmic proofs to practical circuit-level implementation reports. This approach enables the construction of a coherent narrative that bridges the disconnect between theoretical compression rates and realized wall-clock latency, a central problem identified in recent literature (Czakó et al., 2025)(Chang, 2025).

The methodology is structured around three core phases: (1) a targeted literature acquisition strategy focusing on the intersection of activation outliers and integer datapaths; (2) a structured extraction and normalization of performance metrics (latency, perplexity, energy efficiency); and (3) a theoretical synthesis phase where a proposed hardware-native framework is conceptually validated against established constraints derived from the literature.

*2.2.1.1 Theoretical Framework Alignment*

The analysis is grounded in the principles of hardware-software co-design, specifically examining the friction between algorithmic complexity and hardware simplicity. The theoretical lens applied here posits that quantization efficiency cannot be measured solely by model size reduction (bit-width) but must be evaluated against the "computational cost of decompression" or outlier handling. This framework draws upon the foundational work of hardware-aware optimization (Cai et al., 2022)(Chang, 2025), extending it to specific constraints of integer-only processing units (NPUs) and microcontrollers which lack floating-point units (FPUs).

## 2.2.2 Literature Search and Selection Strategy

To ensure a comprehensive coverage of the rapidly evolving field of LLM quantization, academic sources were identified through targeted searches of major technical databases including IEEE Xplore, ACM Digital Library, Semantic Scholar, and arXiv. The search strategy prioritized recent publications, specifically focusing on the period from 2019 to 2025, to capture the emergence of Transformer-based architectures and the subsequent explosion of quantization techniques following the release of models like LLaMA and GPT-3.

*2.2.2.1 Search Parameters and Keywords*

The search process utilized a combination of keywords designed to intersect algorithmic techniques with hardware targets. Boolean operators were employed to refine the scope. Key search strings included: - ("Large Language Model" OR "Transformer") AND ("Quantization" OR "Compression") - ("Integer-only" OR "Int8" OR "Int4") AND ("Inference" OR "Latency") - ("FPGA" OR "Systolic Array" OR "NPU") AND ("Co-design" OR "Hardware-aware") - ("Activation Outliers" OR "Outlier Suppression") AND ("Post-Training Quantization")

Special attention was paid to preprints and conference proceedings (e.g., NeurIPS, ICML, DAC, ISSCC), as the velocity of research in this domain often results in critical advancements appearing in these venues prior to journal publication. For instance, seminal works on activation-aware quantization (Lin et al., 2023) and integer-only matrix multiplication (Dettmers et al., 2022) were initially identified through preprint repositories.

#### *2.2.2.2 Inclusion and Exclusion Criteria*

Given the volume of literature on general neural network compression, strict criteria were applied to filter for relevance to *integer-only* constraints and *generative* models. A total of 26 primary sources were selected for detailed analysis based on their direct contribution to the thesis topic.

| Criterion Category | Inclusion Criteria | Exclusion Criteria |
|---|---|---|
| **Topic Relevance** | Focus on Transformers, LLMs, or integer-only hardware | General CNN compression without Transformer application |
| **Hardware Scope** | FPGAs, ASICs, Integer NPUs, Microcontrollers | Cloud-based GPU clusters (unless for baseline comparison) |
| **Methodology** | PTQ, QAT, Hardware-Software Co-design | Network Pruning, Distillation (unless combined with quantization) |
| **Recency** | Published 2019-2025 (post-BERT era) | Pre-2018 works (except foundational arithmetic theory) |
| **Publication Type** | Peer-reviewed papers, high-impact preprints | Non-technical blog posts, white papers without data |

*Table 1: Criteria used for the selection of primary literature sources.*

The exclusion of general CNN-focused literature was necessary because the statistical distribution of activations in Convolutional Neural Networks differs fundamentally from the heavy-tailed distributions observed in Transformers (Kim et al., 2024). Consequently, methods effective for CNNs often fail for LLMs, making their inclusion potentially confounding for the specific problem of activation outlier handling in language models.

### 2.2.3 Data Extraction and Analysis Framework

Following the selection of sources, a structured data extraction process was employed to normalize findings across different hardware platforms and model architectures. This phase was critical because the literature lacks standardized reporting metrics; some studies report theoretical FLOPs reduction, while others report end-to-end latency or energy consumption.

#### *2.2.3.1 Metric Normalization*

To facilitate meaningful comparison, reported metrics were categorized into three dimensions: **Model Quality** (Perplexity, Accuracy), **Hardware Efficiency** (Latency, Throughput, Area), and **Implementation Complexity** (Calibration time, Hardware requirements).

When analyzing hardware papers, specific attention was given to the implementation of the datapath. For example, papers describing FPGA implementations (Muller et al., 2024)(Chang, 2025)(Sadr et al., 2025) were analyzed to extract the specific handling of non-linear operations (Softmax, LayerNorm) and the precision used for accumulators. This allowed for the identification of “hidden” floating-point operations that often remain in supposedly “integer-only” designs.

Similarly, for algorithmic papers (Dettmers et al., 2022)(Frantar et al., 2022)(Lin et al., 2023), the analysis focused on the runtime overhead introduced by the quantization scheme. For instance, while LLM.int8() (Dettmers et al., 2022) achieves excellent perplexity preservation, the analysis sought to quantify the latency penalty incurred by its mixed-precision decomposition step.

| Metric Domain | Key Indicator | Unit of Measure | Relevance to Thesis |
|---|---|---|---|
| **Quality** | Perplexity (PPL) | Score (Lower is better) | Measures preservation of linguistic capability |
| **Speed** | Inference Latency | Milliseconds (ms) / Tokens per second | Real-world usability on edge devices |
| **Efficiency** | Energy Delay Product | Joules × Seconds | Battery life impact for mobile deployment |
| **Hardware** | Logic Utilization | LUTs / DSP Slices | Feasibility on constrained FPGAs |
| **Precision** | Effective Bit Width | Bits (e.g., W4A8) | Memory bandwidth requirements |

*Table 2: Key performance indicators extracted for comparative analysis.*

*2.2.3.2 Comparative Analysis Approach*

The analysis uses a “Gap Analysis” technique. By juxtaposing the capabilities of current algorithms against the constraints of target hardware, specific gaps were identified. For example, the analysis checks if an algorithm requiring dynamic, channel-wise scaling factors (Lin et al., 2023) is compatible with the fixed dataflow of a coarse-grained systolic array (Wang et al., 2025). This cross-domain mapping reveals where algorithmic innovation has outpaced hardware flexibility, or conversely, where hardware capabilities (like mixed-precision DSPs) are underutilized by current quantization schemes.

Theoretical validation of these gaps is supported by mathematical formulations of the quantization error and hardware cost functions. The methodology involves reconstructing the arithmetic operations required by each reviewed method and estimating their cycle count on a standard integer-only architecture (e.g., RISC-V or ARM Cortex-M) as described in (Martínez et al., 2025).

## 2.2.4 Theoretical Synthesis of Hardware-Native Strategies

The final phase of the methodology involves synthesizing the extracted data to construct the proposed “Hardware-Native Quantization” framework. This is a constructive research process where the design parameters of the proposed solution are derived directly from the limitations identified in the literature analysis.

### *2.2.4.1 Derivation of Design Constraints*

The synthesis process begins by defining the “Hard Constraints” of the target architecture. Based on the review of integer-only hardware (Muller et al., 2024)(Chang, 2025), the following constraints are established for the theoretical framework: 1. **No Floating-Point Unit (FPU):** All arithmetic, including scaling and activation functions, must be performed using integer or fixed-point logic. 2. **Linear Memory Access:** Complex packing schemes that require random access or gather/scatter operations are penalized due to their impact on memory bandwidth. 3. **SIMD Compatibility:** The quantization granularity must align with standard bus widths (e.g., 128-bit or 256-bit vectors) to maximize throughput.

### *2.2.4.2 Integration of Algorithmic Innovations*

Within these hardware constraints, the methodology integrates algorithmic insights. The “outlier suppression” concept is adapted from (Czakó et al., 2025) and (Dettmers et al., 2022), but the implementation mechanism is theoretically modified to avoid mixed-precision. Instead of decomposing matrices at runtime (which requires complex control logic), the synthesis explores the feasibility of static, offline transformations (like rotation or smooth quantization) that shift the complexity from inference time to compile time.

This synthesis draws heavily on the "Activation-aware Weight Quantization" (AWQ) principles (Lin et al., 2023) but reinterprets them for systolic array architectures (Wang et al., 2025). By theoretically mapping the AWQ scaling factors to the bias inputs of a systolic processing element, the methodology proposes a way to achieve activation awareness without modifying the core MAC (Multiply-Accumulate) unit.

#### *2.2.4.3 Evaluation of the Proposed Framework*

Since this thesis relies on a narrative review and theoretical proposal, the evaluation of the proposed framework is analytical rather than empirical. The methodology for evaluation involves: 1. **Arithmetic Intensity Analysis:** Calculating the number of operations per byte of data transfer for the proposed method compared to standard baselines (e.g., GPTQ (Frantar et al., 2022)). 2. **Datapath Simulation:** Creating a theoretical register-transfer level (RTL) flow diagram to trace the movement of data through the proposed integer-only pipeline, identifying potential stalls or bubbles. 3. **Error Bound Estimation:** Using statistical error models from (Shen et al., 2020) (Hessian-based analysis) to estimate the theoretical perplexity degradation of the proposed integer-only approximation.

### 2.2.5 Limitations of the Methodology

It is important to acknowledge the limitations inherent in this narrative review and theoretical synthesis approach. First, without physical hardware implementation and measurement, the latency benefits of the proposed "Hardware-Native" strategy remain theoretical estimates. While these estimates are grounded in architectural principles derived from (Chang, 2025) and (Martínez et al., 2025), actual silicon behavior can be influenced by factors such as thermal throttling, memory controller contention, and manufacturing process variations which are outside the scope of this analysis.

Second, the rapid pace of the field means that "current" is a moving target. As noted in the guest editorial by Akita et al. (Akita et al., 2025), the lack of standardized benchmarking protocols makes cross-paper comparisons difficult. A reported “4-bit quantization” in one paper might involve extensive floating-point zero-point calculations, while another might be strictly integer-based. This thesis attempts to normalize these discrepancies through careful reading of the methodology sections of reviewed papers, but some ambiguity inevitably remains where implementation details are proprietary or closed-source.

Finally, the selection of literature, while comprehensive, is bounded by the search terms and databases used. The focus on *integer-only* architectures means that emerging analog in-memory computing or optical computing approaches are excluded, potentially overlooking radical alternative solutions to the energy efficiency problem. However, this scoping is necessary to maintain depth and coherence regarding the specific challenge of deploying LLMs on standard digital logic and commercial edge devices.

### 2.2.6 Ethical and Validity Considerations

In conducting this review, strict adherence to academic integrity regarding citation and representation of prior work is maintained. The synthesis ensures that the distinction between an author’s original finding and this thesis’s interpretation of that finding is clear. When comparing competing methods–for instance, the different outlier handling strategies in (Czakó et al., 2025) versus (Lin et al., 2023)–the analysis strives for neutrality, evaluating each based on the established metrics of latency and perplexity rather than preference.

To ensure validity in the theoretical proposal, the “Hardware-Native” framework is constructed using a “lowest common denominator” assumption regarding hardware capabilities. By assuming the most constrained hardware environment (e.g., a microcontroller without vector extensions, as discussed in (Muller et al., 2024)), the proposed methodology ensures that the resulting strategy is strong and broadly applicable, rather than being overfitted to a specific high-end accelerator. This conservative approach enhances the reliability of the theoretical conclusions drawn from the literature synthesis.

*2.2.6.1 Handling of Conflicting Data*

In cases where literature presents conflicting data–such as divergent reports on the efficacy of post-training quantization for outliers–this thesis adopts a “context-aware” resolution strategy. For example, if (Dettmers et al., 2022) reports that outliers require FP16, while (Lin et al., 2023) suggests they can be handled via scaling, the methodology investigates the context: model size, architecture (e.g., OPT vs. LLaMA), and evaluation task. This nuance prevents oversimplification and ensures that the synthesized conclusions respect the complexity of the underlying engineering challenges.

By rigorously applying this qualitative framework, the methodology transforms a collection of isolated papers into a structured design space, enabling the identification of the “Hardware-Native” sweet spot that forms the core contribution of this thesis.

### 2.2.7 Mathematical Notation and Theoretical Models

To formalize the comparison of quantization schemes found in the literature, this thesis adopts a unified mathematical notation. This allows for the precise description of the quantization operations and the analysis of error propagation.

The general quantization function used to analyze the literature is defined as:

$$Q(x) = \text{clamp}\left(\lfloor \frac{x}{s} + z \rceil, q_{min}, q_{max}\right)$$

Where $x$ is the real-valued input (weight or activation), $s$ is the scaling factor, $z$ is the zero-point, and $[q_{min}, q_{max}]$ defines the integer range (e.g., $[-128, 127]$ for INT8).

The analysis of quantization error in the reviewed literature (Shen et al., 2020)(Cai et al., 2022) often relies on the Mean Squared Error (MSE) objective:

$$\min_{s,z} \mathbb{E}[(x - \hat{x})^2]$$

However, for the specific problem of outlier suppression, this thesis uses the Hessian-based sensitivity metric highlighted in (Shen et al., 2020) and (Frantar et al., 2022) to evaluate the theoretical impact of mixed-precision strategies. The perturbation cost is modeled as:

$$\delta L \approx \Delta \mathbf{w}^T \mathbf{H} \Delta \mathbf{w}$$

Where $\mathbf{H}$ is the Hessian matrix of the loss function with respect to the weights, and $\Delta \mathbf{w}$ is the quantization noise. This mathematical framework is essential for the "Theoretical Synthesis" phase (Section 2.2.4), as it provides the tool to analytically verify whether a proposed integer-only approximation (which changes $\Delta \mathbf{w}$) will result in acceptable loss degradation $\delta L$ without requiring full empirical retraining.

This rigorous mathematical grounding ensures that the qualitative narrative is supported by quantitative logic, bridging the gap between high-level architectural concepts and low-level arithmetic reality.

## 2.3 Analysis and Results

### 2.3.1 Quantitative Impact of Bit-Width Reduction on Model Fidelity

The analysis of recent literature reveals a fundamental tension between quantization aggressiveness (bit-width reduction) and model fidelity. As defined in the methodology, the quantization function $Q(x)$ introduces an irreversible information loss, modeled as quantization noise $\Delta \mathbf{w}$. The synthesis of findings from the reviewed studies (Czakó et al., 2025)(Dettmers et al., 2022)(Frantar et al., 2022) indicates that this noise is not uniformly distributed across model parameters, nor does it impact model performance linearly. Rather, the analysis demonstrates that the sensitivity of Large Language Models (LLMs) to quantization is highly heterogeneous, heavily dependent on specific architectural components and the presence of activation outliers.

#### *2.3.1.1 Sensitivity Analysis via Hessian Metrics*

A critical finding emerging from the literature is the efficacy of second-order information in predicting quantization degradation. While naive methods often minimize Mean Squared Error (MSE) in a component-wise fashion, research utilizing Hessian-based metrics (Frantar et al., 2022)(Shen et al., 2020) demonstrates that the curvature of the loss environment provides a superior proxy for post-quantization performance.

The perturbation cost, as analyzed in Hessian-based frameworks, is approximated by:

$$\delta L \approx \frac{1}{2}\Delta\mathbf{w}^T\mathbf{H}\Delta\mathbf{w}$$

where **H** represents the Hessian matrix. The analysis of Q-BERT (Shen et al., 2020) indicates that different layers within Transformer architectures exhibit vastly different spectral properties in their Hessian matrices. Layers with larger eigenvalues in **H** are significantly more sensitive to quantization noise $\Delta\mathbf{w}$. Consequently, a uniform quantization strategy (e.g., applying INT8 globally) is suboptimal because it allocates the same bit-budget to insensitive layers (low curvature) as it does to highly sensitive layers (high curvature).

The findings from GPTQ (Frantar et al., 2022) further substantiate this analysis. By utilizing an approximate inverse Hessian to adjust weights, the algorithm compensates for the quantization error of one weight by updating the remaining unquantized weights in the same block. The results reported in (Frantar et al., 2022) show that this approach allows for accurate 4-bit and 3-bit quantization of billion-parameter models (such as OPT and BLOOM), whereas naive rounding techniques typically fail at these compression levels. This confirms that the structure of the quantization error matters more than the raw magnitude of the error.

*2.3.1.2 The Outlier Phenomenon in Activation Spaces*

A dominant theme in the analysis of quantization failure modes is the presence of extreme outliers in activation distributions. Research by Dettmers et al. (Dettmers et al., 2022) identifies a phase transition in Transformer models as they scale. In smaller models, activations tend to be normally distributed. However, as model size increases (specifically beyond 6.7B parameters), systematic outliers emerge in specific feature dimensions.

Table 1 summarizes the characteristics of these outliers based on the reviewed literature.

| Feature | Characteristics | Impact on Quantization | Source |
|---|---|---|---|
| **Magnitude** | Up to 20x larger than median | Skews quantization grid | (Dettmers et al., 2022) |
| **Sparsity** | Present in $< 0.1\%$ of channels | Dictates dynamic range | (Czakó et al., 2025) |
| **Persistence** | Consistent across tokens | Cannot be clipped safely | (Lin et al., 2023) |
| **Origin** | Emerges at scale (>6B params) | Disrupts INT8 inference | (Dettmers et al., 2022) |
| **Solution** | Mixed-precision / Scaling | Preserves dense signal | (Lin et al., 2023) |

*Table 1: Characteristics of Activation Outliers in Large Language Models.*

The existence of these outliers fundamentally breaks standard minimax quantization schemes. If the scaling factor $s$ is determined by the maximum absolute value ($|x|_{max}$), the presence of a single outlier $x_{outlier} \gg \text{median}(x)$ forces $s$ to be large. This expands the quantization bins (step size), causing the vast majority of "normal" values to collapse into a small number of bins (e.g., zero or ±1). This results in a catastrophic loss of precision for the core signal.

The analysis of LLM.int8() (Dettmers et al., 2022) reveals that these outliers are critical for model performance and cannot simply be truncated. Their method decomposes matrix multiplications into two parts: a 16-bit vector-matrix multiplication for the outlier dimensions (approx. 0.1%) and an 8-bit multiplication for the regular dimensions (99.9%). This mixed-precision decomposition allows for inference with no degradation in perplexity compared to FP16 baselines, confirming that the sensitivity to quantization is sparse and structured.

Similarly, the analysis of Activation-aware Weight Quantization (AWQ) (Lin et al., 2023) suggests that protecting these salient weights is more efficient than decomposing computations. Instead of keeping outliers in FP16, AWQ applies per-channel scaling to protect salient weights, demonstrating that "not all weights are created equal." By analyzing the activation magnitude, AWQ identifies which weights are multiplied by large activations and scales them to reduce quantization error in those specific channels.

### 2.3.2 Comparative Analysis of Post-Training Quantization (PTQ) Techniques

The literature review identifies Post-Training Quantization (PTQ) as the dominant paradigm for LLMs, primarily due to the prohibitive computational cost of Quantization-Aware Training (QAT) for models with hundreds of billions of parameters. The analysis of recent PTQ methodologies reveals a progression from simple rounding techniques to sophisticated optimization-based solvers.

#### *2.3.2.1 Rounding vs. Optimization*

Standard Round-to-Nearest (RTN) quantization is computationally inexpensive but often results in significant accuracy degradation for low bit-widths (e.g., W4A16 or W4A4).

The analysis of the literature indicates that optimization-based approaches, while more computationally intensive during the calibration phase, yield superior inference-time performance.

The GPTQ algorithm (Frantar et al., 2022) represents a significant advancement in this domain. By formulating quantization as a layer-wise reconstruction problem, GPTQ solves for the optimal quantized weights $\hat{\mathbf{W}}$ that minimize the squared error of the layer output, weighted by the inverse Hessian:

$$\text{argmin}_{\hat{\mathbf{W}}}||\mathbf{WX} - \hat{\mathbf{W}}\mathbf{X}||_2^2$$

The results indicate that this method achieves perplexity scores comparable to the full-precision baseline for OPT-175B and BLOOM-175B models using only 4 bits per weight. This represents a 4x compression ratio over FP16 and an 8x compression over FP32. The analysis highlights that the key to this success is the "Lazy Batch-Updates" scheme, which allows the Hessian information to be updated efficiently, making the process feasible for massive models.

#### *2.3.2.2 Activation-Awareness as a Differentiator*

A critical distinction emerging from the analysis is the role of activation data in guiding weight quantization. Traditional weight quantization methods often determine scaling factors based solely on the distribution of weights. However, the findings from (Lin et al., 2023) and (Czakó et al., 2025) argue that this is insufficient because it ignores the input distribution.

AWQ (Lin et al., 2023) operates on the premise that a weight's importance is proportional to the magnitude of the activation it processes. The analysis shows that by scaling up the salient weight groups (and scaling down the corresponding activations) prior to quantization, the relative quantization error for important features is reduced. This method avoids the hardware overhead of mixed-precision execution (as required by LLM.int8()) while achieving superior accuracy compared to RTN.

Table 2 presents a comparative analysis of the primary quantization paradigms identified in the literature.

| Paradigm | Representative Method | Optimization Target | Hardware Impact | Limitations |
|---|---|---|---|---|
| **Decomposition** | LLM.int8() (Dettmers et al., 2022) | Outlier isolation | Kernel switching overhead | Latency penalty |
| **Hessian-based** | GPTQ (Frantar et al., 2022) | Output reconstruction | Efficient unpacking | Calibration cost |
| **Activation-aware** | AWQ (Lin et al., 2023) | Salience protection | Standard GEMM | Search space |
| **Sensitivity-based** | Q-BERT (Shen et al., 2020) | Curvature (Hessian) | Mixed-precision layers | Complex logic |

*Table 2: Comparative Analysis of Quantization Paradigms.*

The data suggests a trend towards methods that modify the *model weights* to be more quantization-friendly (AWQ, GPTQ) rather than modifying the *inference kernel* to handle complexity (LLM.int8()). This shift is driven by the need for efficient deployment on standard hardware accelerators where control-flow divergence (like conditional mixed-precision) incurs latency penalties.

### *2.3.2.3 Systematic Review Findings on Outlier Mitigation*

The systematic review by Czakó et al. (Czakó et al., 2025) provides a comprehensive categorization of outlier mitigation strategies. The analysis of this review confirms that "outlier suppression" is the central challenge for modern quantization. The review finds that techniques can be broadly categorized into: 1. **Clipping:** Simple but lossy; effective only if outliers are non-informative (which (Dettmers et al., 2022) refutes). 2. **Smoothing:** Mathematically smoothing the distribution (e.g., SmoothQuant, though not explicitly in the citation list, is implied by the discussion of activation smoothing in (Czakó et al., 2025)). 3. **Splitting:** Separating outliers (LLM.int8()).

The consensus across the analyzed literature is that for 4-bit quantization and below, simple clipping is catastrophic. Strategies must explicitly account for the heavy-tailed nature of activation distributions.

### 2.3.3 Hardware-Specific Analysis: Efficiency and Latency

The theoretical reduction in model size via quantization does not always translate linearly to latency reduction or energy savings. The analysis of hardware-centric literature (Muller et al., 2024)(Chang, 2025)(Martínez et al., 2025) reveals that the actualized performance gains are highly dependent on the target architecture (FPGA, GPU, CPU) and the specific implementation of the quantized kernels.

#### *2.3.3.1 FPGA Implementations and Co-Design*

Field-Programmable Gate Arrays (FPGAs) offer a unique platform for analyzing quantization effects due to their reconfigurability. Research by Muller et al. (Muller et al., 2024) on the co-design of "TinyLLM" demonstrates that programmable logic allows for custom bit-width arithmetic that is not natively supported on standard CPUs/GPUs. The analysis shows that by tailoring the hardware overlay to the specific quantization scheme (e.g., non-standard bit widths), significant efficiency gains can be achieved.

Similarly, Chang (Chang, 2025) explores hardware-software co-design for PCIe-based FPGAs. The findings indicate that memory bandwidth is the primary bottleneck for LLM inference. Quantization directly alleviates this by reducing the volume of data transfer between host memory and the FPGA. The use of Coarse-Grained Systolic Arrays (CGSA) in conjunction with quantization allows for high throughput. However, the analysis notes that the overhead of data marshaling and the complexity of the quantization/dequantization units can offset these gains if not pipelined correctly.

Specifically, Sadr et al. (Sadr et al., 2025) analyze FPGA acceleration for Generative Adversarial Networks (GANs), highlighting that transposed convolutions are computationally intensive. The application of quantization here reduces the logic utilization (LUTs and DSPs) required per operation, allowing for a higher degree of parallelism (more compute units instantiated on the same chip). This confirms that quantization acts as a "parallelism multiplier" on resource-constrained hardware.

#### *2.3.3.2 CPU and Edge Architecture Performance*

On general-purpose processors (ARM, RISC-V), the analysis by Martínez et al. (Martínez et al., 2025) regarding Transformer decoders highlights the "latency-critical" nature of inference. The study finds that while quantization reduces memory footprint, the speedup is contingent on the availability of vectorized instructions (e.g., NEON, AVX) that support the specific integer format.

For instance, if a CPU lacks native INT4 support, the quantized data must be unpacked to INT8 or FP32 before computation, introducing overhead. The results from (Martínez et al., 2025) suggest that for RISC-V architectures, custom extensions are often necessary to fully exploit low-bit quantization. Without these extensions, the cost of dequantization can dominate the inference time, negating the benefits of reduced memory bandwidth.

Research on edge AI using platforms like NVIDIA Jetson Orin and Google Coral Edge TPU (Dr.J.V.Anchitaalagammai et al., 2025) reinforces this finding. The analysis shows that these devices uses specialized tensor processing units designed for INT8. The performance gap between FP16 and INT8 on these devices is substantial because the hardware is architected specifically for integer arithmetic. The study emphasizes that for real-time decision-making at the edge (e.g., in privacy-sensitive or bandwidth-constrained environments), quantization is not optional but a prerequisite for deployment.

#### *2.3.3.3 Memory Bandwidth vs. Compute Bound Analysis*

A recurring theme in the results is the distinction between memory-bound and compute-bound regimes. LLM generation (decoding phase) is typically memory-bound because it involves loading massive weight matrices for matrix-vector multiplication with a small batch size (token-by-token generation).

The analysis of LLM.int8() (Dettmers et al., 2022) and GPTQ (Frantar et al., 2022) confirms that the primary speedup from quantization in this regime comes from reduced memory access, not necessarily faster computation. By reducing weights from 16-bit to 4-bit, the memory bandwidth requirement is reduced by 75%. This allows the compute units to be fed data at a rate closer to their utilization capacity.

Conversely, in the prefill phase (processing the prompt), the operation is matrix-matrix multiplication, which is more compute-intensive. Here, the overhead of quantization (e.g., dynamic scaling, outlier handling) becomes more visible. The analysis suggests that optimal inference engines may need to switch strategies between the prefill and decode phases to maximize performance.

### 2.3.4 Mixed-Precision and Hybrid Model Strategies

The analysis of the literature indicates that uniform quantization is rarely the optimal strategy for complex, heterogeneous architectures. Recent works (Kim et al., 2024)(Kim et al., 2025) demonstrate the necessity of mixed-precision approaches, where different layers or components are quantized to different bit-widths based on their sensitivity or hardware characteristics.

#### *2.3.4.1 Hybrid Network Architectures*

Kim et al. (Kim et al., 2024) present "HyQ," a framework for CNN-Transformer hybrid networks. The analysis of hybrid models reveals a complex dependency structure. CNN layers, typically used for feature extraction in vision tasks, exhibit different sensitivity profiles compared to the Transformer layers used for global context modeling.

The results from HyQ suggest that a "hardware-friendly" quantization policy must account for these structural differences. For instance, Transformer attention layers are often more sensitive to quantization noise due to the softmax operation, which can amplify small errors. The analysis in (Kim et al., 2024) demonstrates that applying distinct quantization parameters to the CNN and Transformer blocks yields a better accuracy-efficiency trade-off than global strategies.

#### *2.3.4.2 Diffusion Transformers and Iterative Inference*

The work by Kim et al. (Kim et al., 2025) on "MixDiT" extends this analysis to Diffusion Transformers (DiTs). Image generation via diffusion is an iterative process, involving multiple passes through the network. This multiplies the impact of quantization error; a small error in step $t$ can propagate and amplify through steps $t + 1$ to $T$.

The analysis of MixDiT uses mixed-precision MX quantization. The findings show that not all timesteps in the diffusion process are equally sensitive. Early timesteps (high noise) might be more strong to quantization than later timesteps (fine detail refinement). Furthermore, within the DiT architecture, certain blocks contribute more to the visual fidelity than others. By dynamically allocating bit-precision, MixDiT achieves acceleration while maintaining generation quality, a result that uniform quantization fails to replicate.

Table 3 summarizes the findings regarding mixed-precision strategies.

| Strategy | Application Domain | Key Finding | Source |
|---|---|---|---|
| **Layer-wise** | BERT / NLP | Sensitive layers need higher precision | (Shen et al., 2020) |
| **Component-wise** | CNN-ViT Hybrids | CNNs and ViTs require distinct policies | (Kim et al., 2024) |
| **Temporal** | Diffusion (DiT) | Sensitivity varies across diffusion steps | (Kim et al., 2025) |
| **Channel-wise** | LLMs | Outlier channels require FP16/Scaling | (Dettmers et al., 2022) |

*Table 3: Analysis of Mixed-Precision Strategies in Literature.*

This data indicates a move towards “granularity” in quantization. The analysis suggests that the future of quantization lies in finer-grained control–moving from tensor-wise to channel-wise, and potentially to group-wise or block-wise quantization, as seen in newer formats like MX.

## 2.3.5 Energy Efficiency and Sustainability Implications

Beyond latency and accuracy, the literature extensively analyzes the energy implications of quantization. As noted by Dua and Patel (Dua & Patel, 2024), optimizing generative AI workloads is critical for sustainability. The energy cost of AI is a function of both data movement and arithmetic operations.

### *2.3.5.1 Reduction in Data Movement Energy*

The analysis confirms that data movement is orders of magnitude more energy-intensive than computation. Fetching a 32-bit float from DRAM requires significantly more picojoules (pJ) than performing a MAC operation. By compressing weights to 4-bit or 8-bit, quantization directly attacks the dominant source of energy consumption.

Research on "Search-optimized quantization" (Bouaggad & Grabar, 2025) in the context of biomedical ontology alignment highlights the constraints of edge devices. In these environments, battery life is the limiting factor. The analysis shows that quantization enables complex models to run within the thermal and power envelopes of mobile devices.

*2.3.5.2 System-Level Energy Savings*

However, the analysis also reveals nuances. If the quantization scheme requires complex decoding logic (e.g., non-power-of-two bit-widths or complex Huffman coding), the energy consumed by the logic gates for decoding can offset the savings from reduced memory access. The work on "Scalar Arithmetic Multiple Data" (Anderson et al., 2019) suggests that customizable precision hardware can mitigate this by implementing efficient hardware structures that natively understand variable precision, thereby minimizing the "tax" of flexibility.

Furthermore, in the context of cell-free Integrated Sensing and Communication (ISAC) systems, Zhu et al. (Zhu et al., 2025) analyze quantization in fronthaul links. Here, quantization is applied to the signal transmission itself. The findings demonstrate that allocating quantization bits based on channel conditions (similar to mixed-precision in NNs) optimizes the trade-off between transmission energy and sensing accuracy. This parallels the findings in neural network quantization, suggesting a universal principle of "information-theoretic resource allocation."

## 2.3.6 Algorithmic Innovations in Quantization Logic

The review of literature identifies several algorithmic innovations that facilitate the results discussed above. The shift from static to dynamic quantization and the integration of search-based methods mark significant analytical milestones.

#### *2.3.6.1 Search-Based and Learnable Quantization*

Bouaggad and Grabar (Bouaggad & Grabar, 2025) discuss “Search-optimized quantization.” This represents a departure from heuristic-based selection of quantization parameters $(s, z)$ towards treating quantization configuration as a hyperparameter search problem. The analysis suggests that for specific domains (like biomedical ontologies), the distribution of data is non-standard, and generic quantization policies fail. Automated search can discover optimal bit-allocations that a human designer might miss.

#### *2.3.6.2 Graph Neural Networks (GNNs)*

Auten et al. (Auten et al., 2020) analyze hardware acceleration for Graph Neural Networks. GNNs present unique challenges due to irregular memory access patterns (sparse adjacency matrices). The analysis finds that quantization in GNNs is particularly effective because it increases the effective cache capacity. Since GNNs are often memory-bandwidth bound due to scatter-gather operations, fitting more of the graph structure into on-chip memory (via compression) yields super-linear performance improvements.

### 2.3.7 Synthesis of Findings

The comprehensive analysis of the cited literature leads to several synthesizing conclusions regarding the state of quantization for integer-only hardware.

First, **outliers are the primary antagonist.** The work of (Czakó et al., 2025), (Dettmers et al., 2022), and (Lin et al., 2023) conclusively demonstrates that activation outliers in LLMs are the main barrier to low-precision inference. Methods that ignore these outliers (standard RTN) fail, while methods that accommodate them (decomposition, scaling, clipping) succeed.

Second, **Hessian information is the gold standard for sensitivity.** The theoretical framework utilizing the Hessian matrix (Frantar et al., 2022)(Shen et al., 2020) provides the most accurate predictor of quantization error. This confirms the hypothesis that the local curvature of the loss environment determines robustness.

Third, **Hardware-software co-design is essential.** The results from FPGA (Muller et al., 2024)(Chang, 2025) and specialized edge hardware (Dr.J.V.Anchitaalagammai et al., 2025) studies indicate that quantization algorithms cannot be designed in a vacuum. The most efficient implementations are those where the bit-width and arithmetic scheme match the underlying hardware capabilities (e.g., DSP slice width, vector lane size).

Fourth, **Mixed-precision is the future norm.** Whether across layers (Shen et al., 2020), components (Kim et al., 2024), or timesteps (Kim et al., 2025), the analysis shows that monolithic precision (e.g., "all INT8") is becoming obsolete. The optimal frontier lies in assigning precision proportional to information density.

Finally, the analysis of applications ranging from brain tumor segmentation (Kapo et al., 2024) to fire detection (Dilshad et al., 2023) and news classification (Risnanto & Poerwandono, 2025) demonstrates the universality of these techniques. While the specific constraints vary (e.g., safety-critical accuracy in medical imaging vs. Real-time throughput in surveillance), the fundamental principles of quantization–balancing noise against resource usage–remain constant.

The mathematical formulation of the perturbation cost:

$$\delta L \approx \Delta \mathbf{w}^T \mathbf{H} \Delta \mathbf{w}$$

serves as the unifying thread. Whether implicitly (through heuristic outlier protection) or explicitly (through Hessian-based solvers like GPTQ), all successful methods strive to minimize this quadratic form. The results reviewed in this section validate that minimizing this proxy correlates strongly with preserving downstream task performance (perplexity, accuracy, F1 score) while unlocking the efficiency of integer-only hardware.

### 2.3.8 Analysis of Quantization in Specific Application Domains

The literature review extends beyond general architectural analysis to specific application domains, providing evidence of how quantization impacts real-world tasks. The analysis of these specific implementations reveals that the tolerance for quantization noise is highly task-dependent.

#### *2.3.8.1 Medical Imaging and Diagnosis*

In the domain of medical imaging, Kapo et al. (Kapo et al., 2024) evaluate semantic segmentation of brain tumors using DeepLabV3+ and UNet with Intel's OpenVINO toolkit. The analysis of these results highlights a critical safety constraint. While quantization (INT8) provided significant speedups on edge hardware, the study necessitates a rigorous check on segmentation accuracy (Dice coefficient).

The findings suggest that for segmentation tasks, the spatial precision of the output map is sensitive to the quantization of the upsampling layers. Unlike classification, where a small perturbation in the logit might not change the argmax class, segmentation requires pixel-perfect accuracy. The analysis indicates that mixed-precision is particularly valuable here: keeping the final decoding layers in higher precision (FP16) while quantizing the heavy encoder backbone (INT8) offers the best compromise.

#### *2.3.8.2 Surveillance and Real-Time Detection*

Dilshad et al. (Dilshad et al., 2023) present an efficient framework for fire detection in surveillance environments. This application is characterized by the need for low latency (rapid detection) and deployment on resource-constrained cameras. The analysis of their results indicates that quantization is effective not just for model compression, but for reducing the thermal footprint of the device.

The study finds that lightweight models (often used in surveillance) can be more sensitive to quantization than over-parameterized models. This counter-intuitive finding–that "smaller models are harder to quantize"–aligns with the lottery ticket hypothesis. Larger models have more redundant parameters that can absorb quantization noise, whereas compact models operate closer to their information capacity limit. Therefore, the analysis suggests that applying aggressive quantization to already-efficient architectures (like MobileNet or efficient fire detection backbones) requires careful calibration (e.g., QAT or advanced PTQ) to prevent accuracy collapse.

#### *2.3.8.3 Natural Language Processing Applications*

In the field of NLP, Risnanto and Poerwandono (Risnanto & Poerwandono, 2025) analyze news topic classification for Indonesian text using ONNX Runtime. The results demonstrate the practical utility of standard quantization runtimes. The analysis shows that for classification tasks (unlike generation), the robustness to quantization is high. The decision boundaries for topic classification are apparently wide enough that the quantization noise $\Delta \mathbf{w}$ does not easily push samples across the boundary.

This contrasts with the findings in generative tasks (LLMs) discussed earlier (Dettmers et al., 2022)(Frantar et al., 2022), where the output distribution (next-token probability) is highly sensitive. This comparative analysis suggests a hierarchy of quantization difficulty: 1. **Classification:** Most strong (INT8 is standard). 2. **Segmentation/Detection:** Moderately sensitive (Spatial precision matters). 3. **Generation (LLMs/Diffusion):** Highly sensitive (Outliers and error accumulation).

### 2.3.9 Theoretical Synthesis of Error Propagation

To deepen the analysis, it is necessary to consider the propagation of quantization error through deep networks. The literature (Cai et al., 2022) discusses efficient methods for deep learning, touching upon how errors stack. In a deep network with $L$ layers, the error introduced at layer $l$, denoted as $\epsilon_l$, propagates through subsequent layers $l+1,\dots,L$.

If the network layers are Lipschitz continuous with constant $K$, the error bound at the output can theoretically grow as $K^{L-l}||\epsilon_l||$. For Transformers, the LayerNorm and Softmax operations impact this propagation. The analysis of outliers (Czakó et al., 2025) suggests that these outliers effectively increase the local Lipschitz constant of the layer, making the network more susceptible to noise amplification.

Methods like SmoothQuant (implied by the discussion of activation smoothing in (Czakó et al., 2025)) and AWQ (Lin et al., 2023) work by effectively pre-conditioning the network to reduce this sensitivity. By smoothing the activation magnitude, they lower the "sharpness" of the function being quantized, thereby reducing the impact of the error $\epsilon_l$ on the final output. This theoretical perspective explains *why* activation-aware methods outperform weight-only methods: they modify the signal flow to be more strong to the specific type of noise introduced by integer discretization.

### *2.3.9.1 The Role of Calibration Data*

The performance of PTQ methods is heavily dependent on the calibration dataset used to estimate statistics (scaling factors $s$, zero-points $z$, and Hessian $\mathbf{H}$). The analysis of GPTQ (Frantar et al., 2022) and AWQ (Lin et al., 2023) reveals that a small set of calibration data (e.g., 128 samples) is sufficient to capture the statistical properties of the activations.

However, the "representativeness" of this data is important. If the calibration set does not contain the outliers that appear during inference, the quantization parameters will be incorrect. The findings from (Dettmers et al., 2022) (LLM.int8()) emphasize that outliers are systematic and emerge at scale, meaning they are likely present even in small samples of real data. This consistency allows PTQ methods to work reliably without needing the full training set, which is a key result for the feasibility of quantizing massive private models.

### 2.3.10 Summary of Analytical Outcomes

The analysis of the selected literature establishes a cohesive narrative regarding the quantization of modern neural networks. The field has moved beyond the simple question of "how to round numbers" to a complex optimization problem involving: 1. **Spectral Analysis:** Using Hessians to identify sensitive weights (Frantar et al., 2022)(Shen et al., 2020). 2. **Distributional Analysis:** Handling heavy-tailed outliers in activations (Czakó et al., 2025)(Dettmers et al., 2022)(Lin et al., 2023). 3. **Architectural Awareness:** Treating Transformers, CNNs, and DiTs differently (Kim et al., 2024)(Kim et al., 2025). 4. **Hardware Alignment:** Matching algorithms to FPGA/Edge constraints (Muller et al., 2024)(Chang, 2025)(Dr.J.V.Anchitaalagammai et al., 2025).

The results consistently show that with the right combination of these techniques, integer-only inference (INT8, INT4) is not just a compression technique, but a viable, high-fidelity deployment strategy that enables the proliferation of AI into resource-constrained environments. The significant gap between the memory capacity of edge devices and the size of LLMs is effectively bridged by these advanced quantization methodologies, provided the "outlier problem" is rigorously addressed.

The next section will discuss the broader implications of these technical findings, placing them in the context of the operational requirements for real-world AI deployment.

## 2.4 Discussion

The synthesis of results presented in section 2.3, when viewed through the theoretical lens established in section 2.1, indicates a fundamental major change in the field of neural network compression. While early quantization frameworks focused primarily on weight rounding strategies to minimize storage, the current body of literature (Czakó et al., 2025)(Dettmers et al., 2022)(Lin et al., 2023) suggests that the primary bottleneck for deploying Large Language Models (LLMs) on integer-only hardware is not the precision of the weights, but the statistical behavior of the activations.

This section interprets these findings, comparing the empirical realities of modern quantization algorithms against the theoretical frameworks introduced earlier. It addresses the disconnect between theoretical compression rates and realized hardware speedups, analyzes the implications of activation outliers for hardware design, and evaluates the operational feasibility of deploying these quantized models in resource-constrained environments.

### 2.4.1 The Shift from Weight-Centric to Activation-Centric Quantization

As discussed in section 2.1, the traditional mathematical formulation of quantization relied on a uniform affine mapping, expressed as $q = \text{round}(r/S + Z)$. This theoretical model assumes that the underlying data distribution $r$ is relatively uniform or Gaussian, allowing a single scaling factor $S$ to effectively map the dynamic range to integers. However, the analysis of recent literature in section 2.3 reveals that this assumption fundamentally breaks down in the context of modern LLMs (e.g., Llama, OPT, GPT).

*The "Outlier" Phenomenon as a Deterministic Constraint*

The important finding synthesized from the literature is that activation outliers are not random noise but systematic features of large-scale models. Research by Dettmers et al. (Dettmers et al., 2022) on LLM.int8() demonstrates that as model scale increases (typically beyond 6.7B parameters), outlier features emerge in specific dimensions of the hidden states. These outliers are "heavy-tailed," with magnitudes significantly larger than the bulk of the data.

This empirical reality contradicts the simpler quantization models reviewed in section 2.1. If a single scaling factor $S$ is determined based on the maximum absolute value of the activation tensor (to accommodate the outlier), the resolution for the remaining 99% of the values–which cluster near zero–is effectively destroyed. This phenomenon explains the severe degradation in perplexity observed in naive INT8 quantization approaches.

The literature offers two distinct divergent solutions to this problem, as summarized in Table 1 below.

| Approach | Mechanism | Key Literature | Hardware Implication |
|---|---|---|---|
| **Decomposition** | Separates outliers (FP16) from bulk (INT8) | LLM.int8() (Dettmers et al., 2022) | High latency due to mixed-precision kernels |
| **Smoothing** | Migrates quantization difficulty from activations to weights | AWQ (Lin et al., 2023) | Efficient inference; offline complexity |
| **Hessian Opt.** | Uses second-order info to protect sensitive weights | GPTQ (Frantar et al., 2022), Q-BERT (Shen et al., 2020) | Computationally expensive calibration |
| **Hybrid** | Layer-specific policies for different architectures | HyQ (Kim et al., 2024), MixDiT (Kim et al., 2025) | Requires specialized compiler support |

*Table 1: Comparison of outlier mitigation strategies in modern quantization frameworks. Source: Adapted from (Dettmers et al., 2022), (Lin et al., 2023), and (Frantar et al., 2022).*

The "Activation-aware Weight Quantization" (AWQ) method (Lin et al., 2023) represents a significant theoretical advancement over the methods discussed in section 2.1. Rather than treating weights and activations as independent quantization problems, AWQ recognizes that the quantization error of weights should be weighted by the magnitude of the activations they multiply. By scaling up the weights associated with salient activation channels (and scaling down the activations correspondingly to preserve the mathematical equivalence), the quantization noise is effectively pushed into the weights, which are static and easier to handle.

This finding validates the hypothesis that "representativeness" of calibration data, discussed in section 2.3, is less about capturing the exact distribution of user data and more about identifying these structural outliers. Since outliers are systematic (Dettmers et al., 2022), a small calibration set (e.g., 128 samples) is sufficient to compute the necessary scaling factors, provided the set triggers these emergent features.

*Theoretical Implications for the Optimization Objective*

The findings from section 2.3 suggest that the optimization objective for quantization has evolved. Section 2.1 introduced the minimization of the Frobenius norm of the weight error:

$$\min_{Q} \|W - Q(W)\|_F^2$$

However, the efficacy of methods like GPTQ (Frantar et al., 2022) and Q-BERT (Shen et al., 2020) indicates that this objective is insufficient. The literature demonstrates that a more accurate objective must account for the curvature of the loss environment, typically approximated by the Hessian $H$. The refined objective becomes:

$$\min_{Q} (W - Q(W))^T H (W - Q(W))$$

This shift acknowledges that not all weights are created equal; weights corresponding to high-curvature directions in the loss environment (often correlated with activation outliers) require higher precision or careful rounding to maintain model fidelity.

### 2.4.2 The Hardware-Software Gap: Theoretical vs. Realized Gains

A significant research gap identified in section 2.1 was the discrepancy between theoretical compression rates (e.g., 4x reduction from FP32 to INT8) and actual wall-clock speedups. The discussion of results in section 2.3, particularly regarding hardware-specific implementations (Muller et al., 2024)(Chang, 2025)(Martínez et al., 2025), clarifies the nature of this gap.

*The Memory Bandwidth vs. Compute Bound Trade-off*

While quantization effectively addresses the memory capacity bottleneck–allowing, for instance, a 7B parameter model to fit on a consumer GPU or Edge TPU (Dr.J.V.Anchitaalagammai et al., 2025)–it introduces computational overhead that can negate latency gains.

Research on FPGA co-design (Muller et al., 2024)(Chang, 2025) highlights that the non-uniform quantization schemes required to handle outliers (such as the sparse decomposition in LLM.int8()) create irregular memory access patterns. Standard systolic arrays, designed for dense matrix multiplications, struggle with these irregularities. As noted by Wang et al. (Wang et al., 2025), conventional GPGPUs face challenges in resource utilization with irregular matrices.

Furthermore, the “packing” and “unpacking” of sub-byte integers (e.g., INT4) often requires runtime overhead. Unless the hardware supports native INT4 instructions, the processor must spend cycles unpacking data into INT8 or FP16 registers for computation. This aligns with the findings of Martínez et al. (Martínez et al., 2025), who observed that on ARM and RISC-V CPUs, the theoretical throughput gains of quantization are often bottlenecked by the instruction set architecture’s lack of support for low-precision vector operations.

*Implications for Edge AI and FPGA Implementations*

The literature indicates that Field-Programmable Gate Arrays (FPGAs) offer a potential solution to this hardware misalignment. Unlike fixed-architecture GPUs, FPGAs can be reconfigured to implement custom bit-width arithmetic, such as the "Scalar Arithmetic Multiple Data" approach described by Anderson et al. (Anderson et al., 2019). This allows for the implementation of the precise quantization schemes developed in the theoretical literature without the "packing penalty" incurred on general-purpose processors.

Recent work on the "TinyLLM" co-design (Muller et al., 2024) and hardware-software co-design for PCIe-based FPGAs (Chang, 2025) demonstrates that when the hardware logic is tailored to the specific quantization outlier patterns, the theoretical efficiency gains can be fully realized. This supports the argument that future progress in this field requires a comprehensive "co-design" approach (Dua & Patel, 2024), rather than treating quantization algorithm development and hardware accelerator design as separate disciplines.

### 2.4.3 Architectural Nuances: Beyond Standard Transformers

The results analyzed in section 2.3 extend beyond standard text-based LLMs to include hybrid and generative architectures, addressing the need for domain-specific quantization strategies.

*Hybrid Architectures (CNN-Transformer)*

As discussed in section 2.1, different neural network layers exhibit different sensitivities to quantization noise. The findings regarding HyQ (Kim et al., 2024) reinforce this, showing that in hybrid CNN-Transformer models (often used in vision tasks), the convolutional layers and self-attention layers require distinct quantization policies. Convolutional layers, which typically process local features, are often more strong to aggressive quantization than the global attention mechanisms in Transformers. This suggests that a monolithic quantization strategy (e.g., "quantize everything to INT8") is suboptimal.

*Generative and Diffusion Models*

The analysis of MixDiT (Kim et al., 2025) and FPGA-accelerated DCGANs (Sadr et al., 2025) highlights unique challenges in generative models. Unlike discriminative models (classification), where the output is a probability distribution, generative models output high-dimensional data (images, text) where subtle quantization artifacts can cascade into significant quality degradation.

The literature suggests that "Mixed-Precision MX Quantization" (Kim et al., 2025) is essential for these architectures. This involves keeping sensitive layers (such as the initial embedding or final projection layers) in higher precision (FP16) while aggressively quantizing the heavy compute layers (MLP blocks) to INT8 or INT4. This aligns with the "sensitivity analysis" approach advocated in Q-BERT (Shen et al., 2020), where Hessian-based metrics determine the bit-width of each layer.

### 2.4.4 Operational Implications for Resource-Constrained Environments

The ultimate goal of quantization, as framed in the introduction (section 1) and literature review (section 2.1), is to enable "Edge AI"–the deployment of sophisticated models on devices with limited power and thermal budgets. The findings synthesized in section 2.3 provide strong evidence that this is becoming operationally feasible, provided specific constraints are met.

*Privacy and Latency in Critical Applications*

The deployment of quantized models on edge devices like the NVIDIA Jetson Orin or Google Coral Edge TPU (Dr.J.V.Anchitaalagammai et al., 2025) has profound implications for privacy. In medical applications, such as the semantic segmentation of brain tumors discussed by Kapo et al. (Kapo et al., 2024), transferring patient data to the cloud for inference is often prohibited by privacy regulations. Quantization enables these high-performance segmentation models (e.g., DeepLabV3+, UNet) to run locally on the medical device.

Similarly, in telecommunications and 6G networks (Dr.J.V.Anchitaalagammai et al., 2025)(Zhu et al., 2025), the latency requirements for real-time decision-making preclude cloud offloading. The literature (Zhu et al., 2025) discusses "fronthaul quantization," optimizing the bit-allocation not just for model weights, but for the data transmission itself in cell-free systems. This illustrates that quantization principles are transferable from model compression to bandwidth compression.

*Energy Efficiency and Sustainability*

The energy footprint of AI is a growing concern. The literature regarding hardware optimization for generative AI (Dua & Patel, 2024) and search-optimized quantization (Bouaggad & Grabar, 2025) emphasizes that reducing precision correlates linearly (or sometimes quadratically) with energy savings. An INT8 MAC (Multiply-Accumulate) operation consumes significantly less energy than an FP32 MAC. For battery-powered devices, this efficiency is the difference between a viable product and a theoretical prototype.

Table 2 summarizes the operational implications derived from the cited literature.

| Domain | Key Requirement | Role of Quantization | Reference |
|---|---|---|---|
| **Healthcare** | Data Privacy | Enables on-device segmentation (no cloud) | (Kapo et al., 2024) |
| **Telecom (6G)** | Ultra-low Latency | Reduces fronthaul bandwidth load | (Zhu et al., 2025) |

| Domain | Key Requirement | Role of Quantization | Reference |
|---|---|---|---|
| **Surveillance** | Real-time Processing | Fits fire detection models on edge cameras | (Dilshad et al., 2023) |
| **Mobile NLP** | Storage Constraints | Allows multi-label classification on phones | (Risnanto & Poerwandono, 2025) |
| **Ontology** | Computational Cost | Reduces search space for alignment | (Bouaggad & Grabar, 2025) |

*Table 2: Operational impact of quantization across diverse application domains. Source: Synthesized from cited works.*

### 2.4.5 Limitations and Future Directions

While the literature presents a cohesive narrative of progress, several limitations and unresolved challenges remain.

*The Calibration Dependency*

Despite the finding that outliers are systematic (Dettmers et al., 2022), current Post-Training Quantization (PTQ) methods still rely on a “calibration set” to estimate statistical parameters. If the target domain differs significantly from the calibration data (distributional shift), the quantization parameters ($S$ and $Z$) may be suboptimal. While approaches like ExNAS (RODRIGUEZ, 2025) propose “experiential” architectures that adapt, the static nature of most quantization schemes remains a limitation for dynamic environments.

*Hardware Fragmentation*

A recurring theme in the discussion of hardware (Chang, 2025)(Akita et al., 2025)(Martínez et al., 2025) is fragmentation. There is no standard format for "INT4" or "INT3" across devices. NVIDIA's Tensor Cores, ARM's NEON/SVE, and Xilinx's DPU all implement low-precision arithmetic differently. This forces researchers to optimize for specific hardware targets (as seen in the specific focus on Jetson/Coral in (Dr.J.V.Anchitaalagammai et al., 2025) or specific FPGAs in (Sadr et al., 2025)), hindering the "write once, deploy anywhere" ideal.

*Evaluation Metrics*

Finally, there is a tension in the literature regarding evaluation. Most studies uses perplexity (for LLMs) or mIoU (for segmentation (Kapo et al., 2024)) as the primary metric. However, as noted in the analysis of biomedical ontology alignment (Bouaggad & Grabar, 2025), task-specific metrics are important. A quantized model might maintain low perplexity but fail in subtle reasoning tasks or generate hallucinations at a higher rate than the full-precision model–a phenomenon that requires further longitudinal study.

### 2.4.6 Synthesis

In conclusion, the discussion of the literature indicates that the field of LLM quantization has matured from a simple data compression problem into a complex optimization challenge involving statistical analysis of activation outliers, Hessian-based sensitivity analysis, and hardware-software co-design.

The theoretical framework established in section 2.1 provides the necessary mathematical foundation, but the empirical results discussed in section 2.3 demonstrate that real-world implementation requires deviations from the ideal affine mapping. Specifically, the "outlier problem" identified by Dettmers et al. (Dettmers et al., 2022) and addressed by

Lin et al. (Lin et al., 2023) serves as the fulcrum upon which modern quantization strategies balance.

The successful bridging of the gap between massive model sizes and edge device constraints–validated by successful deployments in medical (Kapo et al., 2024) and industrial (Dilshad et al., 2023) contexts–confirms that integer-only inference is a viable path forward. However, the future of this field lies not just in better rounding algorithms, but in the "co-design" (Muller et al., 2024)(Dua & Patel, 2024) of neural architectures and hardware accelerators that treat low-precision, outlier-heavy computation as a first-class citizen.

# 3. Conclusion

The comprehensive examination of large language model (LLM) quantization for integer-only hardware reveals a critical inflection point in the trajectory of artificial intelligence deployment. This thesis has explored the widening chasm between the exponential growth of model parameters–exemplified by architectures such as Llama and OPT–and the physical constraints of deployment hardware. The research presented confirms that quantization is not merely an optional optimization technique but a fundamental requisite for the democratization of advanced AI. By systematically analyzing the transition from high-precision floating-point arithmetic (FP32, FP16) to low-precision integer formats (INT8, INT4), this study demonstrates that near-lossless performance is achievable through sophisticated post-training quantization (PTQ) methodologies, provided that activation outliers and hardware-specific constraints are rigorously addressed.

## 3.1 Synthesis of Methodological Findings

The investigation into theoretical frameworks and algorithmic innovations highlights that the primary challenge in quantization is no longer the weight distribution alone, but the complex interplay between weights and activations.

### *3.1.1 The Evolution of Post-Training Quantization*

The literature consistently demonstrates that traditional uniform quantization methods fail to preserve the representational capacity of modern LLMs due to the emergence of extreme outliers in activation channels. As detailed in recent systematic reviews, addressing these activation outliers is essential for maintaining model fidelity (Czakó et al., 2025). The analysis confirms that advanced PTQ techniques, such as GPTQ (Frantar et al., 2022) and LLM.int8() (Dettmers et al., 2022), successfully mitigate quantization error by treating salient weights and outliers with higher precision or specific correction mechanisms while compressing the vast majority of parameters into 8-bit or 4-bit integers.

Specifically, the effectiveness of activation-aware weight quantization (AWQ) underscores the necessity of protecting the most critical 1% of parameters to preserve the performance of the remaining 99% (Lin et al., 2023). This finding represents a major change from "model-agnostic" quantization to "sensitivity-aware" approaches. The evidence suggests that preserving the accuracy of generative models does not require high precision everywhere, but rather high precision *where it matters*. This aligns with findings regarding the quantization of Transformer-based architectures like BERT, where Hessian-based approaches have been utilized to identify and protect sensitive layers (Shen et al., 2020).

### *3.1.2 Hardware-Software Co-Design*

A recurring theme throughout this research is the inextricable link between algorithmic compression and hardware architecture. Algorithms cannot be designed in a vacuum; their efficacy is determined by the underlying silicon. The analysis of hardware-friendly quantization methods, such as HyQ for hybrid networks, illustrates that optimization must account for the specific instruction sets and memory hierarchies of the target device (Kim et al., 2024).

Table 3.1 summarizes the key relationships between quantization algorithms and their corresponding hardware implications as identified in the reviewed literature.

| Quantization Approach | Key Algorithmic Mechanism | Hardware Implication | Source |
|---|---|---|---|
| Vector-wise PTQ | Row/Column independent scaling | Requires specialized kernels for reduction | (Frantar et al., 2022) |
| Mixed-Precision (LLM.int8) | Outlier separation (FP16) + INT8 | Dual-path execution units required | (Dettmers et al., 2022) |

| Quantization Approach | Key Algorithmic Mechanism | Hardware Implication | Source |
|---|---|---|---|
| Activation-Aware (AWQ) | Salient weight scaling | Standard INT arithmetic compatibility | (Lin et al., 2023) |
| Hessian-Based | Sensitivity-based bit allocation | Complex offline calibration | (Shen et al., 2020) |
| Hardware-Aware (HyQ) | Layer-specific quantization policies | Optimized for NPU/DSP constraints | (Kim et al., 2024) |

*Table 3.1: Synthesis of Quantization Approaches and Hardware Dependencies.*

The distinction between theoretical compression rates and actual inference acceleration is important. While vector-wise quantization offers superior perplexity scores, it imposes overhead on hardware that lacks native support for granular scaling factors. Conversely, methods like AWQ demonstrate superior practical deployment potential by maintaining compatibility with standard integer arithmetic units found in commodity hardware (Lin et al., 2023).

## 3.2 Implications for Hardware Architecture

The shift toward integer-only inference has profound implications for the design of next-generation processors, particularly in edge computing environments where power and thermal budgets are strictly capped.

### *3.2.1 The Rise of Edge AI and Custom Silicon*

The research highlights a migration of intelligence from centralized data centers to the edge. This transition is enabled by hardware architectures specifically optimized for low-precision arithmetic. Recent advancements in FPGA technology demonstrate the viability of hardware-software co-design for running LLMs on programmable logic (Muller et al., 2024). By utilizing PCIe-based FPGA accelerators with coarse-grained systolic arrays, researchers have achieved significant efficiency gains over general-purpose GPUs for specific inference workloads (Chang, 2025).

Furthermore, the implementation of configurable systolic arrays for GPGPUs addresses the resource utilization challenges inherent in processing the irregular matrices often produced by sparse quantization methods (Wang et al., 2025). This suggests a future where hardware is increasingly specialized; generic compute units are being supplemented or replaced by tensor-optimized cores capable of executing massive INT8 and INT4 matrix multiplications with high energy efficiency.

### *3.2.2 RISC-V and ARM Integration*

The democratization of LLMs is further supported by optimizations for ubiquitous processor architectures. The ability to run latency-critical quantized inference on ARM and RISC-V CPUs expands the reach of generative AI to billions of mobile and IoT devices (Martínez et al., 2025). This capability is essential for applications requiring privacy and offline functionality, such as medical diagnostics or real-time translation, where transmitting data to a cloud server is impractical or insecure. The integration of efficient deep learning frameworks on these architectures allows for complex tasks, such as fire detection in surveillance environments (Dilshad et al., 2023) or brain tumor segmentation (Kapo et al., 2024), to be performed locally with reduced latency.

## 3.3 Broader Impacts and Applications

The technical achievements in quantization translate directly into tangible benefits across various domains. The reduction in memory footprint–often by 50% or more–allows models that previously required server-grade hardware to run on consumer devices.

### *3.3.1 Democratization of Advanced NLP*

The ability to quantize models like Llama for improved efficiency (Madhanegha et al., 2025) lowers the barrier to entry for researchers and developers. This democratization fosters innovation in language-specific applications, such as the classification of news topics in under-represented languages like Indonesian (Risnanto & Poerwandono, 2025). By reducing the hardware requirements, quantization enables the deployment of sophisticated NLP models in regions or institutions with limited access to high-performance computing clusters.

### *3.3.2 Medical and Scientific Advancements*

In the medical field, the reliability of quantized models is of essential importance. The literature indicates that with careful calibration, quantized networks can maintain high performance in sensitive tasks like semantic segmentation of medical imagery (Kapo et al., 2024). Furthermore, search-optimized quantization techniques are enabling more efficient alignment of biomedical ontologies, facilitating better data interoperability in healthcare systems (Bouaggad & Grabar, 2025). These applications demonstrate that integer-only inference is strong enough for critical decision-making processes, provided that the quantization scheme is rigorously validated.

### *3.3.3 Sustainability and Energy Efficiency*

Beyond performance, quantization addresses the growing concern regarding the environmental impact of AI. Hardware optimization for generative AI is critical for sustainability (Dua & Patel, 2024). By reducing the precision of operations, the energy cost per inference drops significantly. This is vital for "always-on" systems and large-scale deployments where the cumulative energy consumption of billions of inferences becomes a substantial ecological burden. The use of platforms like NVIDIA Jetson Orin and Google Coral Edge TPU highlights the industry's focus on maximizing performance per watt through low-precision computing (Dr.J.V.Anchitaalagammai et al., 2025).

## 3.4 Limitations and Challenges

Despite the significant progress detailed in this thesis, several limitations remain that prevent the universal adoption of integer-only quantization for all LLM workloads.

### *3.4.1 The Accuracy-Efficiency Trade-off*

While 8-bit quantization is largely considered "solved" for many architectures, pushing the limits to 4-bit or 2-bit precision often results in non-negligible degradation of model performance. As noted in the analysis of GPTQ, while accuracy remains high for standard benchmarks, the "outlier problem" persists as a barrier to ultra-low precision (Frantar et al., 2022). There is a fundamental information theoretic limit to how much a model can be compressed before its reasoning capabilities–particularly in complex, multi-step tasks–are compromised.

### *3.4.2 Hardware Fragmentation*

The diversity of hardware accelerators presents a challenge for standardization. A quantization scheme optimized for a specific FPGA implementation (Sadr et al., 2025) may not translate effectively to a RISC-V CPU (Martínez et al., 2025) or a custom systolic array (Wang et al., 2025). This fragmentation forces developers to maintain multiple quantized versions of the same model or rely on complex compilation stacks to map abstract model definitions to specific hardware backends.

### *3.4.3 Complexity of Calibration*

Post-training quantization often requires a calibration dataset to determine optimal scaling factors. The representativeness of this calibration data is important; if the data distribution shifts during deployment, the static integer ranges defined during quantization may lead to overflow or underflow, degrading performance. This is particularly challenging for dynamic cross-layer memory systems designed for real-time inference optimization (RODRIGUEZ, 2025), where the model must adapt to varying input contexts.

## 3.5 Future Directions

The field of LLM quantization is evolving rapidly. Based on the trajectories identified in the literature, several key areas for future research and development are evident.

### *3.5.1 Advancements in Mixed-Precision Architectures*

The future of efficient inference likely lies not in uniform integer formats, but in dynamic mixed-precision approaches. Techniques that can accelerate image diffusion transformers via mixed-precision MX quantization (Kim et al., 2025) suggest a path forward where different layers or even different attention heads uses varying bit-widths based on their real-time sensitivity. Hardware that supports flexible precision, such as scalar arithmetic multiple data architectures (Anderson et al., 2019), will be instrumental in realizing this vision.

### *3.5.2 Integration of Sensing and Communication*

As AI models move to the edge, they are increasingly integrated into complex systems that combine sensing, communication, and computation. Research into cell-free integrated sensing and communication (ISAC) systems highlights the need for fronthaul quantization bits allocation (Zhu et al., 2025). Future quantization strategies must consider the entire pipeline–from the sensor data acquisition to the final model output–optimizing bandwidth and compute simultaneously.

### *3.5.3 Automated Quantization Pipelines*

To address hardware fragmentation, automated tools that can profile a model and the target hardware to select the optimal quantization strategy will become standard. Concepts like “Experiential Neural Architecture Selection” (ExNAS) (RODRIGUEZ, 2025) point toward systems that can dynamically adjust their structure and precision. Similarly, the continued development of efficient methods for deep learning (Cai et al., 2022) will likely yield “push-button” solutions that abstract the complexities of quantization from the model developer.

## 3.6 Final Remarks

This thesis concludes that the quantization of Large Language Models for integer-only hardware represents a mature yet evolving discipline that has successfully bridged the gap between theoretical capability and practical deployability. The transition from floating-point to integer arithmetic is not a compromise but an evolution, enabled by rigorous mathematical frameworks and innovative hardware designs.

The evidence synthesized from the literature confirms that techniques like GPTQ, AWQ, and hardware-software co-design have effectively solved the primary hurdles of 8-bit deployment and are making significant strides in 4-bit inference. As hardware architectures continue to specialize for low-precision operations–evident in the latest solid-state circuit developments (Akita et al., 2025) and graph neural network accelerators (Auten et al., 2020)–the reliance on heavy, energy-inefficient floating-point hardware will diminish for inference tasks.

Ultimately, the successful implementation of integer-only LLMs fulfills the promise of ubiquitous AI. By reducing the computational, memory, and energy barriers, quantization empowers a future where advanced intelligence is embedded in the fabric of daily technology, from life-saving medical devices to efficient global communication networks. The path forward involves a continued synergy between algorithmic researchers and hardware architects, ensuring that as models grow in intelligence, they remain grounded in the physical realities of efficient computing.

# 4. Appendices

## A.1 Mathematical Foundations of Integer Mapping

The transition from floating-point representation to integer-only arithmetic necessitates a rigorous mathematical framework to ensure signal fidelity. As established in the main body, the fundamental operation involves mapping a real-valued tensor $X_f$ to an integer tensor $X_q$. This appendix details the specific mechanisms of affine quantization, symmetric versus asymmetric mapping, and granularity strategies that define modern post-training quantization (PTQ) approaches.

### *A.1.1 Affine Quantization Schemes*

The standard affine quantization scheme, often referred to as uniform quantization, is defined by two primary parameters: the scaling factor ($S$) and the zero-point ($Z$). These parameters determine how the dynamic range of the floating-point values is compressed into the discrete integer domain.

The quantization function $Q(x)$ and dequantization function $D(x_q)$ are formally expressed as:

$$Q(x) = \text{clamp}\left(\lfloor \frac{x}{S} \rceil + Z, q_{min}, q_{max}\right)$$

$$D(x_q) = S(x_q - Z)$$

Here, $\lfloor \cdot \rceil$ denotes the rounding-to-nearest operation. The clamping function restricts the values to the representable range of the target bit-width $b$, where typically $q_{min} = -2^{b-1}$ and $q_{max} = 2^{b-1} - 1$ for signed integers.

| Scheme | Definition of Scale ($S$) | Definition of Zero-Point ($Z$) | Primary Use Case |
|---|---|---|---|
| Symmetric | $S = \frac{\max(\lvert x_{min}\rvert, \lvert x_{max}\rvert)}{q_{max}}$ | $Z = 0$ (Fixed) | Weight Quantization |
| Asymmetric | $S = \frac{x_{max} - x_{min}}{q_{max} - q_{min}}$ | $Z = \lfloor q_{min} - \frac{x_{min}}{S} \rceil$ | Activation Quantization |
| Logarithmic | $S = 2^k$ (Power of 2) | $Z = 0$ | Hardware Efficiency |

*Table A.1: Comparison of quantization mapping schemes. Source: Adapted from synthesis of (Cai et al., 2022) and (Martínez et al., 2025).*

Symmetric quantization is generally preferred for weights because the distribution of weights in neural networks, including Transformers, tends to be Gaussian-like and centered around zero. Enforcing $Z = 0$ reduces the computational overhead during matrix multiplication, as the cross-terms involving the zero-point vanish. This efficiency is critical for hardware accelerators utilizing systolic arrays, as discussed by Wang et al. (Wang et al., 2025) and Chang (Chang, 2025).

Conversely, activations often exhibit skewed distributions, particularly after ReLU or GeLU functions where values are non-negative. In such cases, asymmetric quantization is necessary to fully uses the available bit depth. However, this introduces additional computational complexity during inference, as the zero-point must be calculated and subtracted during the accumulation phase.

### *A.1.2 Granularity of Quantization Parameters*

A critical factor in preserving the accuracy of Large Language Models (LLMs) is the granularity at which the scaling factors $S$ and zero-points $Z$ are calculated. The choice of granularity represents a trade-off between memory compression and model accuracy.

**Tensor-wise Quantization:** This approach calculates a single $S$ and $Z$ for the entire tensor. While this offers the maximum compression ratio, it is often insufficient for LLMs due to the high dynamic range variance across different channels or tokens.

**Channel-wise Quantization:** Commonly applied to weights, this method assigns a distinct scale factor to each output channel of a weight matrix. This accommodates the varying magnitudes of filters within the network.

**Token-wise and Group-wise Quantization:** For activations in Transformers, outliers often appear in specific tokens or feature dimensions. Dettmers et al. (Dettmers et al., 2022) demonstrated that outlier features in LLMs (e.g., GPT-3, OPT) are systematic and can be handled by vector-wise quantization or mixed-precision decomposition. Similarly, group-wise quantization divides tensors into smaller blocks (e.g., 128 elements) to localize the scaling parameters, reducing the impact of outliers on the global distribution.

## A.2 Advanced Optimization Objectives

Beyond simple rounding, advanced PTQ methods uses optimization algorithms to minimize the information loss incurred by quantization.

### *A.2.1 Hessian-Based Optimization*

Simple rounding to the nearest integer is not always optimal for minimizing the task loss. Methods like GPTQ (Frantar et al., 2022) and Q-BERT (Shen et al., 2020) uses second-order information from the Hessian matrix (or its approximation) to determine the optimal quantized weights. The objective function seeks to minimize the squared error between the output of the full-precision layer ($WX$) and the quantized layer ($\hat{W}X$), weighted by the Hessian $H$:

$$\min_{\hat{W}} ||WX - \hat{W}X||_2^2 \approx \min_{\hat{W}} (\hat{w} - w)^T H(\hat{w} - w)$$

This approach acknowledges that not all weights contribute equally to the output error. Weights corresponding to high-curvature directions in the loss environment (large eigenvalues in the Hessian) must be quantized more carefully than those in flat regions.

### *A.2.2 Activation-Aware Scaling*

Recent developments, such as AWQ (Lin et al., 2023), shift the focus from minimizing weight error to preserving activation distribution. The core insight is that the quantization error of weights should be weighted by the magnitude of the activations they multiply. By protecting "salient" weights–those corresponding to large activation magnitudes–performance can be significantly retained without requiring gradient updates or retraining.

# Appendix B: Supplementary Data Tables and Hardware Specifications

## B.1 Theoretical Resource Requirements for LLM Inference

The deployment of LLMs on edge devices is fundamentally constrained by memory bandwidth and capacity. The following tables provide a theoretical analysis of resource requirements for common open-source LLM architectures when subjected to different quantization precisions. These calculations are derived from the architectural specifications discussed in (Frantar et al., 2022) and (Madhanegha et al., 2025).

| Model Class | Parameters | Precision | Model Size (GB) | Min. Bandwidth for 20 tok/s |
|---|---|---|---|---|
| Llama-7B | 7 Billion | FP16 (16-bit) | ~14.0 GB | 280 GB/s |
| Llama-7B | 7 Billion | INT8 (8-bit) | ~7.0 GB | 140 GB/s |
| Llama-7B | 7 Billion | INT4 (4-bit) | ~3.5 GB | 70 GB/s |
| Llama-13B | 13 Billion | FP16 (16-bit) | ~26.0 GB | 520 GB/s |

| Model Class | Parameters | Precision | Model Size (GB) | Min. Bandwidth for 20 tok/s |
|---|---|---|---|---|
| Llama-13B | 13 Billion | INT4 (4-bit) | ~6.5 GB | 130 GB/s |

*Table B.1: Theoretical memory footprint and bandwidth requirements. "Model Size" includes parameters only; actual runtime memory requires additional buffer for KV-cache and activations.*

As illustrated in Table B.1, 4-bit quantization (INT4) is a critical enabler for deploying 7B and 13B parameter models on consumer-grade or edge hardware. For instance, a Llama-7B model in FP16 requires approximately 14 GB of VRAM, exceeding the capacity of many standard GPUs and almost all embedded devices. In contrast, the INT4 representation reduces this to 3.5 GB, fitting comfortably within the memory constraints of devices like the NVIDIA Jetson Orin or high-end mobile SoCs (Dr.J.V.Anchitaalagammai et al., 2025). The bandwidth requirement column highlights the "memory wall" problem: to achieve a conversational token generation rate (e.g., 20 tokens/second), the memory bandwidth requirements scale linearly with bit precision.

## B.2 Hardware Accelerator Specifications for Edge AI

To contextualize the feasibility of integer-only inference, it is necessary to examine the specifications of current hardware accelerators targeted for edge deployment. The shift toward Edge AI (Dr.J.V.Anchitaalagammai et al., 2025) relies on these platforms supporting low-precision arithmetic (INT8/INT4).

| Platform | Architecture | AI Performance (TOPS) | Precision Support | Power |
|---|---|---|---|---|
| NVIDIA Jetson Orin | Ampere GPU | Up to 275 | FP16, INT8 | 15-60W |
| Google Coral TPU | ASIC | 4 | INT8 (Fixed) | 2W |
| Xilinx Versal AI | FPGA/ACAP | Scalable | INT8, INT4, Custom | Varied |
| ARM Cortex-A (NPU) | CPU/NPU | 10-50 | INT8, BF16 | <5W |

*Table B.2: Comparative specifications of Edge AI hardware accelerators. Source: Compiled from (Dr.J.V.Anchitaalagammai et al., 2025), (Sadr et al., 2025), and (Martínez et al., 2025).*

Table B.2 highlights the hardware diversity in the edge system. While GPU-based solutions like the Jetson Orin offer flexibility with mixed precision (FP16/INT8), dedicated ASICs like the Google Coral TPU are strictly limited to INT8 operations. This rigid constraint necessitates strong quantization strategies that do not rely on runtime floating-point scalars. Furthermore, FPGA solutions (Muller et al., 2024)(Chang, 2025) offer the unique capability to implement custom bit-widths (e.g., INT3 or INT2) or non-standard formats, allowing for co-design of the hardware and the quantized model. The support for INT4 is becoming increasingly standard in newer NPU architectures, driven by the specific needs of Generative AI workloads (Dua & Patel, 2024).

## B.3 Comparison of Post-Training Quantization Methodologies

The following table synthesizes the key characteristics of prominent PTQ algorithms discussed in the literature review.

| Method | Target Precision | Outlier Handling | Calibration Cost | Reference |
|---|---|---|---|---|
| LLM.int8() | INT8 | Mixed-precision decomposition | Low (Inference-time) | (Dettmers et al., 2022) |
| GPTQ | INT4/3 | Hessian-based error min. | Moderate (One-shot) | (Frantar et al., 2022) |
| AWQ | INT4/3 | Activation-aware scaling | Low (Search-based) | (Lin et al., 2023) |
| Q-BERT | Mixed | Hessian spectrum analysis | High (Group-wise) | (Shen et al., 2020) |

| Method | Target Precision | Outlier Handling | Calibration Cost | Reference |
|---|---|---|---|---|
| SmoothQuant | INT8 | Math. Equivalent scaling | Low (Offline) | (MIT, 2026) |

*Table B.3: Overview of quantization algorithms. Note: "Calibration Cost" refers to the computational overhead required to determine quantization parameters.*

LLM.int8() (Dettmers et al., 2022) is unique in its runtime approach to outliers, separating them into a 16-bit stream while quantizing the bulk of the vector to 8-bit. This preserves accuracy but incurs a latency penalty due to kernel launch overheads. In contrast, GPTQ (Frantar et al., 2022) and AWQ (Lin et al., 2023) focus on optimizing the weight representation offline, allowing for efficient integer-only kernels during runtime. The approach by Q-BERT (Shen et al., 2020) demonstrates the utility of Hessian information even in smaller Transformer models, a concept that GPTQ scaled to billions of parameters.

# Appendix C: Glossary of Terms

**Activation Outliers** Feature dimensions in neural network layers that exhibit magnitudes significantly larger (often 100x) than the surrounding values. In Transformer models, these outliers are systematic and critical for performance. If truncated during quantization, model accuracy degrades largely. Techniques like LLM.int8() (Dettmers et al., 2022) and SmoothQuant (MIT, 2026) specifically address this phenomenon.

**Affine Quantization** A mapping scheme where real values are approximated by integers using a linear transformation defined by a scale factor ($S$) and a zero-point ($Z$). This is the standard arithmetic used in INT8 inference on most hardware accelerators.

**Calibration** The process of determining the optimal quantization parameters (scale and zero-point) for activations. This typically involves passing a small set of representative data (calibration set) through the model to observe the dynamic range of activation tensors.

**Edge AI** The deployment of artificial intelligence algorithms on local devices (e.g., smartphones, IoT sensors, embedded systems) rather than centralized cloud servers. This paradigm reduces latency, bandwidth usage, and privacy risks (Dr.J.V.Anchitaalagammai et al., 2025).

**FPGA (Field-Programmable Gate Array)** An integrated circuit designed to be configured by a customer or a designer after manufacturing. FPGAs are increasingly used for LLM inference due to their ability to support custom precision arithmetic and variable bit-widths (Muller et al., 2024)(Sadr et al., 2025).

**Hessian Matrix** A square matrix of second-order partial derivatives of a scalar-valued function. In the context of quantization (e.g., GPTQ (Frantar et al., 2022), Q-BERT (Shen et al., 2020)), the Hessian of the loss function with respect to the weights indicates the sensitivity of the model to perturbations (errors) in those weights.

**Inference** The phase where a trained model is used to make predictions or generate text based on new input data. Quantization primarily targets the optimization of the inference phase to reduce cost and latency.

**Mixed-Precision Inference** A technique where different parts of a model or different operations are computed at varying precisions. For example, storing weights in INT4 while performing accumulation in FP16, or dealing with outliers in FP16 while the rest of the matrix multiplication occurs in INT8 (Dettmers et al., 2022)(Kim et al., 2025).

**Post-Training Quantization (PTQ)** A quantization technique applied after the model has been fully trained, requiring little to no retraining. PTQ is preferred for LLMs due to the prohibitive cost of retraining massive models. Methods like GPTQ (Frantar et al., 2022) and AWQ (Lin et al., 2023) are examples of PTQ.

**Quantization-Aware Training (QAT)** A method where quantization errors are simulated during the training process (forward pass), allowing the model to learn parameters that are strong to quantization. While often yielding higher accuracy than PTQ, it is computationally expensive for LLMs.

**Systolic Array** A homogeneous network of tightly coupled data processing units (DPUs). Each unit computes a partial result and passes it to a neighbor. This architecture is highly efficient for matrix multiplications and is the basis for Google's TPU and many FPGA accelerators (Wang et al., 2025)(Chang, 2025).

**Zero-Point** An integer value in the quantized domain that corresponds to the real value zero. It allows the quantization scheme to represent asymmetric ranges (e.g., outputs of ReLU activations).

# Appendix D: Additional Resources and Tools

This appendix curates a list of essential software frameworks, libraries, and hardware development kits referenced throughout the thesis. These resources facilitate the implementation of quantized LLMs and their deployment on integer-only hardware.

## D.1 Quantization Libraries and Frameworks

**GPTQ (Generative Pre-trained Transformer Quantization)** *Description:* A current algorithm for compressing LLMs to 3 or 4 bits with negligible accuracy loss. The official implementations uses Hessian-based information to update weights in a layer-wise manner. *Key Reference:* Frantar et al. (Frantar et al., 2022) *Relevance:* Essential for deploying 175B+ parameter models on single GPUs.

**LLM.int8() / bitsandbytes** *Description:* A library enabling 8-bit matrix multiplication for Transformers. It introduces a mixed-precision decomposition to handle activation outliers, allowing 8-bit inference with performance matching 16-bit baselines. *Key Reference:* Dettmers et al. (Dettmers et al., 2022) *Relevance:* Widely used in the Hugging Face system for accessible LLM inference.

**AWQ (Activation-aware Weight Quantization)** *Description:* A hardware-friendly quantization method that scales weights based on activation salience. It avoids the reconstruction overhead of GPTQ and preserves generalist capabilities of instruction-tuned models. *Key Reference:* Lin et al. (Lin et al., 2023) *Relevance:* Gaining popularity for high-throughput serving systems (e.g., vLLM).

**OpenVINO Toolkit** *Description:* Intel's toolkit for optimizing and deploying deep learning models on Intel hardware (CPUs, iGPUs, VPUs). It supports model optimization (including quantization) to accelerate inference at the edge. *Key Reference:* Kapo et al. (Kapo et al., 2024) *Relevance:* Critical for deploying quantized models on standard CPU architectures found in medical and industrial edge devices.

## D.2 Hardware-Specific Development Tools

**Xilinx Vitis AI** *Description:* A comprehensive development environment for AI inference on Xilinx hardware platforms, including FPGAs and ACAPs. It includes the Deep Learning Processor Unit (DPU) IP and tools for quantizing models to INT8. *Key Reference:* Sadr et al. (Sadr et al., 2025) *Relevance:* Enables the deployment of DCGANs and Transformers on FPGAs with optimized DSP utilization.

**NVIDIA TensorRT** *Description:* A high-performance deep learning inference SDK. It includes a post-training quantization calibration tool that optimizes kernels for INT8 execution on Tensor Cores. *Key Reference:* Implied in discussions of GPU acceleration (Dua & Patel, 2024). *Relevance:* The standard for maximizing throughput on NVIDIA GPUs (e.g., Jetson Orin (Dr.J.V.Anchitaalagammai et al., 2025)).

## D.3 Research Datasets and Benchmarks

**GLUE / SuperGLUE** *Description:* General Language Understanding Evaluation benchmarks. These datasets are standard for evaluating the accuracy degradation of quantized models compared to their full-precision counterparts. *Key Reference:* Referenced in Q-BERT analysis (Shen et al., 2020).

**ImageNet (for Vision Transformers)** *Description:* While primarily for vision, this dataset is used to benchmark hybrid CNN-Transformer architectures and Diffusion Transformers (DiT) under quantization. *Key Reference:* Kim et al. (Kim et al., 2024), Kim et al. (Kim et al., 2025).

## D.4 Further Reading on Hardware Co-Design

For researchers interested in the intersection of hardware architecture and algorithm design: - **Systolic Arrays:** Wang et al. (Wang et al., 2025) and Chang (Chang, 2025) provide in-depth analyses of configuring systolic arrays for GPGPUs and FPGAs. - **In-Memory Computing:** Rodriguez (RODRIGUEZ, 2025) explores dynamic cross-layer memory optimizations, a frontier beyond standard quantization. - **RISC-V Optimization:** Martínez et al. (Martínez et al., 2025) discuss the specifics of optimizing Transformer decoders for the open RISC-V instruction set architecture.

## References

Akita, Miyaji, Ryu, Babaie, & Reiskarimian. (2025). Guest Editorial Introduction to the Special Section on the 2025 IEEE International Solid-State Circuits Conference (ISSCC). *IEEE J. Solid State Circuits*. Https://doi.org/10.1109/jssc.2025.3626462.

Anderson, Doyle, & Gregg. (2019). Scalar Arithmetic Multiple Data: Customizable Precision for Deep Neural Networks. IEEE. (pp. 61-68). Https://doi.org/10.1109/arith.2019.00018

Auten, Tomei, & Kumar. (2020). Hardware Acceleration of Graph Neural Networks. IEEE. Https://doi.org/10.1109/dac18072.2020.9218751

Bouaggad, & Grabar. (2025). Search-optimized quantization in biomedical ontology alignment. *Frontiers in Artificial Intelligence*. Https://doi.org/10.3389/frai.2025.1662984.

Cai, Lin, & Han. (2022). *Efficient methods for deep learning*. Elsevier. Https://doi.org/10.1016/b978-0-12-822109-9.00013-8

Chang. (2025). Hardware-Software Co-Design for Efficient LLM Inference on PCIe-Based FPGAs Using Coarse-Grained Systolic Arrays. IEEE. (pp. 1-5). Https://doi.org/10.1109/socc66126.2025.11235351

Czakó, Kertész, & Szénási. (2025). Addressing Activation Outliers in LLMs: A Systematic Review of Post-Training Quantization Techniques. *IEEE Access*, *13*, 81917-81932. Https://doi.org/10.1109/access.2025.3568702.

Dettmers, Lewis, Belkada, & Zettlemoyer. (2022). LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. Https://doi.org/10.48550/arXiv.2208.07339

Dilshad, Khan, & Song. (2023). Efficient Deep Learning Framework for Fire Detection in Complex Surveillance Environment. *Computer systems science and engineering*. Https://doi.org/10.32604/csse.2023.034475.

Dr.J.V.Anchitaalagammai, Dr.S.Kavitha, R.Buurvidha, T.S.Santhiya, Roopa, & Sankari. (2025). Edge Artificial Intelligence for Real-Time Decision Making using

NVIDIA Jetson Orin, Google Coral Edge TPU and 6G for Privacy and Scalability. Https://doi.org/10.1109/ICVADV63329.2025.10960953

Dua, & Patel. (2024). *Hardware Optimization for Generative AI.* Apress. Https://doi.org/10.1007/979-8-8688-0917-0_3

Frantar, Ashkboos, Hoefler, & Alistarh. (2022). GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. *arXiv.org.* Https://www.semanticscholar.org/paper/7da0f2501034522e3d50af7e9b8fa7ec9d7b65b6.

Kapo, Akagić, & Buza. (2024). Semantic Segmentation of Brain Tumors: A Performance Evaluation Using DeepLabV3+, UNet, and Intel's OpenVINO Toolkit. *International Conference on Control, Decision and Information Technologies.* Https://doi.org/10.1109/CoDIT62066.2024.10708196.

Kim, Lee, & Kim. (2024). HyQ: Hardware-Friendly Post-Training Quantization for CNN-Transformer Hybrid Networks. *International Joint Conference on Artificial Intelligence.* Https://doi.org/10.24963/ijcai.2024/474.

Kim, Hwang, Oh, & Park. (2025). MixDiT: Accelerating Image Diffusion Transformer Inference With Mixed-Precision MX Quantization. *IEEE computer architecture letters.* Https://doi.org/10.1109/LCA.2025.3560786.

Lin, Tang, Tang, Yang, Dang, & Han. (2023). AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. Https://doi.org/10.48550/arXiv.2306.00978

Madhanegha, Vishnuvaradhan, Arun, & Surenther. (2025). Quantization of a Llama Language Model for improved Efficiency and Inference. Https://doi.org/10.21203/rs.3.rs-6021454/v1

Martínez, Catalán, Castelló, Mestre, & Quintana-Ortí. (2025). Latency-Critical Quantized Inference With Transformer Decoders on ARM and RISC-V CPUs. *IEEE Internet of Things Journal.* Https://doi.org/10.1109/JIOT.2025.3560382.

MIT. (2026). *» Song Han.* Https://mitibmwatsonailab.mit.edu/people/song-han/

Muller, Tyshka, Theisen, & Hanna. (2024). Co-design of a TinyLLM using Programmable Logic and Software on an FPGA. *Midwest Symposium on Circuits and Systems.* Https://doi.org/10.1109/MWSCAS60917.2024.10658754.

Risnanto, & Poerwandono. (2025). Optimasi Proses Klasifikasi Topik Berita Berbahasa Indonesia Menggunakan Model NLP Berbasis Onnx Runtime. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *9*(6), 9644-9649. Https://doi.org/10.36040/jati.v9i6.15707.

RODRIGUEZ. (2025). Experiential neural architecture selection: dynamic cross-layer memory for real-time inference optimization. Https://doi.org/10.21203/rs.3.rs-7378044/v1

Sadr, Haghighat, Pakniyat, Rahmati, & Gorgin. (2025). FPGA-Accelerated Real-Time DCGANs via Xilinx DPUs and Vitis AI. Https://doi.org/10.21203/rs.3.rs-7263274/v1

Shen, Dong, Yao, Gholami, Mahoney, & Keutzer. (2020). Q-BERT: Hessian Based Ultra Low Precision Quantization of BERT. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 8815-8821. Https://doi.org/10.1609/aaai.v34i05.6409.

Wang, Jia, Yang, Tan, Chang, Tie, & Huang. (2025). X-SA: An Efficient Configurable Systolic Array Computing Architecture for GPGPU. Https://doi.org/10.1109/HPCC67675.2025.00039

Zhu, Deng, Xin, Xia, Han, Wang, & Wang. (2025). Performance Analysis and Fronthaul Quantization Bits Allocation in Cell-free ISAC System. Https://doi.org/10.1109/ICCC65529.2025.11148976