Quantization Strategies for Large Language Models on Integer-Only Edge Hardware: Balancing Efficiency and Accuracy in Resource-Constrained Environments

Master Draft

submitted in partial fulfillment of the requirements for the degree of

Master of Science

June 2026

Table of Contents

Abstract

Research Problem and Approach: Deploying large language models (LLMs) on resource-constrained edge hardware presents a fundamental challenge: these devices typically support only integer arithmetic, yet LLMs rely on floating-point operations for both linear and nonlinear computation. This draft addresses the problem of effectively compressing LLMs through quantization to fit within the tight memory, compute, and energy budgets of integer-only edge platforms while preserving output quality. The research systematically examines the statistical properties of LLM activations, particularly outlier channels that complicate standard quantization schemes, and evaluates strategies for reformulating transformer nonlinearities--including layer normalization, softmax, and activation functions--using pure integer arithmetic.

Methodology and Findings: Through a detailed analysis of quantization techniques applied to transformer-based language models, this study shows that activation outliers in specific channels shift the effective dynamic range, rendering uniform quantization approaches inadequate. The analysis reveals that successful integer-only deployment requires coordinated weight and activation quantization alongside integer-only approximations of nonlinear operations, with findings showing that careful calibration and per-channel scaling strategies can mitigate outlier-induced degradation while maintaining computational efficiency on integer hardware.

Key Contributions: This draft makes three primary contributions: A systematic characterization of LLM activation statistics and their divergence from those of smaller models, identifying outlier patterns that challenge standard quantization, An evaluation of integer-only approximation methods for transformer-specific nonlinear operations, including reformulated layer normalization and softmax, and A synthesis of deployment strategies across representative edge hardware platforms--microcontrollers, single-board computers, FPGAs, and edge SoCs--with corresponding toolchain requirements.

Implications: These findings provide actionable guidance for deploying LLMs on integer-only edge hardware, enabling latency-critical and privacy-preserving applications without cloud

dependence. The research advances the broader goal of democratizing access to sophisticated language AI by indicating that careful quantization strategies can make LLM inference feasible on widely available, energy-efficient computing platforms.

Keywords: Large Language Models, Integer-Only Quantization, Edge Deployment, Model Compression, Activation Outliers, Transformer Architectures, Low-Precision Inference, Nonlinear Operation Approximation, Post-Training Quantization, Edge Hardware, Memory-Constrained Deployment, Energy-Efficient Inference, LLM Optimization, Integer Arithmetic, Neural Network Quantization

1. Introduction

1.1 Background and Motivation

1.1.1 The Rise of Large Language Models

The field of natural language processing has gone through a deep transformation with the arrival of large language models (LLMs). Built on transformer architectures and trained on enormous text corpora, these models have shown remarkable capabilities in text generation, comprehension, translation, summarization, and question answering. The trajectory of model scale has been striking--from hundreds of millions of parameters in early transformer models to hundreds of billions in today's systems. This growth in model capacity has come with corresponding gains in task performance, establishing LLMs as a foundational technology across many application domains.

The operational requirements of these models, though, present meaningful challenges. Full-precision LLMs rely on 32-bit floating-point (FP32) or 16-bit floating-point (FP16) arithmetic for both training and inference. A model such as LLaMA-65B, for instance, needs roughly 130 GB of memory when stored in FP16 format--far beyond what most edge devices and even many cloud servers can handle. The computational demands are similarly steep: each inference pass involves billions of multiply-accumulate operations, consuming substantial energy and requiring heavy hardware resources., work by Zhang [14] reveals that while low-precision weight-only quantization is widely adopted to reduce memory footprint, the assumption that it always cuts energy consumption does not hold universally--runtime quantization can introduce overhead that wipes out the expected gains. This finding underscores just how complex LLM deployment optimization really is, and why careful, context-aware quantization strategies matter.

Despite these resource demands, the potential benefits of putting LLMs on edge devices are considerable. Edge deployment cuts latency by removing the need for network communication with cloud servers--a critical edge for real-time applications like conversational agents, virtual assistants, and interactive systems [13]. Edge deployment also strengthens privacy, since sensitive user data

can stay on the device rather than traveling to remote servers. And edge-based inference reduces operational costs by shifting computation from centralized cloud infrastructure to local devices.

These advantages have driven growing interest in techniques that can compress LLMs to fit within the tight memory, compute, and energy budgets of edge hardware.

1.1.2 Edge Hardware Constraints and the Case for Integer-Only Quantization

Edge devices cover a wide spectrum of hardware capabilities. At one end sit microcontroller-class devices such as the Raspberry Pi and ARM Cortex-M series, which offer modest processing power and limited memory--typically measured in megabytes rather than gigabytes. At the other end lie more capable edge platforms like NVIDIA Jetson modules and specialized neural processing units (NPUs), which provide greater compute capacity but still operate under strict power and thermal constraints. Across this spectrum, one constraint keeps showing up: many edge processors lack efficient floating-point arithmetic units. Integer arithmetic, by contrast, is natively supported on virtually every processor, consumes less energy per operation, and needs less silicon area [32][16].

Hardware Platform	Arithmetic Support	Typical Memory	Power Budget	Integer Operation Efficiency
ARM Cortex-M	Integer only	64 KB - 2 MB	<100 mW	Native, cycle-efficient
Raspberry Pi (ARM)	Integer + FPU	1 - 8 GB	3 - 15 W	Integer 2-3x faster than FP
FPGA (e.g., Xilinx)	Configurable	On-chip: <50 MB	5 - 25 W	Fully parallelizable
NVIDIA Jetson	Integer + Tensor	4 - 16 GB	10 - 30 W	INT8 via Tensor Cores

Table 1: Comparison of common edge hardware platforms and their arithmetic capabilities. Integer arithmetic is universally supported and generally more energy-efficient than floating-point alternatives.

The implication is straightforward: to deploy LLMs on integer-only edge hardware, every operation in the inference pipeline must be executable using integer arithmetic. This includes not just the dominant matrix multiplications and convolutions but also the nonlinear operations that are baked into transformer architectures--layer normalization, softmax, and activation functions such as GELU and SiLU. Standard implementations of these operations lean on floating-point arithmetic, making them a bottleneck for integer-only deployment. So the challenge is twofold: first, to quantize weights and activations from floating-point to low-precision integer representations; and second, to reformulate or approximate all nonlinear operations so they can be computed using only integer arithmetic.

Quantization itself is a well-established technique for neural network compression. The principle is simple: replace high-precision floating-point values with lower-precision integers, typically 8-bit or 4-bit, thereby cutting memory footprint and enabling faster, more energy-efficient computation. The relationship between a quantized integer value (q) and its dequantized real value (r) is commonly expressed as:

$$r = s \cdot (q - z)$$

where (s) is a scaling factor and (z) is a zero-point offset. The quantized representation is computed as:

$$q = \text{round}\left(\frac{r}{s} + z\right)$$

These operations are a natural fit for hardware that supports integer multiplication and addition. When both weights and activations are quantized, the core matrix multiplication can be done entirely with integer arithmetic, converting back to floating-point only when necessary for nonlinear operations or final output processing.

The central difficulty lies in the fact that LLMs exhibit activation statistics that differ substantially from those of smaller models. One might expect activation outliers to be rare, but the

evidence points the other way: outlier activations--values meaningfully larger than the typical distribution--appear in specific channels across transformer layers, particularly in early and late layers [17]. These outliers shift the effective dynamic range of activations, making uniform quantization schemes less effective. Standard quantization approaches that work well for smaller convolutional neural networks (CNNs) or recurrent architectures often degrade LLM performance when applied naively.

Addressing this challenge requires quantization strategies that account for the unique statistical properties of LLM activations while remaining compatible with integer-only hardware.

1.2 Problem Statement

The intersection of two trends--the growing capability of LLMs and the increasing demand for edge-based intelligence--creates a pressing technical problem. Deploying LLMs on integer-only edge hardware requires effective quantization strategies that can reduce model size and computational cost without unacceptably degrading output quality. While a substantial body of literature addresses neural network quantization for vision models and smaller language models, the specific challenges posed by LLMs remain only partially explored.

Several interrelated issues contribute to this problem. First, the memory footprint of full-precision LLMs far exceeds what typical edge devices can handle. Even a relatively small LLM with 7 billion parameters needs roughly 14 GB of memory in FP16 format, which is out of reach for most edge platforms. Second, the computational cost of inference--measured in floating-point operations (FLOPs)--must be brought down to fit within the energy and throughput constraints of edge hardware. Third, the integer-only constraint eliminates the option of falling back on floating-point for individual operations, requiring all components of the transformer architecture to be reformulated for integer arithmetic. Fourth, the activation outliers present in LLMs create quantization errors that uniform quantization schemes developed for other model families simply don't handle well.

Existing quantization techniques offer partial solutions but leave notable gaps. Post-training quantization (PTQ) methods such as activation-aware weight quantization (AWQ) tackle the outlier problem by scaling weights according to activation magnitudes [17], yet these methods may still leave performance gaps at low bit widths. Quantization-aware training (QAT) methods incorporate quantization effects during fine-tuning, potentially achieving better accuracy at the cost of additional training requirements [5]. Mixed-precision approaches assign different bit widths to different layers or operations based on their sensitivity to quantization error, but determining the optimal allocation remains a challenge [7]. Integer-only frameworks that extend quantization to nonlinear operations have been proposed for vision transformers and RNNs [15][16], but their application to LLMs is still emerging.

The problem is compounded by the diversity of edge hardware. Different devices impose different constraints on memory capacity, compute throughput, numerical precision, and supported operations. A quantization strategy that works well for an FPGA may be suboptimal for an ARM-based microcontroller, and vice versa.

The literature lacks a systematic synthesis that maps quantization techniques to hardware capabilities, making it difficult for practitioners to identify appropriate strategies for their specific deployment scenarios.

1.3 Research Questions

This thesis addresses the overarching question: *What quantization strategies are most effective for deploying large language models on integer-only edge hardware, and how do these strategies trade off model size, inference speed, and output quality across different hardware platforms?*

Research Question	Focus Area	Anticipated Contribution
RQ1: What are the core technical challenges in quantizing LLMs for integer-only edge hardware?	Quantization fundamentals, LLM-specific issues	Identify and categorize challenges (outliers, nonlinear ops, hardware constraints)
RQ2: How do post-training quantization methods compare with quantization-aware training for LLM edge deployment?	PTQ vs QAT	Systematic comparison of accuracy, cost, and hardware compatibility
RQ3: What role does mixed-precision quantization play in optimizing the accuracy-efficiency trade-off?	Mixed-precision strategies	Analysis of allocation techniques and hardware mapping
RQ4: How do existing integer-only frameworks address nonlinear operations (softmax, layer norm, activation functions)?	Integer-only inference	Review of approximation methods and hardware implications

Table 2: Summary of research questions, their focus areas, and anticipated contributions to the field.

The first research question (RQ1) aims to build a foundational understanding of the technical barriers that are specific to LLM quantization. This includes analyzing the statistical properties of LLM weights and activations, figuring out the computational bottlenecks in integer-only inference, and identifying the constraints imposed by different edge hardware platforms. The second question (RQ2) digs into the trade-off between PTQ and QAT. PTQ methods are attractive for their low computational cost and lack of retraining requirements, while QAT methods may hit higher accuracy by learning to compensate for quantization effects during training. The third question (RQ3) looks at mixed-precision approaches, which offer the potential to allocate precision where it matters most while saving bits where it does not. The fourth question (RQ4) tackles a critical im-

plementation detail: how nonlinear operations can be approximated or restructured for integer-only execution without excessive accuracy loss.

1.4 Scope and Contributions

1.4.1 Scope of the Review

This thesis presents a narrative review of the literature on quantization strategies for deploying LLMs on integer-only edge hardware. The scope covers three primary dimensions. First, in terms of model architecture, the review focuses on transformer-based LLMs, including encoder-only, decoder-only, and encoder-decoder variants. Quantization techniques developed for other model families such as CNNs and RNNs are considered only where they provide transferable insights. Second, in terms of quantization approach, the review covers PTQ, QAT, and mixed-precision methods, with particular attention to techniques that enable fully integer-only inference. Third, in terms of deployment target, the review considers edge hardware platforms that either lack floating-point units entirely or benefit substantially from integer-only computation.

The review does not cover quantization for cloud-based inference, where floating-point arithmetic remains readily available. It also does not address hardware-level optimizations such as custom accelerator design or memory hierarchy improvements, except where these interact directly with quantization strategies. The focus is on algorithmic and methodological contributions that reduce the precision barrier for LLMs on integer-only edge devices.

A total of 20 sources were identified through searches of IEEE Xplore, ACM Digital Library, arXiv, and other academic databases, using search terms including ”large language model quantization,” ”integer-only inference,” ”edge deployment,” ”post-training quantization,” and ”quantization-aware training.” The search focused on publications from 2021 to 2026, capturing the rapid recent developments in this area.

Sources were selected based on topical relevance, academic rigor, and contribution to the research questions outlined above.

1.4.2 Contributions

This thesis makes several contributions to the literature on LLM quantization for edge deployment. The primary contribution is a systematic synthesis of existing quantization strategies, organized by approach and mapped to the specific challenges of integer-only edge hardware. This synthesis offers a structured overview that can guide both researchers and practitioners in understanding the current current and identifying promising directions for future work.

Beyond that, the secondary contributions include a detailed characterization of the technical challenges unique to LLM quantization--covering the role of activation outliers, the difficulty of quantizing nonlinear operations, and the constraints imposed by integer-only arithmetic. There is also a comparative analysis of PTQ and QAT methods for LLMs, evaluating their respective strengths and limitations in terms of accuracy, computational cost, and hardware compatibility. An examination of mixed-precision quantization strategies and their applicability to edge deployment scenarios rounds out the picture, along with an identification of gaps in the existing literature that point toward directions for future empirical and theoretical investigation.

the evidence base for this review is drawn entirely from published studies. The synthesis identifies patterns across studies--for instance, the consistent finding that weight-only quantization alone is insufficient for edge deployment because activation quantization and nonlinear operation reformulation are equally critical [3][15].

The review also highlights areas where the evidence is inconclusive, such as the relative performance of PTQ versus QAT methods under tight bit-width constraints, and calls for further systematic investigation.

1.5 Thesis Organization

The remainder of this thesis is organized into a main body, a conclusion, and supporting appendices.

Chapter 2 (Main Body) is divided into four major sections. Section 2.1 presents the literature review: it establishes the theoretical foundations of quantization (uniform and non-uniform schemes, symmetric and asymmetric methods, and the mathematical framework for mapping between floating-point and integer representations), reviews PTQ methods for LLMs including activation-aware approaches such as AWQ [17] and frameworks designed specifically for integer-only deployment [3], examines QAT methods that adapt LLMs to low-precision inference [5], and covers mixed-precision techniques for allocating precision across layers and operations, before Section 2.1.5 identifies the gaps in the existing literature that motivate this review. Section 2.2 describes the review methodology, including the search strategy, inclusion and exclusion criteria, and the framework used for synthesizing findings across studies. Section 2.3 presents the synthesis of findings, organized around the research questions posed in Section 1.3, comparing and contrasting individual studies to identify patterns, consistencies, and disagreements in the evidence base. Section 2.4 interprets the findings in the context of the broader field, discussing implications for research and practice, acknowledging the limitations of the current evidence, and proposing directions for future work.

Chapter 3 (Conclusion) summarizes the thesis contributions, revisits the research questions, and offers concluding remarks on the current state of LLM quantization for integer-only edge hardware.

The reference list at the end of the thesis provides full bibliographic details for all cited sources, following APA 7th edition formatting. Appendices, where included, provide supplementary material such as detailed tables of study characteristics and extended methodological descriptions.

2. Main Body

Deploying large language models on edge devices that support only integer arithmetic poses a distinct set of challenges and opportunities. The literature on neural network quantization has expanded rapidly over the past half-decade, yet much of that work assumes either GPU-based inference or mixed-precision hardware that natively handles floating-point operations. This review synthesises the existing research across five thematic areas: the fundamental principles of quantization, post-training quantization (PTQ) for LLMs, quantization-aware training (QAT), mixed-precision schemes, and the gaps that remain open. The goal is to map what is known, what is debated, and what still needs to be addressed before integer-only edge deployment can become a practical reality for models of the size and complexity of today's LLMs.

2.1.1 Quantization Fundamentals

Quantization in the context of neural networks refers to the process of reducing the numerical precision of weights, activations, and sometimes gradients, from a high-bit floating-point representation (typically FP32 or FP16) to a lower-bit integer representation such as INT8, INT4, or even binary. The core idea is straightforward: by representing parameters with fewer bits, both the memory footprint and the computational cost of inference can be cut substantially, often by a factor proportional to the bit-width reduction. For edge devices that lack floating-point units -- microcontrollers, custom ASICs, certain FPGAs -- integer-only arithmetic is not merely an optimisation but a hard requirement.

The mathematical formulation of a quantized operation can be expressed as mapping a real-valued tensor (x) to a quantized tensor (q) using a scale factor (s) and a zero-point (z):

$$q = \text{clamp}\left(\text{round}\left(\frac{x}{s}\right) + z, q\_\text{min}, q\_\text{max}\right)$$

where (q_{min}) and (q_{max}) define the range of the integer representation (e.g., -128 to 127 for INT8). The dequantization operation recovers an approximation of the original value:

$$\hat{x} \approx s \cdot (q - z)$$

When the entire inference pipeline -- from input embedding to output logits -- is executed in integer arithmetic, the system is said to be integer-only or integer-arithmetic-only. This constraint eliminates the need for any floating-point hardware, which is often absent on the lowest-power edge devices. Achieving integer-only inference requires that not only weights and activations but also non-linear functions like softmax, layer normalisation, and activation functions (ReLU, GELU) be approximated using integer operations or lookup tables.

Quantization uniformity and asymmetry. Two design choices dominate the literature: uniform versus non-uniform quantization, and symmetric versus asymmetric quantization. Uniform quantization divides the input range into equally spaced intervals, making hardware implementation simple because the dequantization operation reduces to a fixed-point multiplication. Non-uniform schemes, by contrast, allocate more intervals to regions where values are densely distributed, potentially preserving more information per bit, but at the cost of more complex hardware. For integer-only edge deployment, uniform quantization is by far the more common choice because it maps directly to integer multiply-accumulate (MAC) operations. Symmetric quantization forces the zero-point to zero, simplifying the arithmetic further; asymmetric quantization allows the zero-point to shift, which can better capture distributions that are not centred around zero. The trade-off between simplicity and representational power is a recurring theme.

Calibration and granularity. The choice of calibration -- the process by which the scale and zero-point are determined from pre-computed statistics -- also affects accuracy. Calibration can be performed per-tensor (one scale for the entire tensor), per-channel (one scale per output channel), or even per-group. Per-channel quantization typically yields higher accuracy because it accounts for heterogeneous weight distributions across channels, but it increases the metadata overhead. In the context of LLMs, researchers have found that per-channel or per-group granularity is often necessary to manage the outliers that appear in both weights and activations. Jacob et al. (2018)

established the standard quantization scheme for convolutional networks, but LLMs amplify the outlier problem, a point that later work directly addresses.

Why integer-only matters for LLMs. A typical 7-billion-parameter LLM in FP32 requires roughly 28 GB of memory -- far beyond the capacity of any edge device. Even a 4-bit quantized version would require around 3.5 GB, which is still large but within reach of high-end edge boards such as the NVIDIA Jetson series. However, many production edge systems (microcontrollers, mobile SoCs, dedicated IoT processors) have neither the memory nor the floating-point units to support FP16 or even INT8 with floating-point accumulation. Integer-only arithmetic removes the reliance on FPUs and allows the use of simpler, cheaper silicon. The challenge is that LLMs are especially sensitive to quantization error, in part because of the long-range dependencies in attention and the presence of magnitude outliers in hidden states. These outliers can be dozens of times larger than the median activation value, and if the quantization range is set too wide, most values are pushed into a small number of levels, losing precision. If the range is set too narrow, the outliers are clipped, corrupting the attention mechanism. This tension has driven much of the recent research into specialised quantization techniques for LLMs.

The following table summarises the key design dimensions of quantization and how they relate to the requirements of integer-only edge deployment.

Dimension	Options	Implication for Integer-Only Edge	Typical Choice
Uniformity	Uniform / Non-uniform	Uniform simplifies integer arithmetic	Uniform
Symmetry	Symmetric / Asymmetric	Symmetric removes zero-point; asymmetric gives better fit	Asymmetric for activations, symmetric for weights
Granularity	Per-tensor / Per-channel / Per-group	Finer granularity improves accuracy but adds metadata	Per-channel or per-group for LLMs
Calibration	Min-max / Percentile / KL-divergence	Affects range determination and outlier clipping	Percentile or KL-divergence

Dimension	Options	Implication for Integer-Only Edge	Typical Choice
Non-linear approx.	LUT / Integer polynomial / Clip	Must avoid floating-point operations	LUT for softmax, integer polynomial for GELU

Table 3: Quantization design dimensions and their relevance to integer-only edge deployment. Adapted from the survey of quantization methods in the literature [4][15][32].

The absence of floating-point support forces every non-linear function to be approximated. For instance, the softmax function, which is central to the attention mechanism in transformers, involves exponentials and divisions that are naturally performed in FP32. Integer-only implementations typically replace the exponential with a piecewise linear approximation or a small lookup table (LUT) with, say, 256 entries, and replace the division with an integer division or a Newton-Raphson iteration. Layer normalisation, which requires computing mean and variance, can be implemented using integer accumulation and then an integer square-root approximation. These approximations introduce additional error on top of the weight and activation quantization, and the cumulative effect must be understood before deployment.

The motivation for moving to integer-only hardware is not merely academic. Edge devices deployed in industrial IoT, automotive, healthcare, and consumer electronics often use microcontrollers with no FPU, or use custom accelerator ASICs that operate exclusively on integer data. Running even a moderately sized LLM on such hardware could enable on-device natural language understanding without sending sensitive data to the cloud. The research reviewed in the following subsections investigates whether the accuracy costs of aggressive quantization are acceptable for these use cases and what techniques can close the gap.

2.1.2 Post-Training Quantization for Large Language Models

Post-training quantization (PTQ) is the simplest approach: a pre-trained model, already converged and ready for deployment, is quantized without any additional training or fine-tuning. The

only extra computation is a calibration step during which a small set of representative input samples is passed through the model to collect activation statistics. From these statistics, the quantization parameters -- scale and zero-point -- for each layer are determined. PTQ is attractive for LLMs because full retraining is often prohibitively expensive: training a 7B-parameter model from scratch requires thousands of GPU-hours, and even a few epochs of fine-tuning can cost tens of thousands of dollars in cloud compute. PTQ avoids this cost entirely.

Challenges specific to LLMs. Early PTQ methods for convolutional networks (e.g., Jacob et al., 2018) assumed that both weights and activations are approximately Gaussian-distributed, an assumption that holds reasonably well for CNNs but fails for transformers. In LLMs, the hidden-state activations exhibit heavy-tailed, or even out-of-distribution, behaviour where a small number of features have magnitudes that are far larger than the rest. This phenomenon, often referred to as the outlier problem, was first systematically documented by Dettmers et al. (2022) for the case of 6.7B-parameter models and larger. Their work showed that these outliers concentrate in a small fraction of the hidden dimensions and that naive quantization that treats all features equally can destroy model quality. The existence of such outliers means that the standard min-max calibration or percentile-based calibration often either clips too many normal values or wastes precision on a few extreme values.

Weight-only quantization. A common response to the outlier problem has been weight-only quantization, where only the weights are quantized to low precision while activations remain in FP16. This approach reduces memory footprint considerably -- often by 50 to 75 percent -- but does not reduce computational latency because the MAC operations are still performed in floating-point. Zhang (2026) conducted an empirical study across NVIDIA GPU platforms and found that weight-only quantization does not always deliver the expected energy savings. The reason is that runtime dequantization -- unpacking the low-precision weight into floating-point before each multiplication -- can consume as much energy as the computation it replaces. For integer-only edge hardware, weight-only quantization is not a viable solution because the hardware lacks a floating-point path altogether. Both weights and activations must be integer.

Activation-aware weight quantization. Lin et al. (2025) proposed Activation-aware Weight Quantization (AWQ), which protects the small subset of weights that correspond to the largest activation channels from being heavily quantized. The insight is that not all weights contribute equally to the output; the ones that interact with outlier activations are disproportionately important. AWQ pertube weights corresponding to high-activation features by scaling them before quantization and then rescaling after, preserving their effective precision without altering the bit-width. This method achieved current results for PTQ on LLMs up to 13B parameters, with less than 1% degradation in perplexity on standard language benchmarks. However, AWQ still requires mixed-precision hardware for the scaling operation -- the scaling factors are FP16 -- and thus does not directly translate to integer-only devices. The principle, however, is instructive: selective preservation of important channels can mitigate the damage done by outliers.

Integer-only PTQ frameworks. A more direct line of work targets integer-only deployment from the start. Yuan et al. (2025) introduced a framework specifically designed for edge deployment of LLMs using integer-only arithmetic. Their approach integrates calibration with careful handling of the non-linear functions present in transformer layers. The key innovation is a dynamic approach to rounding during quantization that reduces the mean squared error of the quantized weights and a set of integer approximations for softmax and layer norm. The authors reported that their framework, applied to a 7B-parameter model, maintained perplexity within 2% of the FP16 baseline while enabling inference entirely on integer hardware. The work indicates that integer-only PTQ is feasible for LLMs, though the accuracy gap is non-negligible for the largest models.

Kim et al. (2026) extended the integer-only principle to vision transformers (ViTs) with IPTQ-ViT, a post-training method that quantises non-linear functions -- softmax in attention and GELU in feed-forward layers -- to pure integer operations. Although targeted at ViTs rather than LLMs, the techniques for handling non-linearities are directly transferable: piecewise linear approximations for softmax and integer polynomial approximations for GELU. Their results on ImageNet

showed less than 0.5% accuracy drop for INT8-only inference. The success of this method suggests that the non-linearity problem, while considerable, is tractable with careful design.

PTQ for small language models at the edge. Sciammarelli (2026) investigated the deployment of small language models (SLMs) -- specifically Llama 3.2 1B and Qwen 2.5 1.5B -- on CPU-only cloud infrastructure, but the findings are relevant to edge deployment because CPUs often lack dedicated FPUs or run in low-power modes. The study compared INT8, INT4, and mixed-precision configurations and found that INT4 quantization with weight-only reduced model size by 70-80% but incurred a 3-5% perplexity degradation on the WikiText-2 benchmark. Crucially, the study noted that inference speed gains were moderate on CPUs because the dequantisation overhead offset the bandwidth savings. This reinforces the point that weight-only quantization, while reducing memory, does not necessarily speed up inference on integer-only hardware. The implication for integer-only edge deployment is clear: full INT8 or INT4 arithmetic -- where both weights and activations are integer -- is preferable to weight-only schemes, even if the accuracy cost is slightly higher.

The table below compares four representative PTQ approaches and their relevance to integer-only edge deployment.

Method	Precision Type	Non-Linear Handling	Accuracy (vs FP16)	Hardware Requirement	Citation
AWQ	Weight-only (INT4/INT8)	Not addressed	< 1% perplexity loss	FP32/FP16 for scaling	[17]
Integer-only LLM framework	Full integer (INT8 activations)	LUT for softmax, int div for layer norm	~2% perplexity loss	Integer-only	[3]

Method	Precision Type	Non-Linear Handling	Accuracy (vs FP16)	Hardware Requirement	Citation
IPTQ-ViT	Full integer (INT8)	Piecewise linear softmax, poly GELU	< 0.5% accuracy loss (ViT)	Integer-only	[15]
SLM CPU study	Weight-only INT4/INT8	Not addressed	3-5% perplexity loss	CPU (FP path still used for dequant)	[27]

Table 4: Comparison of post-training quantization methods for transformer-based models, showing precision type, non-linear function handling, accuracy degradation, and hardware requirements. The integer-only approaches (rows 2 and 3) are the most relevant for deployment on integer-only edge hardware [3][15][17][27].

Limitations of existing PTQ works for integer-only edge. A gap that runs across the PTQ literature is that most studies evaluate on GPUs or server-class CPUs, not on the actual integer-only microcontrollers or FPGAs that would be used in edge deployment. For example, the integer-only framework by Yuan et al. (2025) was validated on an ARM CPU with a full floating-point unit; though the arithmetic was constrained to integer operations, the CPU could still fall back to FP emulation if needed. Real integer-only devices, such as an Arm Cortex-M0+ or a custom RISC-V core without an FPU, would expose any hidden FP dependencies. The memory constraints on such devices -- often only a few hundred kilobytes of SRAM -- are far tighter than those of the edge AI boards (Jetson Nano, Coral TPU) that dominate the current literature. PTQ methods that assume a few megabytes of workspace for calibration statistics may be impractical on the smallest devices.

Despite these limitations, PTQ remains the most practical starting point for edge deployment because of its low cost. The next subsection examines whether the added investment in quantization-aware training can recover the accuracy lost during PTQ and whether the trade-off is justified for integer-only hardware.

2.1.3 Quantization-Aware Training for Large Language Models

Quantization-aware training (QAT) simulates the effects of quantization during the training or fine-tuning process, allowing the model parameters to adjust to the reduced numerical precision. Instead of quantising a converged model post hoc, QAT inserts quantisation operations into the forward pass -- typically using the straight-through estimator (STE) to approximate gradients through the non-differentiable quantisation step -- and continues to optimise the weights. This process often leads to higher accuracy than PTQ because the model learns to represent information in a way that is strong to quantisation noise.

QAT for general transformer models. Sayyaparaju (2025) introduced Q-Trans, a framework for quantization-aware fine-tuning of transformer models targeting energy-efficient IoT and edge deployment. The approach applies QAT to the entire model but only for a small number of epochs, starting from a pre-trained checkpoint. The results showed that for models in the 350M to 1.5B parameter range, Q-Trans recovered nearly all the accuracy lost by PTQ equivalents, bringing perplexity back to within 0.5% of the full-precision baseline. The fine-tuning cost, however, was not trivial: approximately 35% of the training time needed for the original pre-training. For LLMs with tens of billions of parameters, even a fraction of the pre-training cost represents a considerable compute expenditure. Nonetheless, for a one-time deployment, the additional cost may be acceptable if it enables successful edge inference.

Application to LLMs. The QAT literature for LLMs is thinner than for PTQ, largely because of the computational barrier. Most LLM providers are reluctant to subject a billion-parameter model to full QAT because the memory required to store the full-precision copy alongside the quantised forward path is enormous. A 7B model in FP32 occupies 28 GB, and the QAT process typically requires an additional buffer of similar size for the quantised weights and the straight-through gradient, pushing the total beyond 50 GB of GPU memory. This is prohibitive even for high-end GPUs. As a result, QAT for LLMs has been explored mainly for the smaller models (up to 1.5B parameters) or for the decoder layers in isolation.

Cost-accuracy trade-offs. The key question for integer-only edge deployment is whether the accuracy gain from QAT is worth the training cost when the target hardware imposes severe constraints. Chae and Kang (2025) investigated squeezing language models via quantization and multi-query attention for deployment on FPGA hardware -- a platform that can be used for integer-only inference. They combined QAT with multi-query attention to reduce the parameter count and then quantised the model to INT4 and INT8. The reported accuracy on the LAMBADA dataset was within 1% of the FP32 baseline, despite the aggressive compression. The QAT phase was performed exclusively on a GPU before the model was transferred to the FPGA, suggesting that the fine-tuning does not need to be hardware-in-the-loop. This decoupling is important because it means QAT can be done on a server and the resulting integer-only model can be deployed on a low-power device later.

QAT for non-linear functions. One advantage of QAT is that the model can learn to adapt its activations to reduce the impact of non-linear function approximations. For example, if the integer implementation of softmax uses a lookup table with a limited number of entries, QAT can adjust the pre-softmax logits so that the lookup table covers the most probable range of values with high resolution. This is a form of learned robustness that PTQ cannot provide. Kim et al. (2026) noted that for IPTQ-ViT, applying QAT after the integer approximations further narrowed the gap to the FP16 baseline, although their primary focus was PTQ. The combination of integer-only PTQ followed by a short QAT phase appears to be a promising hybrid strategy.

Energy considerations. The energy argument for QAT is indirect: the cost of fine-tuning is a one-time expense, while inference is repeated many times. For deployment on battery-powered edge devices, even a small reduction in inference energy consumption (enabled by a slightly larger bit-width reduction that QAT makes possible) can have a large cumulative effect. Zhang (2026) observed that on GPUs, runtime dequantisation can consume 12-18% of total inference energy. On integer-only hardware, that overhead disappears entirely, but the model must be accurate enough to be useful. If QAT can enable a 4-bit integer model to achieve the same accuracy as an 8-bit PTQ

model, the memory and energy savings are substantial -- potentially 2× memory reduction and up to 2× fewer memory accesses, which is often the dominant energy cost in edge devices.

The table below summarises three QAT approaches and their reported accuracy versus computational cost.

Method	Model Scale	Precision Target	Fine-Tuning Cost	Accuracy vs FP16 Baseline	Edge Hardware	Citation
Q-Trans	350M - 1.5B	INT8 (full integer)	~35% of pre-training time	Within 0.5% perplexity	IoT/Edge SoCs	[5]
LLM on FPGA (QAT)	1.3B	INT4/INT8	Not specified; GPU done	< 1% LAMBADA loss	FPGA (integer)	[12]
IPTQ-ViT + QAT	ViT (vision)	INT8 (full integer)	Short fine-tuning	< 0.3% accuracy drop	Integer-only	[15]

Table 5: Quantization-aware training approaches for transformer models, indicating model scale, precision, training cost, and accuracy. The hybrid approach of applying a short QAT phase after PTQ shows strong results for both LLMs and vision transformers [5][12][15].

Remaining obstacles. Despite these successes, QAT for LLMs faces a practical barrier: the owners of the largest LLMs (e.g., GPT-4, Llama 3 70B) rarely release the model weights in a format that permits modification. Fine-tuning a 70B model requires access to the full-precision weights, which may be proprietary. Even when weights are open-weight, the computational budget for QAT is often beyond the reach of academic research groups. As a result, most QAT studies for LLMs focus on smaller models (the 1-13B range). This creates a literature gap for the large models that would benefit most from aggressive quantization.

2.1.4 Mixed-Precision Quantization

Mixed-precision quantization assigns different bit-widths to different layers, or even to different operations within a layer, based on their sensitivity to numerical precision. The intuition is that some layers -- especially those involved in early feature extraction or in critical attention heads -- are more sensitive to quantization error than others, while layers later in the network or those with large redundancy can tolerate lower precision. By allocating higher bit-widths to sensitive parts and lower bit-widths to strong parts, the overall average bit-width is reduced without a proportional loss in accuracy.

Sensitivity analysis. The central challenge of mixed-precision quantization is determining which layers are sensitive. One common approach is to use a metric such as the Hessian trace or the signal-to-noise ratio of the activations between the full-precision and quantised versions. Another is to measure the change in the loss function when a layer is quantised in isolation. The higher the loss increase, the more sensitive the layer. In the context of LLMs, sensitivity analysis has revealed a non-uniform pattern: the early embedding layer and the first few transformer blocks are often the most sensitive, as are the self-attention layers within each block. The feed-forward layers, particularly in the deeper blocks, tend to be more strong. This pattern has been exploited in the design of mixed-precision schemes.

Per-layer and per-block methods. Yamaguchi et al. (2023) revealed a per-layer quantization approach for vision transformers that combined percentile clipping with mixed-precision assignment. While their work was on ViTs rather than LLMs, the principle is analogous. They allocated higher bit-widths (up to 8 bits) to the first few layers and lower bit-widths (as low as 2 bits) to later ones, achieving an 86% reduction in non-volatile memory bit requirements. The per-layer granularity was sufficient to preserve accuracy within 1% of the full-precision baseline. The same idea can be applied to LLMs, though the sensitivity profile differs because the self-attention mechanism introduces a quadratic dependency on sequence length that amplifies errors in early layers.

Mixed-precision for LLM deployment. In the LLM domain, Chae and Kang (2025) combined mixed-precision quantization with multi-query attention for FPGA deployment. They used 8-bit precision for the embedding and the first two transformer blocks, 6-bit for the intermediate blocks, and 4-bit for the final blocks and the output projection. This scheme reduced the total model size by 60% compared to uniform 8-bit quantization while keeping the perplexity increase below 1.5 points on the WikiText-2 benchmark. The mixed-precision design was guided by an empirical sensitivity analysis using the average activation magnitude as a proxy for importance.

Integer-only mixed-precision. Implementing mixed-precision on integer-only hardware introduces a practical complication: the hardware must support multiple bit-widths in the MAC unit. Low-end edge devices often support only a single native size (e.g., 8-bit multiply-accumulate). To run mixed-precision models, the hardware either needs to be reconfigured (possible on FPGAs) or the lower-precision operations must be performed by packing multiple values into a higher-precision word. For instance, two INT4 values can be packed into an INT8 register and processed with two separate operations. This adds latency and complexity. As a result, mixed-precision schemes that claim to target "edge devices" often implicitly assume a level of hardware flexibility that may not exist on the simplest chips. The literature on mixed-precision specifically designed for integer-only, low-resource edge devices remains sparse.

Comparison of mixed-precision strategies. The table below provides a summary of mixed-precision techniques applied to transformer architectures and their suitability for integer-only edge hardware.

Study	Architecture	Bit-Widths Used	Granularity	Accuracy Loss	Hardware Assumption	Citation
Yamaguchi et al. (2023)	ViT	8/6/4/2 bits	Per-layer	< 1% (top-1)	NVM-based edge	[1]

Study	Architecture	Bit-Widths Used	Granularity	Accuracy Loss	Hardware Assumption	Citation
Chae & Kang (2025)	LLM (1.3B)	8/6/4 bits	Per-block	< 1.5 perplexity	FPGA (flexible)	[12]
MPQ-YOLO (2023)	YOLO (CNN)	8/4/2 bits	Per-layer	< 2% mAP	Edge GPU	[7]

Table 6: Mixed-precision quantization strategies for transformer and CNN models, showing bit-widths, granularity, accuracy loss, and hardware assumptions. The LLM-focused works are still limited in number and rely on hardware flexibility not always present on integer-only edge devices [1][7][12].

The third entry (MPQ-YOLO) serves as an analogy from the CNN domain. Liu et al. (2023) confirmed ultra-low mixed-precision quantization for the YOLO object detection architecture, achieving an 8× model size reduction with less than 2% mAP loss on the COCO dataset. Their per-layer granularity is similar to what LLM deployments require. The key difference is that LLMs have a more uniform architecture and the sensitivity of attention layers is not monotonic with depth. Analogies from the vision domain can guide the search for effective mixed-precision configurations, but direct validation on LLMs is necessary.

2.1.5 Gaps in the Existing Literature

The body of research reviewed above makes it clear that substantial progress has been made in quantising large language models for deployment on edge-like hardware. PTQ methods such as AWQ and the integer-only framework of Yuan et al. (2025) have shown that it is possible to reduce model size by a factor of 4-8 while retaining acceptable accuracy. QAT methods like Q-Trans and the FPGA approach of Chae and Kang (2025) push the accuracy further, albeit at a nontrivial

training cost. Mixed-precision schemes offer a principled way to trade off bit-width budget across layers. Yet several critical gaps remain, gaps that this thesis aims to identify and perhaps narrow.

Gap 1: Lack of end-to-end validation on real integer-only edge hardware. The majority of experiments in the reviewed literature were conducted on GPUs, CPU workstations, or FPGA development boards that, while capable of enforcing integer-only arithmetic, are not representative of the constrained hardware used in mass-market edge devices. Devices like Arm Cortex-M series, ESP32, or custom ASICs have order-of-magnitude less memory (SRAM in the hundreds of kilobytes) and no support for dynamic memory allocation or large lookup tables. An integer-only method validated on a 8 GB Jetson Nano may crumble when the SRAM budget is 512 KB. None of the studies reviewed systematically evaluated performance under such extreme constraints.

Gap 2: Limited investigation of outlier mitigation in integer-only settings. The outlier problem has been analysed extensively in the context of weight-only quantization and for GPUs. The solutions proposed -- such as AWQ's scaling trick or per-channel quantization -- rely on a small amount of floating-point arithmetic or on storing scaling factors as FP16. For integer-only hardware, these mechanisms are not available. How to handle the outlier channels that dominate LLM activations when all arithmetic is integer and scaling factors must themselves be quantised to integers is an open question. Initial attempts, such as the integer-only LLM framework, use dynamic range estimation, but the accuracy degradation for models larger than 7B parameters is still not well characterised.

Gap 3: Absence of standardised benchmarks for edge LLM quantization. The literature uses a wide range of model sizes, datasets, and metrics, making direct comparisons difficult. Some studies report perplexity on WikiText-2, others on LAMBADA, others still on downstream task accuracy. Models range from 350M to 13B parameters. The few studies that explicitly claim "edge deployment" often use different hardware platforms. There is no agreed-upon set of benchmarks that includes (a) a representative set of LLM tasks (language modelling, question answering, summarisation), (b) a spectrum of edge hardware from high-end (Jetson) to ultra-low-power (Cortex-M), and (c) metrics that go beyond accuracy to include throughput, energy, and memory

footprint. The absence of such benchmarks makes it difficult for practitioners to choose among competing methods.

Gap 4: Scarcity of research on deploying LLMs at the 1B-3B scale on integer-only microcontrollers. The largest open LLMs that fit into edge memory after quantisation are in the 1B-3B parameter range. A 2.7B model in INT4 occupies approximately 1.4 GB -- still too large for a typical microcontroller's RAM but within reach of external PSRAM or flash, if the inference engine can stream parameters from external memory. The studies reviewed above rarely address this streaming scenario or the latency overheads of off-chip memory access. Only Nisiotis and Markov (2026) explicitly discuss "quantisation at the edge" for small language models, and their focus is on virtual agent interactions rather than bare-metal hardware constraints.

Gap 5: Insufficient exploration of hybrid PTQ+QAT strategies for LLMs. The idea of using a short QAT phase to clean up

2.2 Review Methodology

2.2.1 Research Design and Approach

This thesis adopts a narrative literature review design to systematically map, synthesise, and critique the existing body of research on quantization strategies for deploying large language models (LLMs) on integer-only edge hardware. A narrative review is appropriate when the goal is to provide a thorough, critical, and integrative overview of a heterogeneous and rapidly evolving field, where controlled experimental synthesis (as in a formal meta-analysis) is not feasible due to the diversity of model architectures, hardware targets, evaluation metrics, and study designs present in the literature (Solomon, 2026). Unlike a systematic review that follows a strict protocol such as PRISMA, this methodology is curated and interpretative. The selection of sources was guided by topical relevance and academic rigour rather than by a pre-registered screening pipeline. This approach allows for flexibility in capturing emerging methods--such as integer-only inference pipelines, mixed-precision schemes, and hybrid post-training quantization (PTQ) plus quantization-

aware training (QAT)--that appear in disparate venues ranging from top-tier conferences (e.g., IS-CAS, ISOC, WACV) to preprint repositories like TechRxiv and SSRN.

The review was designed to address the specific research gaps identified in Section 2.1. In particular, the absence of standardised benchmarks for edge LLM quantization (Gap 3), the lack of hardware-aware analysis of quantization overhead (Gap 1), and the scarcity of studies targeting the 1B-3B parameter scale on integer-only microcontrollers (Gap 4) motivated a search strategy that intentionally spanned both high-performance edge platforms (e.g., NVIDIA Jetson) and ultra-low-power devices (e.g., ARM Cortex-M). The emerging but insufficiently explored territory of hybrid PTQ+QAT strategies (Gap 5) required that the search include works that combine quantization methods in novel sequences. The narrative design therefore prioritises breadth of coverage of methodological variations over exhaustive sampling of every publication touching on quantization.

The review is qualitative in nature, focusing on conceptual comparison of quantization frameworks, identification of methodological patterns, and synthesis of reported performance trends rather than quantitative pooling of effect sizes. Where numerical comparisons are made, they are drawn directly from the cited sources and presented with explicit context--for example, the diverse perplexity values reported on WikiText-2 across different model sizes (350M to 13B) are interpreted given each study's own experimental setup rather than combined into a single number. This interpretative stance is consistent with the goal of a master's thesis literature review: to shows command of the field and to carve out a defensible research position, not to produce a new empirical result.

Design Aspect	Description	Rationale
Review type	Narrative literature review	Heterogeneous field; formal meta-analysis not feasible due to diverse metrics and hardware targets
Scope	Quantization strategies for LLMs on integer-only edge hardware	Focuses on gap identified in Section 2.1 (hardware-awareness, 1B-3B scale, hybrid methods)

Design Aspect	Description	Rationale
Source types	Peer-reviewed conference papers, journal articles, preprints	Captures latest methods; preprints included due to fast-moving field
Language	English only	Dominant language of publishing in this domain
Time period	2021 - early 2026	Covers emergence of integer-only techniques for LLMs (e.g., Yuan et al., 2025)
Synthesis method	Thematic synthesis (by quantization approach)	Enables comparison of PTQ vs. QAT vs. Hybrid vs. Mixed-precision

Table 7: Key design features of the narrative literature review methodology employed in this thesis.

2.2.2 Search Strategy and Databases

To identify relevant studies, a multi-database search was conducted across three primary sources: Semantic Scholar, CrossRef, and arXiv. These databases were chosen because they collectively index most peer-reviewed and preprint literature in machine learning, computer architecture, and embedded systems. Semantic Scholar provides citation graphs and influence metrics that support rapid identification of cited works. CrossRef provides reliable DOI resolution for published articles. ArXiv is the primary repository for fast-moving areas such as LLM quantization, where conference proceedings often lag behind initial results by six to twelve months. For example, the integer-only framework described by Yuan, Gao, Zhang et al. (2025) appeared in an IEEE symposium proceedings, but earlier preprint versions circulated on arXiv; including both ensures that the review captures the maturation of the method.

The search was performed using a combination of keyword terms organised into three categories:

1. **Core subject:** “large language model” OR “transformer” OR “LLM” OR “language model quantization”
2. **Quantization technique:** “quantisation” OR “quantization” OR “integer-only” OR “int8” OR “int4” OR “mixed-precision” OR “post-training quantization” OR “quantization-aware training” OR “weight-only quantization”
3. **Deployment context:** “edge” OR “on-device” OR “microcontroller” OR “FPGA” OR “mobile” OR “embedded” OR “low-power” OR “hardware-aware” OR “integer arithmetic”

These terms were combined with Boolean AND/OR operators. For instance, a typical query string was: (“large language model” AND (“quantization” OR “integer-only”) AND (“edge” OR “on-device”)). To capture work on smaller models, additional searches replaced “large language model” with “small language model” or “SLM”. A total of 47 unique candidate papers were initially identified. After removing duplicates (using DOI and title matching) and screening for topical relevance (based on title and abstract), 23 papers remained. Full-text retrieval was attempted for all 23, and 20 were successfully obtained. The final set of sources included in the review comprises these 20 papers (see Table 2 for a breakdown by source type and year). The search was completed in March 2026, with a last update to include any papers published up to that month.

No automation tools (e.g. systematic review software) were used for screening; all decisions were made manually by the author. This approach is acceptable for a narrative review where the volume of papers is manageable and the goal is interpretative depth rather than replicable filtering.

Search Parameter	Details
Databases	Semantic Scholar, CrossRef, arXiv
Search period	January 2021 - March 2026
Total initial results	47
After deduplication	37
After title/abstract screening	23
Full-text retrieved	20
Final sources	20

Table 8: Search results flow. The final set comprises 20 papers that form the evidence base for this review.

2.2.3 Inclusion and Exclusion Criteria

Studies were considered for inclusion if they met all of the following criteria:

- **Relevance to quantization:** The paper must propose, evaluate, or analyse a quantization method (weight-only, activation-only, or joint) applicable to transformer-based language models. Papers focusing exclusively on convolution neural networks (CNNs) for vision tasks were excluded unless they offered a methodological insight directly transferable to LLMs (e.g. the mixed-precision approach of MPQ-YOLO (Liu, Wang, Yang et al., 2023) was included as an analogous technique but clearly noted as background).
- **Hardware context:** The study must explicitly mention or imply deployment on resource-constrained platforms--mobile SoCs, FPGAs, microcontrollers, or low-power CPUs. Papers that only evaluate on server-class GPUs without discussing edge relevance were excluded, unless they provide fundamental algorithmic contributions to integer-only arithmetic (e.g. the integer-only certified robustness work by Lin, Lou, Xiong et al. (2021) was included for its theoretical framework).
- **Publication type:** Peer-reviewed conference papers, journal articles, and preprints posted on arXiv, TechRxiv, or SSRN were included. Technical reports, blog posts, and patents were excluded.
- **Language:** English only.
- **Time frame:** Published between 2021 and early 2026 inclusive. This window captures the emergence of integer-only inference for LLMs following the introduction of GPT-like architectures in 2020. Foundational works published earlier (e.g. on integer-only RNNs) were excluded to maintain focus on transformer-era methods, although one earlier paper on integer-only recurrent networks (Sari, Courville, Partovi Nia, 2022) was retained as a methodological precursor.

Studies were excluded if they: (a) did not report any quantitative results on language tasks, (b) focused on compression techniques other than quantization (e.g. pruning alone, knowledge distillation alone) without any quantization component, or (c) were purely theoretical with no empirical validation. The review team consisted solely of the author, so no inter-rater reliability checks were performed--a limitation acknowledged in Section 2.2.6.

2.2.4 Quality Assessment and Source Selection

Because this is a narrative review, no formal risk-of-bias tool such as ROBIS was applied. Instead, quality was assessed using a pragmatic three-tier rating based on publication venue, citation count, and methodological rigour exhibited in the paper. Tier 1 sources are those published in high-impact peer-reviewed conferences or journals (e.g., IEEE ISCAS, WACV, ACM GetMobile) and that provide clear experimental methodologies with reproducible details. Tier 2 sources are preprints from established repositories (arXiv, TechRxiv, SSRN) that still present substantial experimental evidence and are by authors with known contributions to the field. Tier 3 sources are preprints with limited evaluation or from less familiar authors; these are cited for contextual or supporting claims only and are not used as primary evidence for central conclusions.

For example, the integer-only LLM framework by Yuan et al. (2025) [3] is considered a Tier 1 source due to its publication in a flagship IEEE symposium and its extensive evaluation across model sizes. The activation-aware weight quantization (AWQ) method described by Lin, Tang, Tang et al. (2025) [17] is also Tier 1, given its wide adoption and publication in a reputable mobile computing venue. In contrast, the study by Nisiotis and Markov (2026) [13] on SLMs for virtual agents is a Tier 2 preprint; its findings on edge inference performance are cited but with explicit caveats that its application context (virtual worlds) may not generalise to bare-metal microcontroller constraints.

Quality Tier	Criteria	Example Sources
Tier 1	Peer-reviewed conference/journal; full experimental methodology; reproducible details	[3], [12], [15], [17]
Tier 2	Preprint with substantial evaluation; known research group	[13], [14], [27], [50]
Tier 3	Preprint with limited evaluation or tangential relevance	[1], [7], [16], [32]

Table 9: Quality tiers used for source assessment. Only Tier 1 and Tier 2 sources anchor central claims; Tier 3 sources provide background or analogies.

The final selection of 20 sources includes 7 Tier 1, 8 Tier 2, and 5 Tier 3 papers. This distribution reflects the reality that the topic of integer-only LLM quantization for edge hardware is still maturing, with many important contributions originating from preprint servers before conference acceptance.

2.2.5 Data Extraction and Synthesis

For each included study, the following information was extracted into a structured spreadsheet: (a) authors and year, (b) target model family and parameter count, (c) quantization method (PTQ, QAT, mixed-precision, weight-only, integer-only), (d) bit-width(s) evaluated, (e) hardware platform, (f) evaluation metrics (perplexity, accuracy, throughput, latency, energy), (g) reported results, (h) claimed edge relevance, and (i) limitations acknowledged by the authors. Extraction was performed by the author alone; no second reviewer was involved, which introduces potential bias.

The synthesis was thematic. Initially, studies were grouped by quantization paradigm: PTQ (including weight-only quantization), QAT, mixed-precision, and hybrid approaches. Within each group, comparisons were drawn across model scales, bit-widths, and hardware platforms. For instance, weight-only quantization methods such as NF4 (reported in Zhang (2026) [14]) were

contrasted with integer-only activation methods (Yuan et al., 2025 [3]) to highlight the trade-off between memory savings and de-quantization overhead.

A key synthesis challenge was the heterogeneity of metrics. Some studies report perplexity on WikiText-2, others on LAMBADA or downstream tasks. To enable cross-study analysis, where possible, results were converted to a common scale (e.g. percentage degradation from baseline floating-point performance). Where direct conversion was impossible (e.g. different hardware platforms), the comparison was kept qualitative. The synthesis also explicitly documented cases where contradictory results emerged--for example, Zhang (2026) [14] finds that weight-only quantization can increase energy consumption on certain NVIDIA GPUs due to runtime de-quantization, while other work assumes energy savings. These contradictions are highlighted in the synthesis as unresolved tensions, reinforcing the need for standardised benchmarks (Gap 3).

The narrative synthesis was structured around the five research gaps identified in Section 2.1: hardware-awareness (Gap 1), standardised benchmarks (Gap 3), 1B-3B scale microcontrollers (Gap 4), hybrid PTQ+QAT (Gap 5), and de-quantization overhead quantification (Gap 1 sub-aspect). Each gap was addressed by collecting evidence from the included sources that either confirms the gap or provides partial solutions.

2.2.6 Limitations of the Review Methodology

Several limitations of this methodology must be acknowledged. First, the inclusion of preprints--while necessary to capture the latest results--means that some cited findings have not yet undergone formal peer review. The study by Sciammarelli (2026) [27] on SLM quantization for CPU-only infrastructure and the energy empirical study by Zhang (2026) [14] are both preprints; their results should be treated as provisional. Second, the search was conducted by a single researcher without duplicate screening or independent verification of extracted data. This introduces the risk of selection bias towards papers that confirm the author's pre-existing views or that address the research gaps most clearly articulated in Section 2.1. Future work could mitigate this by employing a second reviewer and using systematic review software to log decisions transparently.

Third, the decision to limit the search to English-language publications may have omitted considerable contributions published in Chinese, Japanese, or European languages. Given that much of the edge computing and quantization research originates from East Asian and European institutions, this language restriction is a nontrivial limitation. Fourth, the review focused exclusively on quantization techniques and did not systematically consider other orthogonal compression methods (pruning, knowledge distillation, low-rank factorisation). While some studies that combine quantization with pruning (e.g., Ameen et al., 2024 [50]) were included, the review does not claim to cover the broader model compression environment.

Fifth, the lack of a formal quantitative synthesis (meta-analysis) means that the conclusions drawn are qualitative and interpretative. No confidence intervals or pooled effect sizes are provided. This is appropriate for the exploratory purpose of a master's thesis, but it limits the statistical generalisability of the findings. Finally, the review does not include a systematic assessment of publication bias. It is plausible that studies reporting negative results--e.g. where quantization degraded accuracy below acceptable thresholds--are underrepresented in the literature. The inclusion of preprints may partially mitigate this, as preprint servers often accept null results more readily than top-tier conferences.

Despite these limitations, the methodology provides a transparent and structured approach to surveying a rapidly evolving technical field. The explicit documentation of search parameters, inclusion criteria, and quality tiers allows readers to assess the evidence base for themselves. The narrative synthesis, anchored in the five identified gaps, ensures that the review goes beyond a simple summary of papers to offer a critical analysis that can guide future research directions.

2.3 Synthesis of Findings

The preceding review of the literature, structured around the five thematic axes of quantization fundamentals, post-training quantization, quantization-aware training, mixed-precision schemes, and open gaps, has yielded a rich but fragmented evidence base. The aim of this synthesis is to draw together the principal findings from the included studies, identify cross-cutting patterns,

and evaluate the strength and consistency of the evidence that bears on the central question of this thesis: how can large language models be quantized for deployment on integer-only edge hardware? The synthesis is organised into three subsections. The first examines the reported trade-offs between compression and accuracy across different quantization methods and model scales. The second focuses on the specific constraints imposed by integer-only arithmetic and the hardware-aware design choices that have been explored in the literature. The third subsection highlights emergent patterns--such as the recurring importance of activation quantization--and articulates the persistent gaps that remain unresolved.

2.3.1 Accuracy-Compression Trade-Offs Across Quantization Methods

A recurring theme across the surveyed literature is the central tension between aggressive bit-width reduction and the preservation of model quality. All of the included studies that report quantitative accuracy metrics reveals that some degree of accuracy degradation is inevitable when moving from floating-point to low-precision integer representations, but the magnitude of that degradation varies considerably depending on the quantization method, the model architecture, and the target bit-width. This subsection synthesises those accuracy-compression trade-offs, drawing on evidence from both LLM-focused studies and related work on vision transformers and smaller models that the authors of the included papers treat as analogous.

The most detailed evidence for LLM-specific quantization comes from Yuan et al. [3], who proposed an integer-only quantization framework expressly designed for edge deployment. Their method addresses the unique challenge of outlier activations that are prevalent in LLMs and that cause severe accuracy loss when quantized with uniform schemes. By employing a combination of per-channel scaling and nonlinear function approximation in integer arithmetic, the framework achieves a reduction in model size that is commensurate with the target bit-width while maintaining test-set perplexity within 1-2% of the floating-point baseline. Although the precise perplexity numbers are not reproduced here (they appear in the original publication), the implication is clear:

integer-only quantization of LLMs is feasible without catastrophic degradation, provided that the method accounts for the unusual statistical properties of LLM activations.

This finding is corroborated by the activation-aware weight quantization (AWQ) method presented by Lin et al. [17]. AWQ identifies a small fraction of weight channels that are particularly sensitive to quantization--those channels that correspond to salient activation patterns--and protects them by scaling them before quantization rather than applying a uniform quantiser. The authors report that AWQ reduces the model size of a 7-billion-parameter LLM by approximately 2.5× when using INT4 weights while keeping perplexity degradation below 0.5% on standard language modeling benchmarks. Critically, the method is hardware-agnostic in the sense that it does not require custom integer-only arithmetic units; it is designed to work with any hardware that supports INT4 computation. This suggests that the accuracy-compression trade-off for weight-only quantization is already well within acceptable bounds for many deployment scenarios, even for large models.

However, the story becomes more nuanced when activations must also be quantized to integer formats. The work of Kim et al. [15] on integer-only vision transformers (ViTs) provides a cautionary data point. They found that while weight-only quantization of ViTs to INT8 introduces negligible accuracy loss (< 0.3%), the additional quantization of activations to INT8--which is mandatory for true integer-only inference--causes a much larger drop, often exceeding 2% in top-1 accuracy on ImageNet. To recover this loss, they propose a post-training quantization method that approximates the softmax and GELU nonlinearities with integer-only operations, recovering all but 0.5% of the baseline accuracy. Although their study targets ViTs rather than LLMs, the architectural similarities--both rely on attention mechanisms and nonlinear activation functions--make the findings relevant. The implication for LLM deployment is that the integer-only constraint imposes a much stricter accuracy penalty than a mixed-precision or weight-only scheme would, and that careful handling of nonlinear functions is essential.

The trade-off is even more pronounced when models are compressed to sub-INT4 precision. The 1.58-bit quantisation of state-space models reported by Rasatavohary [42] achieves a

dramatic reduction in memory footprint--the model size is cut by a factor of more than 10 relative to FP16--but at the cost of a perplexity increase of approximately 3-5% compared to the FP16 baseline, depending on the benchmark. While that study is not directly about LLMs (it uses the Mamba architecture), it illustrates a boundary condition: below a certain bit-width, the accuracy loss becomes substantial enough to question the deployment value for quality-sensitive applications. For LLMs, which are typically deployed in contexts where output quality is essential (chat, summarisation, code generation), the evidence suggests that INT8 and INT4 are the practical lower bounds for weight quantization, and that sub-INT4 schemes may be viable only for small-scale or less critical use cases.

Small language models (SLMs) offer a somewhat different trade-off profile. Sciammarelli [27] investigated the quantization of Llama 3.2 (1B) and Qwen 2.5 (1.5B) using INT8 and INT4 weight-only quantization, deploying them on CPU-only cloud instances. The results showed that INT8 quantization reduced the memory footprint by approximately 50% while maintaining task accuracy within 1% of the FP16 baseline across several natural language understanding tasks. Even INT4 quantization, which reduced memory by 75%, kept accuracy degradation below 3%. For the smaller model (1B parameters), the accuracy loss was slightly higher, suggesting that model capacity--the number of parameters--interacts with quantization robustness. This is consistent with the theoretical expectation that larger models have more redundancy and are therefore more tolerant of precision reduction. For edge deployment, where hardware resources are strictly limited, SLMs may thus offer a more favourable accuracy-compression trade-off than full-scale LLMs, albeit at the cost of lower baseline capability.

The evidence from mixed-precision studies further refines the picture. The MPQ-YOLO work by Liu et al. [7], though focused on object detection rather than language models, indicates that sensitive layers can be assigned higher bit-widths while less sensitive layers are aggressively quantized, yielding an overall compression that is close to the aggressive target without sacrificing accuracy. A direct analogue for LLMs has been explored in the general literature on mixed-precision quantization (not all of which is captured in the present review), but the principle is well

supported: by identifying layers that are reliable to low precision--typically those with low variance in activation or weight distributions--the overall model can be compressed more heavily without a proportional accuracy penalty. This suggests that a homogeneous integer-only scheme, which forces all layers to the same bit-width, is likely suboptimal, and that mixed-precision integer-only approaches could achieve better trade-offs.

To make these trade-offs more concrete, Table 2.3-1 summarises the key accuracy-compression results from the most directly relevant studies.

Table 10.3-1: Summary of accuracy-compression trade-offs reported in selected studies

Study	Model Type	Bit-Width (W/A)	Compression Ratio (approx.)	Accuracy Degradation	Key Enabling Method
Yuan et al. [3]	LLM (undisclosed size)	INT8/INT8 (integer-only)	4× vs FP32	Perplexity ≤ +2%	Per-channel scaling + integer non-linear approx.
Lin et al. [17]	LLM 7B	INT4 weight-only	2.5× vs FP16	Perplexity < +0.5%	Activation-aware weight scaling
Kim et al. [15]	ViT (ImageNet)	INT8/INT8 (integer-only)	4× vs FP32	Top-1 loss ≤ 0.5%	Integer-only softmax + GELU approximation
Rasatavohary [42]	SSM (Mamba, 255M)	1.58-bit ternary	~10× vs FP16	Perplexity +3-5%	Binary/ternary weight representation
Sciammarella [27]	SLM 1B-1.5B	INT4 weight-only	4× vs FP16	Task accuracy ≤ 3%	Standard round-to-nearest

Study	Model Type	Bit-Width (W/A)	Compression Ratio (approx.)	Accuracy Degradation	Key Enabling Method
Chae & Kang [12]	LLM (size undis-closed)	INT8 (FPGA)	Not specified	Perplexity within 1%	FPGA-specific arithmetic design

Note: Compression ratios are approximate and assume FP32 or FP16 baseline as reported in each study. Accuracy degradation metrics are not directly comparable across tasks (perplexity, top-1 accuracy, task accuracy).

When read together, these results point to a clear pattern: for quantisation of LLMs to INT8 or INT4, the accuracy loss is manageable (typically < 3% in task-dependent metrics), provided that the quantization method is tailored to the model's sensitivity profile. Weight-only quantization is more forgiving than activation quantization, and integer-only quantization (which requires both) represents the most challenging regime. Sub-INT4 quantization, while memory-efficient, incurs a substantial accuracy penalty that currently limits its applicability to use cases where quality requirements are relaxed. The trade-offs are not uniform across model scales: smaller models (SLMs) are more vulnerable to degradation than larger ones, but even for SLMs the losses are within acceptable bounds for many edge applications.

2.3.2 Integer-Only Hardware Constraints and Quantization Design

The second major theme that emerges from the synthesis concerns the interaction between quantization design and the specific constraints of integer-only hardware. The literature distinguishes sharply between two deployment scenarios: first, hardware that lacks any floating-point unit (FPU) and therefore requires all arithmetic to be performed in integer arithmetic; and second, hardware that has a limited FPU (e.g., FP16 support only) but for which integer arithmetic is preferred for reasons of energy efficiency or latency. The studies reviewed here address both scenarios,

but the more stringent case--genuine integer-only hardware--is the focus of this thesis and receives correspondingly greater attention.

Yuan et al. [3] present the most directly relevant evidence for the integer-only LLM scenario. Their framework was designed from the ground up to run entirely in integer arithmetic, including the computation of layer-normalization statistics, softmax, and the gating functions used in feed-forward networks. The critical insight from their work is that the nonlinear operations that are standard in LLMs--softmax, GELU, SiLU--cannot be simply replaced by look-up tables on integer-only hardware because the input range is large and the memory cost of a high-resolution LUT is prohibitive. Instead, they propose piecewise polynomial approximations that are evaluated using only integer multiplication and addition. The approximation error introduced by these polynomials is shown to be less than the quantization noise from the weight and activation quantization themselves, meaning that the overall model quality is not dominated by the nonlinearity approximation. This finding is important because it suggests that integer-only deployment is not fundamentally limited by the need to approximate nonlinearities; it merely requires careful design of those approximations.

Complementary evidence comes from Kim et al. [15], who faced a similar challenge for vision transformers. Their IPTQ-ViT method approximates the softmax function using an integer-only exponential function computed via a combination of bit-shifts and table lookup with piecewise linear interpolation. They show that this approximation adds less than 0.1% to the top-1 accuracy loss on ImageNet when the model is quantized to INT8. For the GELU activation, they used a linear approximation over the positive range, which incurs negligible error because GELU is approximately linear for inputs far from zero. The implication for LLMs is direct: both the softmax and the activation functions in transformer models can be implemented in integer arithmetic with minimal accuracy impact, provided the approximation is designed to match the distribution of activations observed in the quantized model.

Hardware-specific studies add another layer of evidence. Chae and Kang [12] implemented an LLM inference accelerator on an FPGA that uses INT8 for all arithmetic operations, including

matrix multiplication and nonlinear functions. Their design uses the fact that FPGAs can be configured to implement arbitrary integer-arithmetic datapaths, allowing them to avoid the overhead of software-based nonlinearity approximation. They report that the FPGA implementation achieves a throughput of 0.5-1.0 tokens per millisecond per watt, which is competitive with some GPU-based deployments when normalized for power consumption. Their results show that the integer-only constraint does not impose a fundamental latency penalty when the hardware is optimised for integer arithmetic; the primary bottleneck becomes memory bandwidth rather than arithmetic capability. This suggests that integer-only edge devices that integrate dedicated integer matrix-multiply units (as many emerging edge AI accelerators do) can achieve inference speeds that are adequate for interactive applications.

The work of Sayyaparaju [5] on Q-Trans takes a slightly different angle by focusing on the fine-tuning stage. Rather than designing a static integer-only inference pipeline, Q-Trans uses quantization-aware fine-tuning (QAT) on the target integer-only hardware, thereby allowing the model to adapt its weights and activation distributions to the specifics of the integer arithmetic. The study reports that models fine-tuned in this way recover 90-95% of the accuracy gap between a naive post-training quantisation and the floating-point baseline. While QAT is computationally expensive--it requires a training run on the target hardware--the evidence from Sayyaparaju suggests that for edge deployment scenarios where the model is fine-tuned once and then deployed on thousands of devices, the cost is justified by the improved accuracy. This finding is consistent with the broader QAT literature and underscores the point that integer-only deployment is not simply a matter of converting weights; the model's behaviour during inference must be accommodated during training or fine-tuning.

The hybrid asymmetric integer-only method proposed by Zhong [4] offers an alternative approach that does not require full QAT. Instead, it applies different quantization schemes to different parts of the network: asymmetric quantization (with separate scale and zero-point for each channel) for weights, and symmetric quantization for activations. The asymmetry allows the quantisation to better match the non-zero-mean distributions typical of weights in neural networks, reducing

the quantization error without the computational overhead of per-layer QAT. Zhong reports that this hybrid scheme reduces the accuracy loss on several benchmark tasks by 30-50% compared to a uniform asymmetric scheme, while still operating entirely in integer arithmetic. This is a practically important result because it illustrates that careful design of the quantization parameters alone--without fine-tuning--can narrow the accuracy gap.

To synthesise these hardware-related findings, Table 2.3-2 summarises the key design choices and their reported impacts.

Table 11.3-2: Integer-only hardware constraints and design responses in the literature

Constraint	Study Response	Reported Impact	Hardware Target
No FPU for nonlinearities	Piecewise polynomial approx. [3]	<0.5% accuracy loss	Generic integer-only
Large input range for softmax	LUT + piecewise linear [15]	<0.1% additional loss	Generic integer-only
Custom integer datapath	FPGA-specific design [12]	0.5-1.0 tok/ms/W	FPGA
Activation distribution mismatch	Asymmetric hybrid quantization [4]	30-50% reduction in loss	Generic integer-only
Accuracy gap after PTQ	QAT fine-tuning [5]	90-95% recovery of loss gap	Edge IoT devices
Memory bandwidth bottleneck	Not an arithmetic issue [12]	Bandwidth dominates latency	FPGA/ASIC

Note: "Loss" refers to the increase in perplexity or task-specific error rate relative to the floating-point baseline. The "hardware target" column indicates the platform on which the method was evaluated.

A notable pattern across these studies is the recognition that the integer-only constraint shifts the design bottleneck from arithmetic to memory access and nonlinear function approximation. For the arithmetic operations themselves--matrix multiplication, additions, bit-shifts--modern

edge hardware is already efficient. The difficulties lie in the non-linearities and in the statistical mismatches between the quantisation grid and the actual weight/activation distributions. The evidence suggests that with appropriate design, both of these challenges can be managed without sacrificing more than a few percentage points of accuracy, and that integer-only inference for LLMs is not a theoretical fantasy but a practical near-term possibility.

2.3.3 Emergent Patterns and Persistent Gaps

The preceding subsections have highlighted individual findings; this final subsection steps back to identify cross-cutting patterns that recur across the literature and to articulate the gaps that remain open.

2.3.3.1 The Recurring Centrality of Activation Quantization One of the clearest patterns is that activation quantization is consistently more challenging than weight quantization. Every study that reports results for both weight-only and activation-quantized inference shows a larger accuracy degradation when activations are quantized to integer formats. This pattern is evident in the LLM studies [3][17], the ViT study [15], and even in the RNN work of Sari et al. [16], which found that integer-only RNNs suffer disproportionately from activation quantisation compared to weight-only schemes. The reason is well understood: activations vary dynamically with each input, and their distributions are often multimodal or heavy-tailed (especially after nonlinearities like GELU), making them harder to quantise without a large calibration set or adaptive scaling. Weight distributions, by contrast, are static after training and can be carefully calibrated offline.

This pattern has direct implications for the integer-only edge deployment of LLMs. If the hardware supports only integer arithmetic, activations must be quantized, and the 2-5% accuracy loss that the literature reports for activation quantization becomes the dominant factor. Strategies that mitigate this loss--such as per-token scaling, channel-wise quantization, and calibration-data selection--are therefore the most critical components of any integer-only deployment pipeline. The

evidence suggests that these strategies are effective but that they add engineering complexity; off-the-shelf quantization toolkits may not implement them.

2.3.3.2 The Importance of Calibration Data and Quantization Ranges A second pattern concerns the heavy dependence of quantization accuracy on the calibration dataset and the method used to determine the quantization range (the minimum and maximum values for weights or activations). Lin et al. [17] explicitly show that the choice of calibration data--whether it is representative of the target deployment distribution--can appreciably change the perplexity degradation for a given quantization scheme. Zhong [4] also reports that his hybrid asymmetric method is sensitive to the calibration set size: collecting 1,024 samples yields a 30% reduction in accuracy loss compared to using only 128 samples, but increasing beyond 2,048 yields diminishing returns. This finding has practical implications: edge deployment scenarios often lack access to large, representative calibration sets because the target data distribution may not be fully known at development time. The literature provides guidance--at least 1,000 samples appear to be needed for stable calibration--but no study has yet systematically addressed the scenario in which the calibration distribution differs substantially from the deployment distribution (a form of domain shift). This is a gap that future work should address.

2.3.3.3 The Mixed-Precision Imperative A third pattern is that homogeneous integer-only quantization--applying the same bit-width to all layers--appears to be suboptimal compared to mixed-precision approaches. The MPQ-YOLO study [7] and the general mixed-precision literature (most of which is not included in this review) provide a strong theoretical and empirical argument that different layers and even different channels within a layer have different sensitivity to quantization. The core evidence from the present review comes from the sensitivity analysis embedded in the AWQ method [17], which identifies the most sensitive weight channels and preserves them at higher precision. Although AWQ is a weight-only method, the same principle should apply to activation quantization. The implication for integer-only deployment is that the hardware should ideally support multiple integer precisions (e.g., INT8, INT4, maybe INT2) to

enable per-layer or per-channel mixed precision. Many current edge AI accelerators, however, support only a single integer precision (typically INT8), which forces a one-size-fits-all approach and leaves accuracy on the table.

2.3.3.4 Persistent Gaps Despite the progress reflected in the studies reviewed, several gaps remain open. First, there is a striking lack of standardised benchmarks for evaluating quantized LLMs on integer-only edge hardware. The studies use different model sizes, different metrics (perplexity, task accuracy, latency, energy), and different hardware platforms, making cross-study comparisons difficult. A community-wide benchmark--analogous to the GLUE or HELM benchmarks for LLM quality, but extended to include integer-only deployment constraints--would accelerate progress.

Second, the evidence for large LLMs (e.g., 13B-70B parameters) is thin. The integer-only framework of Yuan et al. [3] was applied to an undisclosed model size; AWQ was tested on 7B models. For models beyond 13B, the outlier activation problem becomes more severe, and the integer-only methods have not yet been validated. This is a critical gap because many real-world edge deployments might benefit from a 13B model rather than a 7B or 1B model, especially for tasks requiring deep reasoning.

Third, the energy-accuracy trade-off is underexplored. The work of Zhang [14] on weight-only quantisation across NVIDIA GPUs shows that reducing precision does not always save energy because of runtime quantisation overhead. For integer-only edge hardware, the situation is different: if the hardware is designed to compute only in integer arithmetic, the energy per operation is inherently lower than for floating-point, but the overhead of data movement may dominate. None of the reviewed studies provides a detailed energy breakdown for integer-only LLM inference on edge devices. Such data are needed to make informed decisions about whether integer-only deployment is worth the engineering effort.

Fourth, the interaction between quantisation and other compression techniques--pruning, distillation, low-rank factorisation--has been touched on by only a few studies. Ameen et al. [50] combine pruning and quantisation for CNNs on Raspberry Pi, and the cited work of Novkin et al.

[11] does so for GNNs, but no study in this review applies a truly combined pipeline to LLMs for integer-only edge deployment. The potential synergy is notable: pruning removes the less important weights, which might be exactly those that are hardest to quantise; knowledge distillation can transfer the behaviour of a larger model to a smaller one that is more amenable to quantisation. The absence of such studies is a clear gap.

Finally, there is a methodological gap in the evaluation of quantized models on edge hardware. Almost all of the reviewed studies measure accuracy using standard benchmarks (perplexity on Wikipedia, accuracy on GLUE tasks), but these benchmarks do not capture the full range of behaviours that matter for interactive edge applications--such as the model's robustness to adversarial inputs, its tendency to hallucinate, or the consistency of its responses under temperature sampling. The evaluation framework for quantized edge models needs to extend beyond static benchmarks to include qualitative and reliability assessments.

2.3.3.5 Synthesis Summary Taken together, the findings from the literature paint a picture of cautious optimism. Integer-only quantization for LLMs is both possible and practical, provided that the method accounts for activation outlier sensitivity, uses appropriate calibration data, and optionally employs mixed-precision or fine-tuning to recover accuracy. The evidence points to INT8 as the sweet spot for integer-only deployment of models up to 7B parameters, with INT4 weight-only being a viable alternative when the hardware supports separate weight and activation formats. For larger models, more research is needed. The persistent gaps--lack of standardised benchmarks, limited coverage of large models, insufficient energy analysis, and the missing connection to other compression techniques--define the frontier for future work.

This synthesis affirms the gap identified in Section 2.1.5: while the fundamental building blocks of integer-only LLM deployment exist, no study provides a complete, validated pipeline that yields production-ready accuracy for a broad range of LLM sizes and task types on representative integer-only edge hardware. The thesis that follows, therefore, cannot close all of these gaps, but it

can contribute by synthesising the existing evidence into a coherent framework and by identifying the most promising directions for empirical validation.

2.4 Discussion

The findings from the literature synthesized in section 2.3 reveal several marked insights that both align with and extend the theoretical frameworks discussed in section 2.1. While the fundamental principles of quantization--mapping continuous values to discrete levels, choosing symmetric versus asymmetric ranges, and selecting calibration data--are well established, the application of these principles to large language models running on integer-only edge hardware introduces constraints that existing theory does not fully anticipate. As noted in the literature review (section 2.1), earlier quantization work for convolutional neural networks often assumed uniform sensitivity across layers and activations. The evidence from recent studies, however, paints a more complex picture.

The most striking pattern across the synthesized research is that activation outliers--those extreme values that appear in intermediate feature maps--constitute the primary bottleneck for integer-only LLM deployment. This observation, discussed in section 2.1 as a side note in the context of mixed-precision experiments, now emerges as a central theme. Activation-aware quantization schemes such as AWQ [17] explicitly address this issue by preserving the high-precision representation of a small subset of critically important weights and activations. One might expect that any sophisticated clipping or per-channel scaling strategy would suffice, but the data suggests that the structural distribution of outliers in LLMs (often concentrated in specific attention heads or token positions) demands targeted solutions rather than generic calibration.

2.4.1 Interpretation of Synthesised Findings

The literature consistently indicates that integer-only precision down to INT8 is achievable for LLMs up to approximately 7 billion parameters without catastrophic accuracy loss, provided that the quantization method accounts for activation outlier sensitivity. This is a meaningful result,

because many edge scenarios--on-device chatbots, real-time text assistants, or privacy-preserving inference in a local network--operate with models in the 1B-7B range. As discussed in section 2.3, Yuan and colleagues [3] indicated a complete integer-only pipeline for LLM deployment that achieved perplexity within 2-5% of the FP16 baseline across multiple benchmarks. Similarly, the activation-aware methods described by Lin et al. [17] showed that focusing calibration on outlier-affected channels substantially reduced accuracy drop compared to uniform quantization.

What stands out here is that these methods share a common design pattern: they introduce an extra pre-processing step that identifies and compensates for outlier activations before rounding to integer formats. The activation-aware weight quantization technique [17] scales the weights of outlier channels to align with the quantization grid, effectively reducing the error in those critical dimensions. Other studies, such as the integer-only framework proposed at ISCAS 2025 [3], use per-tensor dynamic range statistics collected from a small calibration dataset to choose clipping thresholds on a per-layer basis. The hybrid asymmetric approach of Zhong [4] takes this further by allowing different quantization parameters for positive and negative ranges within the same tensor, which can be especially beneficial when outlier distributions are skewed.

The numbers are telling. For models up to 7B parameters, INT8 weight-only quantization retains nearly all of the original model's task performance across generation and classification tasks, as reported by multiple sources [5][12][27]. When activations are also quantized to INT8, the degradation remains acceptable (typically <5% perplexity increase) provided that outlier-aware calibration is used. This is consistent with earlier findings for smaller transformer models, but extends the scale boundary substantially--a point that was not fully anticipated by the literature reviewed in section 2.1, which largely focused on CNNs and smaller BERT-scale models.

Beyond INT8, the evidence for INT4 is less uniform. Some studies [27] show that INT4 weight-only quantization can substantially reduce memory footprint while maintaining acceptable quality for small language models (1-1.5B parameters) on CPU-only cloud infrastructure. However, the practical deployment of full INT4 integer-only inference remains challenging. The work by Zhang [14] carries a cautionary message: weight-only quantization does not always save en-

ergy, because runtime de-quantization to FP16 for matrix multiplication introduces overhead that can offset memory savings, especially on architectures without native INT4 ALUs. This finding resonates with the hardware-awareness gap identified in section 2.1.5--the assumption that lower bit-width automatically yields lower energy is invalid without considering the target compute unit.

2.4.1.1 Implications for Edge Hardware Design The interplay between quantization algorithm and hardware architecture emerges as a critical dimension. As discussed in section 2.1, the ideal scenario for integer-only deployment is a processor that directly executes INT8 or INT4 MAC operations. Chae and Kang [12] implemented a full FPGA-based LLM accelerator that uses integer arithmetic throughout, achieving real-time inference for a quantized GPT-2 model. Their results suggest that when the hardware is designed for the precision, there is no energy penalty from format conversions. However, for general-purpose edge CPUs or microcontrollers that lack native low-precision integer support, the literature indicates that the effective speedup from quantization may be far smaller than the bit-width reduction would imply [14].

This tension leads to a practical recommendation: before committing to a quantization strategy, one must profile the target hardware's integer arithmetic capabilities. The findings from Nisiotis and Markov [13] on quantised SLMs for conversational agents in virtual worlds support this view--they achieved sub-100ms inference latency on a modest edge device only after selecting a quantization scheme that matched the hardware's instruction set. Similarly, the integer-only RNN work by Sari et al. [16] provides a precedent: by redesigning the recursive computation to use only integer arithmetic, they avoided any floating-point dependencies, but the approach required careful scheduling of accumulation to prevent overflow.

2.4.2 Comparison with Theoretical Frameworks

The theoretical foundations outlined in section 2.1--specifically the trade-off between bit-width, model size, and accuracy--are largely confirmed by the synthesized evidence. The quantization-error bound analysis from early CNN quantization work (reviewed in section 2.1)

predicted that careful outlier management could reduce the effective error per layer. Recent LLM studies [3][17] establishes exactly that: by aligning quantization grids with activation distributions, the actual error on downstream tasks is lower than a naive uniform-quantization bound would suggest.

However, the synthesized findings also challenge a key assumption implicit in the literature review: that fine-tuning (quantization-aware training) is necessary to recover full accuracy. Several studies [5][15] show that post-training quantization with proper calibration can achieve results within 1-2% of QAT for many tasks, especially when the model is large enough to exhibit some redundancy. This is an unexpected but welcome result, because QAT requires access to the original training pipeline, which is often unavailable for third-party LLMs. The literature suggests that for integer-only edge deployment, PTQ combined with outlier-aware calibration is a viable first-line strategy, with QAT reserved for stress-test scenarios or applications with strict accuracy requirements.

The comparison between theory and evidence is neatly illustrated by the activation outlier phenomenon. Standard quantization theory (section 2.1.1) assumed that the quantization error depends on the range and granularity of the value distribution. It did not fully account for the long-tailed structure of LLM activations, where a small fraction of channels carry values 10-100× larger than the median. The work on activation-aware quantization [17] effectively extends the theory by showing that clipping or re-scaling those specific channels yields disproportionate gains. This suggests that future quantisation theory should incorporate per-channel sensitivity metrics as a first-class parameter, rather than treating them as an empirical afterthought.

2.4.3 Addressing Research Gaps

The research gaps identified in section 2.1.5--specifically the lack of standardised benchmarks for integer-only edge deployment, the limited coverage of large models (>7B parameters), the insufficient energy analysis, and the missing integration with other compression techniques--are only partially addressed by the existing literature.

Table 12: Summary of Research Gaps and Corresponding Evidence from Literature

Research Gap	Evidence from Literature	Degree of Addressal	Remaining Challenges
Standardised edge deployment benchmarks	Few cross-study comparisons; individual works use different datasets [3][17]	Low	Need common metric set (latency, energy, accuracy) on representative edge hardware
Coverage of large models (>7B)	Most studies test ≤7B; only weight-only INT4 explored for larger [12]	Low	Full integer-only pipeline for 13B+ models not demonstrated
Energy consumption analysis	Only Zhang [14] directly measures energy; others use proxy metrics	low	Few studies measure on real edge hardware; energy models needed
Integration with sparsity, pruning, adapters	Combined quantization+pruning studied only for CNNs [50]	Minimal	Joint optimisation for LLMs not explored in depth
Robustness and reliability of quantised models	No studies evaluate hallucination rate or adversarial robustness post-quantization	None	Field is wide open

Table 13: Mapping of identified research gaps to literature coverage. Source: compiled from [3][5][12][14][17][27][50].

The gap that has seen the most progress is the development of integer-only pipelines. Yuan et al. [3] and Lin et al. [17] both describe complete frameworks that operate entirely in integer arithmetic after calibration. These frameworks directly address the core challenge identified in section 2.1: can LLMs be served on integer-only hardware without floating-point dependencies? The answer from the literature is a qualified yes--provided the model is not too large and the activation outliers are handled.

Nevertheless, the gap around standardised benchmarking remains wide. Each study uses a different combination of tasks (perplexity, GLUE, MMLU, custom generation metrics) and different edge-hardware proxies (Raspberry Pi, FPGA, Jetson Nano, simulated CPU). This fragmentation makes it difficult to compare results across works. The evaluation literature discussed in section 2.1.4 (e.g., [23][24]) touches on the complexity of LLM evaluation but does not provide a specific benchmark suite for quantised edge models. One might expect that the field would converge on a small set of representative tasks (e.g. summarisation, question answering, code generation) plus hardware metrics (latency per token, peak memory, energy per query), but no such consensus exists yet.

2.4.4 Theoretical and Practical Implications

The findings from the literature carry meaningful implications for both the theory of neural network quantisation and the practical engineering of edge-deployed LLMs.

2.4.4.1 Theoretical Implications First, the centrality of activation outliers forces a re-examination of the uniform-random-error model that underpins much of classical quantisation theory. When a small number of channels dominate the total range, the worst-case error is decoupled from the average error--a phenomenon that can be exploited by per-channel or per-token scaling. The mathematical framework for certification of quantised networks (e.g., [32]) provides tools to bound the impact of quantisation on robustness, but those bounds become loose if outlier channel behaviour is not explicitly modelled. Future theoretical work should develop tighter bounds that incorporate the structured distribution of outliers.

Second, the disconnect between bit-width and actual energy consumption, highlighted by Zhang [14], calls for a more nuanced cost model. Simple models that assume linear scaling with bit-width are insufficient; a realistic energy model must account for the cost of format conversion, memory bandwidth for different word sizes, and the efficiency of the MAC unit at each precision.

Theoretical frameworks that incorporate the hardware instruction set as a parameter (rather than assuming a uniform multiplier) would better guide algorithm designers.

2.4.4.2 Practical Implications On the practical side, the findings offer clear guidance for engineers deploying LLMs on integer-only edge devices. For a device with hardware INT8 support, the recommended approach is to use activation-aware static quantization, calibrating on a representative sample of the target domain and preserving outlier channels through per-channel scaling or selective high-precision retention. This approach, validated across multiple sources [3][17], yields a solid integer-only model with minimal accuracy loss.

For devices without native INT8 (e.g. older ARM Cortex-A cores that only support INT32 or FP16), weight-only quantization to INT4 combined with mixed-precision for activations may be a reasonable compromise, but the engineer must measure the actual latency and energy on the target hardware--the assumption of automatic energy savings does not hold [14]. The FPGA work of Chae and Kang [12] shows that a custom integer-only datapath can achieve near-ideal efficiency, but this requires specialised silicon, which is not available in most off-the-shelf edge devices.

A further practical implication concerns the choice of calibration data. The literature shows that using a small, domain-matched calibration set (a few hundred to a few thousand samples) is sufficient for effective outlier detection [3][17]. However, if the calibration set is mismatched from the deployment distribution, the outlier structure may change and cause accuracy degradation. This underscores the need for domain-aware calibration procedures.

2.4.5 Limitations of Existing Literature

While the synthesized evidence provides a solid foundation, several limitations must be acknowledged. First, the majority of integer-only studies focus on relatively small models (≤7B parameters). For larger LLMs (13B, 70B, or beyond), the activation outlier problem may become more severe because the model depth amplifies the propagation of outlier values. The literature

offers no conclusive evidence on whether the current techniques scale beyond 7B for full integer-only inference.

Second, the evaluation metrics used in the existing studies are predominantly task-specific (perplexity, accuracy on a single benchmark). Few works examine the broader reliability of quantised models--their susceptibility to adversarial attacks, their tendency to hallucinate, or the consistency of their outputs under multiple queries. As discussed in section 2.1.4, the LLM evaluation literature [23][24] highlights that metrics like perplexity are weak proxies for real-world quality. This limitation means that the true cost of quantization in interactive edge applications may be higher than the static benchmarks suggest.

Third, the energy and latency measurements are often performed in simulation or on development boards that are several generations behind commercial edge hardware. The findings from Zhang [14] are a rare exception, and they reveal that simulation-based estimates can be misleading. Without a standardised hardware testbed, the field lacks credible, reproducible energy claims.

Fourth, the integration of quantization with other compression techniques (pruning, knowledge distillation, approximate multipliers) is almost entirely absent for LLMs. For CNNs, studies like [50] successfully combine pruning and quantization on a Raspberry Pi, achieving a large total model size reduction with minimal accuracy loss. Whether similar compound gains are achievable for LLMs remains an open question.

2.4.6 Future Research Directions

Based on the gaps and limitations discussed above, several future directions emerge from the literature.

Standardised benchmark suite for integer-only edge LLMs. The most pressing need is a common evaluation framework that includes both task-level accuracy and hardware-level metrics (latency per token, peak memory, energy per query) across a representative set of edge platforms. Such a benchmark would accelerate progress by enabling direct comparison across methods and revealing the true Pareto frontier of accuracy versus efficiency.

Scalability to larger models. The literature should extend integer-only studies to model sizes up to 70B parameters. The key research question is whether activation-outlier compensation techniques that work at 7B are sufficient at larger scales, or whether fundamentally new approaches (e.g. mixed-precision by module or dynamic per-token quantization) become necessary.

Energy-aware quantization framework. Building on the findings of Zhang [14], a research direction that jointly optimises quantization bit-width and target-hardware instruction-set characteristics could yield substantial practical gains. This is essentially a hardware-software co-design problem, and it is currently underexplored.

Reliability and safety of quantised edge models. The field needs studies that evaluate the robustness of quantised LLMs against adversarial inputs, distribution shift, and sampling variability, especially for safety-critical edge applications. The certified-robustness framework from integer-arithmetic-only networks [32] could be extended to LLMs.

Integration with other compression techniques. Exploring combined quantization + sparsity + pruning for LLMs on edge devices could unlock further reductions in model size and energy consumption. The CNN precedent [50] suggests that compound compression can exceed the additive benefits of individual techniques.

Domain-specific calibration and finetuning. Given the sensitivity of quantization to calibration data distribution, future work should investigate adaptive calibration strategies that allow the quantized model to be refined with a small amount of target-domain data. Quantization-aware fine-tuning methods [5] provide a starting point, but they require access to model parameters and training infrastructure that may not be available in every deployment scenario.

Table 14: Summary of Future Research Directions and Expected Impact

Research Direction	Key Question	Expected Impact	Priority
Standardised benchmark suite	What is the common metric set for edge LLM deployment?	Enables reproducible comparisons across methods	High

Research Direction	Key Question	Expected Impact	Priority
Scaling to 70B models	Do existing outlier methods transfer to larger scales?	Determines upper limit of integer-only deployment	High
Energy-aware quantisation	How to jointly optimise bit-width and hardware ISA?	Reduces real energy consumption, not just theoretical	High
Robustness evaluation	Do quantised models hallucinate more?	Ensures safety in interactive applications	Medium
Combined compression	Can pruning + quantisation compound for LLMs?	Enables larger models on smaller devices	Medium
Adaptive calibration	Can quantised models be adapted to new domains with little data?	Increases practical deployability	Medium

Table 15: Priority-ranked future research directions derived from the synthesis and gap analysis. Sources informing each direction are cited in the surrounding text.

2.4.7 Concluding Remarks

The discussion above makes clear that integer-only quantization for LLMs on edge hardware is no longer a theoretical curiosity--several complete pipelines exist and have been validated on models up to 7B parameters. The evidence points toward activation-outlier-aware calibration as the key enabler, with per-channel scaling and selective high-precision retention being the most effective tactics. The literature also confirms that the assumption of automatic energy savings from lower bit-width is flawed, and that hardware-specific profiling is essential.

Nevertheless, the field remains fragmented. The absence of standardised benchmarks, the limited coverage of large models, and the near-total neglect of reliability metrics mean that the path from research prototype to production-grade edge LLM is not yet fully mapped. The synthesis presented in this thesis (section 2.3) and discussed here thus offers a snapshot of a rapidly evolving environment--where the building blocks are in place, but the integrated, validated pipeline that

spans algorithm, calibration, hardware, and deployment remains a goal for future work rather than a present achievement.

3. Conclusion

The preceding chapters have mapped out the technical environment, the open debates, and the unresolved challenges that define quantization for large language models on integer-only edge hardware. This final section pulls together the principal findings, acknowledges the limitations of both the existing literature and the present review, and outlines a set of directions that need sustained empirical attention.

3.1 Summary of Key Findings

Look at the literature examined here and what you see is a field moving fast--but unevenly. The basic mathematical framework of quantization--uniform affine mapping, scale factor and zero-point calibration, the trade-off between bit-width and representational fidelity--is well established. What stays unresolved is how these general principles translate to models at the scale and architectural complexity of contemporary LLMs, especially under the hard constraint of integer-only arithmetic.

3.1.1 The Primacy of Calibration Strategy A consistent finding across the surveyed work is that the choice of calibration method exerts a disproportionate influence on final model quality when no floating-point fallback is available. Post-training quantization (PTQ) approaches that lean on simple min-max or percentile-based calibration preserve reasonable fidelity for weights but degrade sharply on activations, particularly for outlier-prone channels that appear in modern Transformer architectures. Activation-aware methods--like those explored by Yuan et al. [3] and Lin et al. [17]--show that selective per-channel scaling, informed by the distribution of activation magnitudes, can recover a substantial fraction of the accuracy lost under uniform schemes. The message is straightforward: integer-only deployment cannot succeed without calibration strategies that are sensitive to the heterogeneous statistical properties of LLM activations.

3.1.2 The Limitations of Weight-Only Quantization A second insight concerns the limited utility of weight-only quantization when the target hardware lacks floating-point units. Weight-only schemes cut memory footprint and can speed up memory-bound inference on GPU platforms, but as Zhang [14] has shown empirically, the assumed energy savings don't always materialise--and on integer-only edge devices the benefit is further squeezed because activation quantisation stays a bottleneck. What the literature suggests is that for truly edge-native deployment, both weights and activations must be quantised, and the entire compute graph must execute in integer arithmetic. Partial approaches that handle only one side of the equation leave performance on the table or, worse, introduce runtime overheads from repeated dequantisation and requantisation cycles.

3.1.3 The Promise and Cost of QAT Quantisation-aware training (QAT) offers the most reliable path to high-fidelity integer-only inference, but it comes with a steep practical price. Sayyaparaju [5] and others have shown that fine-tuning a pre-trained LLM with simulated quantisation in the forward pass can recover near-floating-point accuracy at INT8 and, in some cases, at INT4 precision. The cost, however, is computational: QAT requires access to representative training data, extra GPU cycles, and careful tuning of the quantisation schedule. For many edge deployment scenarios--especially those with small teams or limited budgets--the resources required for QAT may rival those needed for the original model training. This creates a tension between model quality and practical accessibility that the literature has not yet resolved.

Key Finding	Supporting Evidence	Practical Implication
Activation-aware calibration outperforms uniform schemes for integer-only LLM inference	[3][17]	Calibration strategy is a primary design choice; one-size-fits-all approaches are inadequate
Weight-only quantisation provides limited benefit on integer-only edge hardware	[14]	Both weights and activations must be quantised for edge-native deployment

Key Finding	Supporting Evidence	Practical Implication
QAT recovers near-floating-point accuracy but incurs substantial computational cost	[5]	Resource-constrained teams may need to accept lower fidelity or seek alternative approaches
Hardware-algorithm co-design can offload non-linear operations to dedicated integer logic	[12][15]	Future edge accelerators should standardise support for activation functions and normalisation layers
Small language models (SLMs) are more forgiving of aggressive quantisation than larger LLMs	[13][27]	Model size and quantisation strategy must be chosen jointly; smaller models may be preferable for integer-only hardware

Table 16: Summary of principal findings from the literature on integer-only quantisation of large language models for edge deployment.

3.1.4 The Architectural Gap: Non-Linear Operations in Integer Arithmetic One of the most stubborn obstacles to integer-only inference is the handling of non-linear operations that are native to floating-point arithmetic but have no direct integer analogue. Layer normalisation, softmax, SiLU and GELU activations, even simple division--these operations recur throughout every modern LLM, yet none maps cleanly to the integer arithmetic unit of a typical edge processor. The literature approaches this problem in two ways. The first is to approximate these operations with integer polynomial or lookup-table substitutes, an approach shown by Kim et al. [15] for vision transformers and applicable by extension to LLM architectures. The second is to redesign the hardware itself, as Chae and Kang [12] propose with FPGA-based accelerators that include dedicated integer logic for softmax and layer normalisation. Both paths are viable, but neither has achieved the maturity or standardisation needed for widespread adoption.

3.2 Contributions of This Thesis, the most useful thing this thesis does is bring together a fragmented literature, organising existing work along the axes of calibration method, quantisation granularity, training regime, and hardware target. It also identifies a set of persistent research gaps--listed below--that have received insufficient attention in the published record, and offers a critical assessment of the practical trade-offs that practitioners face when choosing among PTQ, QAT, mixed-precision, and hardware-algorithm co-design approaches, with particular attention to the constraints of integer-only arithmetic.

This review shows that the field is at a transitional moment. The fundamental techniques exist; what is missing is the engineering maturity, the standardised benchmarks, and the cross-disciplinary collaboration that would turn these techniques into reliable deployment recipes.

3.3 Limitations

The conclusions drawn in this thesis must be interpreted given several limitations. The most important concerns the scope of the literature reviewed. The field evolves at an extraordinary pace--new quantisation methods appear on preprint servers weekly--and any review is necessarily a snapshot of a moving target. The selection of sources, while guided by systematic search criteria, reflects the biases of the search terms, databases, and time window employed.

A second limitation is methodological. Because this is a narrative review rather than a meta-analysis, it does not provide quantitative effect-size estimates or formal comparisons of reported accuracy metrics across studies. The heterogeneity of evaluation protocols--different benchmarks, different model families, different bit-widths, different hardware backends--makes direct numerical comparison unreliable. The patterns identified here are qualitative and should be tested in controlled experimental settings.

A third limitation concerns the hardware focus. The review emphasises integer-only arithmetic as a constraint, but the specific characteristics of edge hardware--memory bandwidth, cache hierarchy, power budget, on-chip SRAM size--vary enormously across device classes. A strategy

that works on an FPGA may not transfer to a microcontroller, and vice versa. So the conclusions of this thesis are general at the level of principle but may require site-specific adaptation.

Limitation	Description	Mitigation in This Thesis
Rapid field evolution	New methods appear continuously, making any review a snapshot	Explicit date range stated; focus on established themes rather than individual results
Narrative review design	No quantitative meta-analysis or pooled effect sizes	Qualitative synthesis with thematic organisation; claims hedged appropriately
Hardware heterogeneity	Edge device characteristics vary widely	Focus maintained on integer-only constraint as a common denominator
Evaluation protocol diversity	No standard benchmark for integer-only LLM inference	Gaps identified point toward the need for standardised evaluation

Table 17: Principal limitations of the present review and the steps taken to address each.

3.4 Future Research Directions

The gaps identified in the literature suggest several avenues for future work. One priority is the development of standardised evaluation benchmarks for integer-only LLM inference on edge hardware. Without a common reference--a set of models, calibration datasets, target bit-widths, and hardware platforms--comparing methods will remain a matter of expert judgment rather than reproducible science. Initiatives similar to the MLPerf inference benchmark, but targeted at integer-only edge deployment, would accelerate progress notably.

A second direction concerns the expansion of hardware-algorithm co-design. The literature reviewed here contains promising examples of FPGA-based accelerators and custom integer arithmetic units [12], but these remain isolated proof-of-concept demonstrations. What is needed

is a systematic design space exploration that maps the accuracy-hardware cost frontier across bit-widths, architectural variants, and approximation strategies for non-linear operations.

A third priority is the investigation of quantisation jointly with other compression techniques. Pruning, knowledge distillation, and architecture search interact with quantisation in ways that the current literature has only begun to explore. A model that is aggressively pruned may respond differently to quantisation than its dense counterpart, and the optimal strategy may involve coordinating all three techniques rather than treating them independently.

Finally, the literature would benefit from more studies that directly compare PTQ and QAT under strictly identical conditions--same model, same hardware, same calibration data, same evaluation protocol. The current evidence base is composed of studies that each use different experimental configurations, making it difficult to attribute performance differences to the quantisation method itself rather than to confounding factors.

3.5 Concluding Remarks

Deploying large language models on integer-only edge hardware is not a single problem but a family of problems, each defined by the interplay of model architecture, bit-width, calibration strategy, hardware capability, and deployment budget. The literature has made meaningful progress: activation-aware calibration, hardware-algorithm co-design, and quantisation-aware training have each shown that integer-only inference is feasible at practical quality levels for models of moderate size. Yet the field remains immature. Standardised benchmarks are absent. The engineering tooling for integer-only deployment is fragmented. The interaction of quantisation with other compression techniques is poorly understood.

What the evidence points toward is a contingent, context-dependent picture. No single quantisation strategy dominates across all models and all hardware targets. The best choice depends on whether the deployment scenario prioritises accuracy, latency, energy efficiency, or ease of implementation--and in practice, these priorities often conflict.

The numbers are worth restating. A model that retains above 95 percent of its floating-point accuracy after INT8 quantisation can be deployed on a microcontroller-class device with a memory budget measured in megabytes rather than gigabytes. That is not a marginal improvement--it changes what is possible in edge AI. But achieving that result reliably requires careful calibration, appropriate hardware support, and a willingness to accept that the floating-point baseline may not be fully recoverable. The field is closer to that goal than it was three years ago, but the final stretch--turning promising techniques into stable, reproducible, and accessible deployment workflows--remains ahead.

4. Appendices

Appendix A: Conceptual Framework for Integer-Only Quantization

A.1 Overview

The following conceptual framework synthesises the key design dimensions that emerge from the literature on integer-only quantization for large language models. Rather than proposing a novel taxonomy, the framework organises the technical choices and trade-offs reported across the studies reviewed in this thesis, providing a structured lens for evaluating and comparing quantization approaches.

A.2 Core Dimensions of the Framework

The framework identifies four principal dimensions that any integer-only quantization pipeline must address: representation scheme, calibration strategy, granularity of quantization, and arithmetic handling. Each dimension contains a set of design choices that interact with one another, producing distinct accuracy-efficiency profiles.

Dimension	Key Design Choices	Reported Trade-Offs
Representation	Uniform affine, asymmetric, hybrid asymmetric	Uniform is simpler; asymmetric captures skewed distributions better
Calibration	Min-max, percentile, entropy-based, clipping	Percentile clipping reduces outlier impact; entropy preserves distributional fidelity
Granularity	Per-tensor, per-channel, per-group, mixed-precision	Finer granularity improves accuracy but increases overhead
Arithmetic Handling	Full integer-only, partial float fallback, lookup tables	Integer-only avoids FPU dependency; float fallback sacrifices hardware compatibility

Table A.1: Core dimensions of integer-only quantization frameworks, synthesised from the reviewed literature.

The representation dimension concerns how real-valued numbers are mapped to the target integer grid. Uniform affine quantization, expressed through a scale factor and zero-point, remains the most widely adopted scheme due to its straightforward hardware implementation. Yuan [3] reveals that uniform affine mapping can be applied effectively to transformer-based LLMs for edge deployment, achieving functional models without resorting to floating-point fallback. Zhong [4] extends this direction by proposing a hybrid asymmetric scheme that applies different zero-point adjustments for weights and activations, reporting improved fidelity on skewed activation distributions commonly found in transformer layers.

Calibration strategy determines the range boundaries used to compute the scale factor and zero-point. Min-max calibration, which uses the extreme observed values, is simple but sensitive to outliers. Percentile clipping, where a small fraction of extreme values is discarded, offers a more strong alternative. Yamaguchi [1] report that percentile-based clipping combined with per-layer quantization reduces the bit storage required for a vision transformer by 86% while maintaining task accuracy. For LLMs specifically, the choice of calibration set -- typically a few hundred to a few thousand text samples drawn from the training distribution -- can substantially affect the quality of the resulting quantized model.

Granularity refers to whether a single scale factor and zero-point are shared across an entire tensor, computed per output channel, or applied to smaller groups of parameters. Per-tensor quantization is the most efficient in terms of hardware cost but can suffer when weight or activation distributions vary considerably across channels. Per-channel and per-group schemes improve accuracy at the cost of additional storage for scale factors and more complex hardware data paths. The literature on LLMs suggests that per-channel quantization of weights combined with per-tensor quantization of activations offers a practical compromise for integer-only deployment.

Arithmetic handling is perhaps the most consequential dimension for edge hardware lacking floating-point units. Full integer-only quantization requires that every operation in the model --

including non-linear functions such as softmax, layer normalization, and the SiLU activation -- be approximated using integer arithmetic. Kim [15] address this challenge for vision transformers by proposing integer-only approximations of non-linear functions, achieving competitive accuracy on ImageNet classification. Sari [16] similarly indicates integer-only inference for recurrent neural networks, showing that non-linearities can be implemented through lookup tables and piecewise polynomial approximations without resorting to floating-point computation.

A.3 Interaction Effects Between Dimensions

The dimensions described above do not operate independently. The choice of calibration strategy, for instance, directly influences whether a per-tensor or per-channel granularity is sufficient. When percentile clipping successfully suppresses outliers, per-tensor quantization can often match the accuracy of per-channel schemes. Conversely, when calibration is performed using min-max ranges, per-channel quantization becomes more necessary to compensate for the wider ranges introduced by extreme values.

Similarly, the representation scheme and the arithmetic handling dimension are tightly coupled. Asymmetric quantization, which introduces a non-zero zero-point, adds an extra integer subtraction operation to each arithmetic step. While this is straightforward to implement in hardware, it increases both the computational cost and the complexity of hardware data paths. Lin [32] explore the implications of integer-only arithmetic for certified robustness, noting that the additional operations introduced by asymmetric schemes can complicate formal verification.

Interaction Pair	Observed Effect	Relevant Source
Calibration × Granularity	Percentile clipping reduces the need for fine granularity	[1]
Representation × Arithmetic	Asymmetric schemes increase hardware complexity	[32]
Granularity × Arithmetic Handling	Finer granularity multiplies scale factor storage and compute	[3]

Table A.2: Documented interaction effects between quantization dimensions.

A.4 Applying the Framework to Integer-Only Hardware

Integer-only edge hardware, such as ARM Cortex-M series microcontrollers, custom ASICs designed for INT8 arithmetic, and certain FPGA configurations, imposes a strict constraint: no floating-point operations may be used during inference. Under this constraint, the framework suggests a preferred combination of design choices.

First, the representation scheme should be uniform affine with careful handling of the zero-point to avoid additional computational overhead. Second, calibration should employ percentile clipping with a calibration set of at least 512 text samples to ensure representative range estimation. Third, weights should be quantized per-channel to preserve accuracy, while activations can be quantized per-tensor given the stabilizing effect of layer normalization in transformer architectures. Fourth, all non-linear functions should be replaced with integer-only approximations -- either lookup tables with linear interpolation or piecewise polynomial evaluations -- eliminating any dependency on floating-point arithmetic.

A.5 Limitations of the Framework

The framework presented here is a synthesis of findings from across several studies that employ different models, hardware targets, and evaluation protocols. Direct comparisons between studies are complicated by these differences, and the interaction effects reported in Table A.2 should be treated as suggestive rather than definitive. Further empirical work is needed to quantify the magnitude of each interaction under controlled conditions.

Appendix B: Supplementary Data Tables

B.1 Comparative Summary of Quantization Approaches

The following tables provide supplementary data that contextualise the analysis presented in the main body of the thesis. The studies included span a range of model architectures, quantization schemes, and deployment targets.

B.1.1 Post-Training Quantization Studies

Study	Model Class	Bit Width	Calibration Method	Target Hardware
[3]	LLM (transformer)	INT8 weights, INT8 activations	Percentile, 512 samples	Edge SoC, FPGA
[15]	Vision transformer	INT8 weights, INT4 activations	Entropy-based	Edge TPU, mobile
[17]	LLM (transformer)	INT4 weights, INT8 activations	Activation-aware	Mobile CPU, GPU
[27]	Small LM (1B-1.5B)	INT8, INT4, mixed	Min-max, percentile	CPU-only cloud

Table B.1: Representative PTQ studies for transformer-based models.

The studies in Table B.1 illustrate the diversity of calibration strategies applied to post-training quantization. Yuan [3] adopt percentile-based calibration with 512 text samples, reporting minimal accuracy degradation for the quantized LLM relative to the FP16 baseline. Kim [15] use an entropy-based calibration method that selects quantization ranges by minimising the Kullback-Leibler divergence between the original and quantized activation distributions. Lin [17] introduce activation-aware scaling, which assigns higher precision to weights corresponding to activations with larger magnitudes. Sciammarelli [27] compare min-max and percentile calibration across SLMs, finding that percentile clipping consistently yields better perplexity scores on the target corpus.

B.1.2 Quantization-Aware Training Studies

Study	Model Class	Training Strategy	Fine-Tuning Data	Reported Metric
[5]	Transformer (general)	QAT with straight-through estimator	Task-specific (100k-500k tokens)	Perplexity within 0.5 of FP16
[12]	LLM (transformer)	QAT with knowledge distillation	1B tokens (subset of training data)	Perplexity within 0.3 of FP16
[1]	Vision transformer	QAT with percentile clipping	ImageNet subset (1.2M images)	Top-1 accuracy within 0.8%

Table B.2: Representative QAT studies for transformer-based models.

Sayyaparaju [5] apply quantization-aware fine-tuning to transformer models using the straight-through estimator for gradient approximation during backpropagation. The fine-tuning process uses between 100,000 and 500,000 tokens of task-specific data, achieving perplexity scores within 0.5 of the full-precision baseline. Chae and Kang [12] combine QAT with knowledge distillation, where the quantized student model is trained to match the output distribution of the full-precision teacher model. Their approach yields perplexity within 0.3 of the baseline while enabling deployment on FPGA hardware. Yamaguchi [1] illustrates a similar technique for vision transformers, combining QAT with percentile clipping to achieve an 86% reduction in non-volatile memory usage.

B.1.3 Mixed-Precision Quantization Studies

Study	Precision Scheme	Allocation Criterion	Hardware Target	Reported Efficiency Gain
[7]	INT8 + INT4 layers	Sensitivity analysis	Edge GPU, NPU	2.3× speedup over INT8
[3]	INT8 + INT4 per-layer	Outlier ratio	Edge SoC	1.8× memory reduction

Study	Precision Scheme	Allocation Criterion	Hardware Target	Reported Efficiency Gain
[14]	NF4 + INT8 mixed	Weight importance	NVIDIA GPU	Energy savings only with runtime support

Table B.3: Representative mixed-precision quantization studies.

Liu [7] apply mixed-precision quantization to YOLO-based object detectors, allocating INT8 precision to sensitive layers and INT4 precision to reliable layers based on a per-layer sensitivity analysis. Yuan [3] adopt a similar allocation strategy for LLMs, using the outlier ratio -- the proportion of values exceeding a threshold -- as the criterion for assigning higher precision. Zhang [14] focus on the energy implications of mixed-precision schemes, indicating that weight-only quantization does not always save energy on NVIDIA GPUs because the runtime dequantization and re-quantization steps can offset the memory bandwidth savings.

B.2 Integer-Only Deployment Constraints

The following table summarises the hardware constraints reported across the studies that target integer-only deployment.

Constraint	Typical Range	Impact on Quantization Design
Available on-chip SRAM	256 KB - 8 MB	Limits model capacity; requires aggressive quantization
Peak compute throughput	1 - 256 GOPS	Determines achievable token generation rate
Supported bit widths	INT8, INT4, INT2	Restricts precision choices
Presence of SIMD units	ARM NEON, RISC-V P	Enables efficient vectorised integer operations
Power budget	100 mW - 5 W	Caps both compute and memory access frequency

Table B.4: Typical hardware constraints reported for integer-only edge devices.

These constraints directly shape the feasible quantization strategy. Limited on-chip SRAM, for instance, forces a choice between storing the full model in off-chip memory -- incurring high latency and energy costs -- or applying aggressive quantization to fit within the available on-chip buffers. The absence of SIMD units for certain bit widths can negate the theoretical throughput benefits of lower-precision arithmetic.

B.3 Reported Accuracy-Efficiency Trade-Offs

Study	Bit Width	Memory Reduction vs FP16	Perplexity Increase (vs FP16)	Tokens per Second
[3]	INT8	50%	+0.2	45.3 (edge SoC)
[3]	INT4	75%	+1.1	82.1 (edge SoC)
[27]	INT8	50%	+0.3	38.7 (CPU only)
[27]	INT4	75%	+1.8	71.2 (CPU only)
[17]	INT4	75%	+0.6	Not reported

Table B.5: Accuracy-efficiency trade-offs reported across studies.

The data in Table B.5 reveal a consistent pattern: moving from INT8 to INT4 roughly halves the memory footprint while doubling the throughput, but comes at the cost of a measurable increase in perplexity. The magnitude of the perplexity increase varies across studies, likely due to differences in model architecture, calibration method, and evaluation dataset.

Appendix C: Glossary of Terms

The following glossary defines key technical terms used throughout this thesis. Each definition is grounded in the context of neural network quantization and integer-only deployment.

Term	Definition
Activation	Output of a neural network layer; intermediate tensor passed between layers during inference
Asymmetric Quantization	Quantization where the zero-point is non-zero, allowing the mapping of asymmetric distributions
Bit Width	Number of bits used to represent each quantized value (e.g., INT8, INT4)
Calibration	Process of determining quantization parameters (scale factor, zero-point) using representative data
Dequantization	The inverse of quantization; converting quantized values back to a floating-point approximation
Edge Device	Hardware with limited compute, memory, and power resources, typically deployed at the network periphery
Floating-Point Unit	Hardware component dedicated to performing arithmetic on floating-point numbers
Inference	The forward pass of a neural network to produce a prediction from an input
Integer-Only Arithmetic	Computation performed entirely using integer operations, without any floating-point fallback
Knowledge Distillation	Training a smaller student model to mimic the output distribution of a larger teacher model
Layer Normalization	Normalisation technique applied across the feature dimension of transformer layers
Lookup Table	Precomputed mapping from input values to output values, used to approximate non-linear functions
Mixed-Precision Quantization	Quantization scheme where different layers or operations use different bit widths

Term	Definition
Outlier	A value in a weight or activation tensor that lies significantly outside the typical range
Per-Channel Quantization	Quantization where each output channel has its own scale factor and zero-point
Per-Tensor Quantization	Quantization where a single scale factor and zero-point apply to the entire tensor
Percentile Clipping	Calibration method that discards a specified percentage of extreme values when setting quantization ranges
Post-Training Quantization	Quantization applied to a pre-trained model without additional training
Quantization	The process of mapping a continuous or high-precision value to a discrete, lower-precision representation
Quantization-Aware Training	Training or fine-tuning that simulates quantization effects, allowing the model to adapt
Quantization Scale Factor	A scalar parameter that maps the range of real values to the range of integer values
Quantization Zero-Point	An integer offset that shifts the quantized range, used in asymmetric quantization
Scale Factor	See Quantization Scale Factor
Straight-Through Estimator	Gradient approximation technique that passes gradients through non-differentiable quantization operations
Symmetric Quantization	Quantization where the zero-point is zero, assuming a symmetric distribution around zero
Transformer	Neural network architecture based on self-attention, forming the foundation of modern LLMs
Zero-Point	See Quantization Zero-Point

Table C.1: Glossary of key terms.

The terms listed above are used throughout the main body of the thesis. Readers unfamiliar with the quantization literature should refer to this glossary for definitions that are specifically scoped to the context of LLM deployment on integer-only hardware.

Appendix D: Selected Frameworks and Toolchains

D.1 Overview of Available Quantization Toolchains

The following appendix provides a descriptive overview of software frameworks and toolchains that support quantization for edge deployment. This information is intended to guide readers toward practical implementation resources.

Framework	Supported Bit Widths	Quantization Type	Target Hardware
TensorFlow Lite	INT8, INT16, FP16	PTQ, QAT	ARM, x86, mobile GPUs
PyTorch (FX quantization)	INT8, INT4	PTQ, QAT	CPU, GPU, mobile
Apache TVM	INT8, INT4, mixed	PTQ (auto-tuning)	CPU, GPU, FPGA, edge
ONNX Runtime	INT8, INT4, FP16	PTQ, QAT	CPU, GPU, NPU
Qualcomm AI Engine	INT8, INT16	PTQ, QAT	Qualcomm DSP, Adreno GPU

Table D.1: Selected quantization frameworks and their supported configurations.

TensorFlow Lite offers mature support for INT8 quantization on ARM-based edge devices, with both post-training and quantization-aware training pipelines available. The framework handles the conversion of non-linear operations -- such as softmax and tanh -- to integer approximations internally, enabling full integer-only execution on devices that lack floating-point units.

PyTorch's FX quantization framework provides a more flexible interface for defining custom quantization schemes, including support for mixed-precision configurations. The framework

is particularly well-suited for research prototyping, though its deployment target support is broader for CPU than for specialized edge accelerators.

Apache TVM distinguishes itself through its automated tuning capability. The framework can explore different quantization configurations and operator implementations, selecting the combination that maximises throughput on a given hardware target. This automation is valuable for integer-only FPGA deployments, where the design space is large and manual exploration is impractical.

D.2 Documented Deployment Examples from the Literature

Study	Framework Used	Model Deployed	Deployed Hardware	Reported Throughput
[3]	Custom (PyTorch-based)	LLM (transformer)	Edge SoC (INT8)	45.3 tokens/sec
[12]	Custom (Vivado HLS)	LLM (transformer)	FPGA (INT8)	32.7 tokens/sec
[27]	TensorFlow Lite	SLM (1B-1.5B)	CPU only (INT8)	38.7 tokens/sec
[50]	TensorFlow Lite	CNN	Raspberry Pi (INT8)	Not reported

Table D.2: Documented deployment examples with reported throughput.

Yuan [3] deploy a quantized LLM on an edge system-on-chip using a custom quantization pipeline built on PyTorch. The deployment achieves 45.3 tokens per second at INT8 precision, showing that integer-only inference is feasible for interactive applications. Chae and Kang [12] adopt a different path, implementing a custom hardware design on FPGA using Vivado HLS. Their design achieves 32.7 tokens per second for a quantized transformer model, illustrating the additional overhead introduced by hardware-level implementation constraints.

Sciammarelli [27] and Ameen [50] both use TensorFlow Lite for deployment on CPU-only hardware. The SLM deployment reported by Sciammarelli achieves 38.7 tokens per second on a

server-class CPU, suggesting that quantization can enable acceptable performance even without dedicated accelerator hardware.

D.3 Open Research Tools and Repositories

Several open-source tools and repositories have been instrumental in advancing quantization research for LLMs. While a wide-ranging listing is beyond the scope of this appendix, three notable examples are highlighted.

Activation-aware weight quantization (AWQ) [17] provides an open-source implementation of the proposed algorithm alongside pre-computed scaling factors for several popular LLM architectures. The repository includes scripts for applying AWQ to models from the LLaMA and Mistral families, making it a practical starting point for researchers seeking to reproduce the reported results.

The integer-only quantization framework proposed by Yuan [3] is accompanied by source code that implements the full pipeline -- from calibration and quantization to deployment -- on representative edge SoCs. The repository includes example configurations for INT8 and mixed INT8-INT4 deployments.

BitMamba-2 [42], while focused on state space models rather than transformers, provides a reference implementation of 1.58-bit quantization using ARM NEON intrinsics. The codebase establishes how sub-2-bit quantization can be implemented efficiently on CPU architectures, offering lessons that may transfer to transformer-based LLMs.

D.4 Future Directions for Toolchain Development

The current environment of quantization toolchains, while advancing rapidly, still presents gaps for integer-only LLM deployment. First, few frameworks provide native support for the full integer-only constraint, where all non-linearities must be approximated without floating-point fallback. Most frameworks assume the presence of at least a scalar floating-point unit for non-critical operations. Second, the calibration workflows across frameworks are not standardised, making it

difficult to reproduce accuracy results across different toolchains. Third, support for sub-INT4 bit widths -- INT2 and binary -- remains experimental in most frameworks, despite growing research interest in extreme quantization.

Addressing these gaps will require closer collaboration between the hardware design and software framework communities, with standardised benchmarks and reproducible evaluation protocols providing a shared foundation for progress.

References

[001] Yamaguchi. Et al., ”Compact Edge Vision Transformer with 86% Non-volatile Memory Bit Reduction by Percentile Clipping, Per Layer Quantization, and Quantization Aware Training,” 2023.

[003] Yuan. Et al., ”An Integer-only Quantization Framework for Edge Deployment of Large Language Models,” 2025.

[004] Zhong., Chaojie., ”A Hybrid Asymmetric Integer-Only Quantization Method of Neural Networks for Efficient Inference,” 2022.

[005] Sayyaparaju., ”Q-Trans: Quantization-Aware Fine-Tuning of Transformer Models for Energy-Efficient IoT and Edge Deployment,” 2025.

[007] Liu. Et al., ”MPQ-YOLO: Ultra Low Mixed-Precision Quantization of YOLO for Edge Devices Deployment,” 2023.

[011] Novkin., Amrouch., Klemme., ”Approximation-Aware and Quantization-Aware Training for Graph Neural Networks,” 2023.

[012] Chae., Kang., ”LLM on FPGA: Squeezing Language Models by Quantization and Multi-Query Attention and its Efficient Hardware Architecture,” 2025.

[013] Nisiotis., Markov., ”Quantization at the Edge: Evaluating Inference Performance and Quality for SLM Driven Conversational Agents in Virtual Worlds,” 2026.

[014] Zhang., ”Weight-Only Quantization Does Not Always Save Energy: An Empirical Study of LLM Inference Across NVIDIA GPU Platforms,” 2026.

[015] Kim., Lee., Kim., ”IPTQ-ViT: Post-Training Quantization of Non-linear Functions for Integer-only Vision Transformers,” 2026.

[016] Sari., Courville., Partovi Nia., ”iRNN: Integer-only Recurrent Neural Network,” 2022.

[017] Lin. Et al., ”AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration,” *GetMobile: Mobile Computing and Communications*, vol. 28, pp. 12-17, 2025.

[023] Solomon., ”A Systematic Literature Review of LLM Evaluation Methods,” 2026.

[024] Alfaro Pizana., Noguer I Alonso., ”Modern LLM Evaluation Techniques: A Mathematical Framework From Classical Metrics to LLM-as-Judge and Psychometric Foundations,” 2026.

[027] Sciammarelli., ”Empirical Analysis of Small Language Model Quantization for CPU-Only Cloud Infrastructure,” 2026.

[032] Lin. Et al., ”Integer-arithmetic-only Certified Robustness for Quantized Neural Networks,” 2021.

[042] Rasatavohary., ”State Space Models as CPU-Native Neural Network Architectures,” 2026.

[050] Ameen., Theodoridis., Siriwardana., ”Efficient Convolutional Neural Networks on Raspberry Pi: Enhancing Performance with Pruning and Quantization,” 2024.