Quantization-Aware Training: A Comprehensive Survey

Anonymous ACL submission

Abstract

With the unprecedentedly rapid development of LLMs, the prohibitive memory footprints and intensive computational demands of models have emerged as critical bottlenecks. Quantization-Aware Training (QAT) techniques have emerged as a primary solution to address these challenges by explicitly simulating quantization effects within the training loop, enabling low-bit models to achieve accuracy comparable to their full-precision counterparts. In this work, we strive to provide a comprehensive survey on QAT, serving as a valuable resource for researchers aiming to understand the theory of QAT and its evolving implementation landscape. To the best of our knowledge, this is the first systematic survey dedicated to reviewing the recent developments of QAT. In this paper, we systematically review existing QAT methodologies based on a taxonomy organized by quantization targets. We provide an in-depth analysis of the technical connections and distinctions among these methods and summarize their evaluation paradigms. Furthermore, we discuss persistent challenges and outline potential directions for future research.

1 Introduction

Recent years have witnessed rapid progress in large-scale foundation models across language, vision, and generative modeling, including large language models (LLMs), vision transformers (ViTs), and diffusion-based generative models. However, as the scale of these models grows exponentially, their demands on memory and computational resources far exceed the capacity of most edge devices and even consumer-grade GPUs (Nagel et al., 2021; Wei et al., 2024; Yang et al., 2025; Liu et al., 2025a). To mitigate these resource bottlenecks, model quantization serves as a fundamental technique, compressing the numerical representation of these massive architectures by lowering the bit-width of their parameters (Jacob et al., 2018; Nagel et al., 2021; Wei et al., 2024).

Model quantization methods can be broadly divided into Post-training quantization (PTQ) and Quantization-aware training (QAT) (Nagel et al., 2021; Wei et al., 2024; Yang et al., 2025). PTQ applies low-precision quantization to a pre-trained language model without requiring further training, thereby reducing model size while improving inference efficiency (Xiao et al., 2023; Liu et al., 2024b; Zhang and Shrivastava, 2024). However, directly quantizing the model can result in a significant performance degradation due to numerical overflow and limited representational range, which has been repeatedly observed across numerous quantization methods, making researchers cautious about directly applying PTQ. For instance, Bondarenko et al. (2021) identified that activations in residual connections contain structured outliers with high dynamic ranges, which are difficult to represent in low-bit formats, thereby causing significant quantization error.

In contrast to PTQ, QAT simulates quantization during training using fake quantization modules, optimizing the model directly under quantized inference constraints and often recovering the accuracy loss of PTQ, especially at ultra-low bit-widths such as INT4 (Jacob et al., 2018; Bengio et al., 2013). When combined with other parameter-efficient adaptation techniques such as low-rank adaptation (Ke et al., 2024; Xu et al., 2024) and knowledge distillation (Hinton et al., 2015; Kim et al., 2022), QAT can further improve ultra-low precision training and quantized performance. Specifically, Xu et al. (2024) proposed QA-LoRA to integrate group-wise quantization into low-rank adapters, effectively closing the accuracy gap between PTQ and QAT with minimal overhead. Furthermore, Chen et al. (2025a) demonstrated that such efficient QAT methods can successfully retain the performance of LLMs even in ultra-low

precision (e.g., 2-bit) settings.

Despite the growing interest and advancements, there is a lack of comprehensive surveys that thoroughly examine the various methods, challenges, and applications specific to QAT. Most existing surveys on neural network quantization adopt a broad perspective, and consequently provide only a coarse-grained analysis of QAT. For instance, Wei et al. (2024) present a broad taxonomy of neural network quantization, but their discussion of QAT remains largely introductory and does not offer a detailed analysis of key challenges. Likewise, Nagel et al. (2021) focus on deployment-oriented industrial pipelines, with limited theoretical or analytical depth of QAT. More recently, Yang et al. (2025) organize quantization methods primarily by target architectures, but this coarse-grained categorization offers limited insight into module-specific QAT strategies. In contrast, this survey focuses primarily on QAT and organizes the literature accordingly. This perspective highlights how the numerical characteristics and access patterns of different targets shape the design of QAT schemes, and why the benefits and costs of QAT differ across targets.

We organize recent progress in QAT from four aspects: theoretical foundations, quantization targets, evaluation paradigms, and future horizons. Firstly, we establish the theoretical foundations, focusing on the mathematical formulation of uniform quantization, the fake quantization mechanism, and gradient approximation techniques such as the Straight-Through Estimator (STE). As QAT adapts to diverse model architectures, we explore the quantization targets involved in compressing them, emphasizing not only standard weights and activations but also emerging bottlenecks like the KV cache and gradients. We provide a comprehensive overview of evaluation methods, distinguishing between structural metrics and functional metrics. Lastly, we discuss persistent limitations such as optimization instability and explore potential directions for future research.

The remainder of this survey is organized as follows: Section 2 establishes the preliminary knowledge and theoretical framework. Section 3 categorizes existing research based on quantization targets. Finally, Section 4 identifies key challenges and outlines future opportunities, while evaluation paradigms and metrics are detailed in Appendix A. Figure 1 provides a chronological overview of representative QAT methods (2020–2025), complementing our target-based taxonomy in Section 3.

2 Preliminary Knowledge

This section introduces the notation and mathematical formulation of uniform affine quantization, and summarizes the core mechanism of QAT (Jacob et al., 2018; Nagel et al., 2021). We focus on the quantize–dequantize (fake quantization) operator used during training and the surrogate gradients (e.g., STE) needed to backpropagate through rounding and clipping.

2.1 Uniform Quantization

Quantization maps high-precision real-valued tensors (typically FP32) to a low-precision discrete set amenable to efficient integer arithmetic and reduced memory (Jacob et al., 2018; Nagel et al., 2021). Different precisions (bit-widths b) offer a trade-off between model efficiency and representational capacity: while FP32 provides high dynamic range and precision, lower bit-widths (e.g., INT8, INT4) significantly reduce memory footprint and computational cost, yet at the risk of information loss. Uniform quantization is a de facto standard due to its simplicity and compatibility with integer arithmetic (Jacob et al., 2018; Nagel et al., 2021).

Let $x \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ denote a real-valued tensor (e.g., weights or activations). Uniform quantization is parameterized by a positive step size $s > 0$ and an integer zero-point $z \in \mathbb{Z}$. The step size sets the grid resolution, and the zero-point makes real zero exactly representable in the integer domain.

Let b be the bit-width and let $[q_{\min}, q_{\max}]$ be the integer representable range (e.g., $q_{\min} = -2^{b-1}$ and $q_{\max} = 2^{b-1} - 1$ for signed integers). The quantization operator first maps x to an integer tensor $x_{\text{int}} \in \mathbb{Z}^{d_1 \times \cdots \times d_k}$:

$$x_{\text{int}} = \text{clamp}\Big(\Big\lfloor \frac{x}{s} \Big\rceil + z,\ q_{\min},\ q_{\max}\Big), \qquad (1)$$

where $\lfloor \cdot \rceil$ denotes deterministic rounding-to-nearest, and $\text{clamp}(\cdot, q_{\min}, q_{\max})$ clips values to the integer range (Jacob et al., 2018).

During training, QAT typically performs computations in floating point while emulating quantization effects via a quantize–dequantize (fake quantization) step (Jacob et al., 2018):

$$\hat{x} = s \cdot (x_{\text{int}} - z), \qquad (2)$$

where $\hat{x}$ is the fake-quantized proxy of x used in the forward pass. The quantization error (or injected noise) is $\epsilon = x - \hat{x}$.

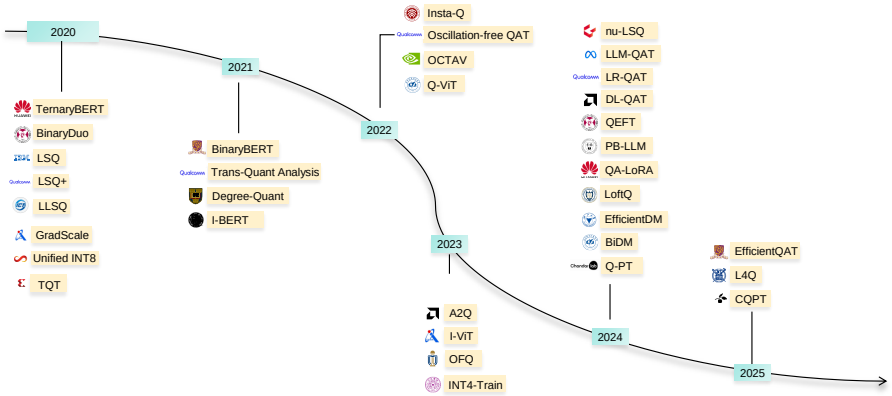

Figure 1: Timeline of representative QAT methods from 2020 to 2025. Works are grouped by publication year along the timeline to provide a high-level view of the field's evolution (non-exhaustive).

A common choice is to derive (s, z) from a chosen clipping range $[x_{\min}, x_{\max}]$:

$$s = \frac{x_{\max} - x_{\min}}{q_{\max} - q_{\min}}, \qquad z = \left\lfloor q_{\min} - \frac{x_{\min}}{s} \right\rceil. \tag{3}$$

In practice, $[x_{\min}, x_{\max}]$ can be estimated from calibration statistics (e.g., min–max) (Nagel et al., 2021), optimized by minimizing local reconstruction error (Sakr et al., 2022), or learned jointly with model parameters (Esser et al., 2020).

Symmetric vs. Asymmetric Quantization. Uniform affine quantization is commonly instantiated in two forms: *Asymmetric quantization* uses a nonzero z (Eq. 3), which is beneficial for non-zero-centered or non-negative activations (e.g., ReLU). It reduces wasted quantization levels on skewed ranges, at the cost of extra zero-point terms in integer kernels (Jacob et al., 2018; Nagel et al., 2021). *Symmetric quantization* typically enforces $z = 0$ and uses a symmetric clipping range (e.g., $x_{\min} = -x_{\max}$):

$$\hat{x} = s \cdot x_{\text{int}}. \tag{4}$$

Consequently, many deployment pipelines apply symmetric quantization to network weights to simplify integer kernels, while network activations are often quantized asymmetrically due to their skewed ranges.

2.2 QAT Mechanism

Unlike PTQ, QAT incorporates quantization effects during training by inserting quantization operators into the training graph (Jacob et al., 2018; Nagel et al., 2021). Specifically, QAT inserts fake quantization operators into the forward pass so that the training process can adapt the network parameters to minimize the task loss under discretization. This narrows down the training-inference discrepancy, resulting in models whose predictions are robust to the low-precision scenarios during inference.

Fake Quantization. In QAT, quantization is simulated by replacing some tensors with their fake-quantized counterparts. For a layer that computes $y = f(Wx + b)$, QAT computes

$$\begin{aligned} \hat{W} = \mathcal{Q}(W), \quad \hat{x} = \mathcal{Q}(x), \\ y = f(\hat{W}\,\hat{x} + b), \end{aligned} \tag{5}$$

where $\mathcal{Q}(\cdot)$ denotes the quantize–dequantize operator (Eqs. 1–2). Model parameters are often not stored as integers during training; instead, QAT maintains full-precision master weights W and uses $\hat{W}$ only for forward computation. This preserves small gradient updates that would otherwise vanish after discretization and stabilizes training.

Gradient Mismatch. A critical challenge is that Eq. 1 involves a non-differentiable rounding operator. Formally, the derivative of rounding is zero

almost everywhere and undefined at integers:

$$\frac{\partial \lfloor u \rceil}{\partial u} = 0 \quad \text{a.e.} \tag{6}$$

Therefore, naive backpropagation through the quantization path yields vanishing or ill-defined gradients, preventing effective optimization of W (Nagel et al., 2022).

Gradient Approximation. To address the gradient mismatch issue, QAT relies on surrogate gradients that provide a usable gradient through rounding and clipping. Straight-Through Estimator (STE) Bengio et al. (2013) is commonly used, treating rounding as an identity operation in the backward pass (optionally combined with clipping). A common formulation is

$$\frac{\partial \mathcal{L}}{\partial x} \approx \frac{\partial \mathcal{L}}{\partial \hat{x}} \cdot \mathbb{I}(x \in [x_{\min}, x_{\max}]), \tag{7}$$

where $[x_{\min}, x_{\max}]$ is the chosen clipping range and $\mathbb{I}(\cdot)$ is the indicator function. Intuitively, STE allows gradients to pass through quantization as if $\hat{x} \approx x$ within the valid range, while saturating gradients when inputs are clipped.

Optimizing Quantization Parameters. Beyond model weights, some QAT methods also learn quantization parameters such as step sizes, offsets (zero-points, Bhalgat et al. (2020)), and clipping thresholds (Jain et al., 2020). LSQ (Esser et al., 2020) treats s c.f. Eqs. 1 as a learnable variable and updates it via gradient descent using an STE-style surrogate. For example, consider the symmetric quantization case ($z = 0$, c.f. Eq. 4) with

$$\hat{x} = s \cdot \mathrm{clamp}\Big(\Big\lfloor \frac{x}{s} \Big\rceil,\ q_{\min},\ q_{\max}\Big). \tag{8}$$

According to LSQ, the gradient w.r.t. s can be written piecewise as

$$\frac{\partial \hat{x}}{\partial s} = \begin{cases} \mathrm{clamp}\Big(\Big\lfloor \frac{x}{s} \Big\rceil,\ q_{\min},\ q_{\max}\Big) - \frac{x}{s}, \\ \qquad q_{\min} < \frac{x}{s} < q_{\max} \\ q_{\min}, \quad \frac{x}{s} \leqslant q_{\min}, \\ q_{\max}, \quad \frac{x}{s} \geqslant q_{\max}. \end{cases} \tag{9}$$

Learning s enables the model to balance clipping-induced distortion and rounding-induced error in a task-driven manner. In practice, QAT could further incorporate distillation or regularization terms to improve stability under aggressive low-bit settings (Hinton et al., 2015; Kim et al., 2022).

3 Taxonomy

In this section, we firstly structure the QAT literature by the *quantization target*, e.g., weights, activations, KV cache, and gradients. While many methods quantize multiple targets simultaneously, we group each work under the targets where its key ideas, algorithmic designs, and ablations are centered. A taxonomy of representative QAT frameworks under this view is summarized in Table 1.

3.1 Quantization Target: Model Weights

This section surveys works in which the QAT method primarily quantizes the model weights. However, it should be noted that QAT methods often simultaneously quantize the activations — often at a fixed precision such as 8-bit — as an auxiliary design choice. The goal of weight-centric QAT is to compress model parameters while keeping activations in high precision. The key obstacles include unstable optimization under STE, limited expressivity of low-bit grids, and deployment constraints such as accumulator overflow and supported granularities. We summarize recent progress along two themes covering PEFT integration and ultra-low bit.

3.1.1 PEFT Integration

Conventional end-to-end QAT quantizes all parameters, which is infeasible for billion-scale models. Recent work therefore focuses on parameter-efficient weight-QAT, where quantization is restrictively applied to structured or low-rank subspaces. One line of research reduces training memory by updating only quantization-related or auxiliary parameters. EfficientQAT (Chen et al., 2025a) progressively learns only quantization parameters, while DL-QAT (Ke et al., 2024) formulates QAT as low-rank adaptation in a decomposed quantization space, enabling effective ultra-low-bit adaptation. Another line of work integrates QAT with low-rank adaptation, inspired by LoRA (Hu et al., 2022). L4Q (Jeon et al., 2025) achieves LoRA-like training cost with a fully quantized final model, whereas QA-LoRA (Xu et al., 2024) increases quantization flexibility via group-wise operators to enable lossless adapter merging.

Initialization and precision allocation are also crucial. LoftQ (Li et al., 2024b) reduces quantization error through joint backbone quantization and low-rank initialization, while QEFT (Lee et al., 2024) introduces sensitivity-aware mixed precision with hardware-friendly reordering.

3.1.2 Ultra-Low Bit

Pushing the precision of model weights below 3 bits often leads to severe optimization instability and capacity degradation (Bai et al., 2021; Zhang et al., 2020). Prior work mitigates these issues through sensitivity-aware precision allocation (Shen et al., 2019) and strong supervision, e.g., knowledge distillation and progressive training (Bai et al., 2021). For decoder-only LLMs, preserving a small subset of salient weights becomes particularly important in ultra-low-bit regimes. PB-LLM (Yuan et al., 2024) shows that full binarization harms reasoning ability and instead advocates partial binarization, where salient weights remain in higher precision. Similarly, Nielsen et al. (2025) demonstrates that gradually transitioning from high to ultra-low precision, together with optimizer-state preservation, can alleviate loss spikes under extreme quantization.

Overall, ultra-low-bit weight-centric QAT can reduce model memory footprint by an order of magnitude, enabling integer-only inference and on-device deployment. A key takeaway from these works is that success in ultra-low-bit regimes depends less on increasingly sophisticated low-bit optimizers (Xu et al., 2025) and more on selectively preserving critical weights and stabilizing training with strong supervision.

3.2 Quantization Target: Activations

Activation-centric QAT maps intermediate features (e.g., token embeddings and residual streams) from high-precision floating-point formats to low-bit representations during the forward pass. Activation-centric quantization enables integer-only inference, reduces memory overhead in long-sequence processing, and improves energy efficiency on resource-constrained hardware. However, activation QAT faces several challenges: (i) structural outliers that dominate dynamic ranges, (ii) gradient instability caused by discrete mappings, and (iii) non-stationary activation statistics, particularly in diffusion models.

To mitigate the performance degradation caused by discretizing activations, recent work in activation QAT focuses on introducing learnable quantization parameters and enhanced gradient estimation tailored for activation distributions. LSQ+ (Bhalgat et al., 2020) introduces asymmetric activation quantization with learnable offsets to better fit asymmetric activation distributions. Under extreme compression such as binarization, gradient mismatch becomes the primary bottleneck. BinaryDuo (Kim et al., 2020) formalizes this issue via Coordinate Discrete Gradient (CDG) and mitigates it through training–inference decoupling: ternary activations are used during training to stabilize gradients, while binary activations are adopted at inference to maintain deployment efficiency.

3.3 Quantization Target: Weight + Activation

Joint weight and activation quantization aligns training with the coupled distortion of the genuine inference graph used in practical deployment, enabling weights, activations, and operators to co-adapt to the quantization noise. However, it introduces coupled challenges including non-stationary activation distributions, global range dependencies, and severe gradient mismatch at ultra-low bit widths, alongside the need for integer-friendly operator redesigns. We organized existing work into four directions: representationalinteger Transformers, dynamic adaptation, quantized generators.

3.3.1 Quantization Range

A key challenge in QAT is accurately modeling the quantization range of activations. Under joint quantization, upstream quantization continuously alters activation distributions, causing the effective dynamic range to drift during training. Consequently, static, layer-wise range calibration is often inadequate. To address this issue, many methods treat the quantization range as a learnable parameter and optimize it jointly with model weights. The range can be adapted in different forms. Some approaches adjust it implicitly by learning non-uniform step sizes to better accommodate heavy-tailed or multi-modal activations (Gongyo et al., 2024). Others explicitly constrain the range through symmetric linear quantization, as in LLSQ (Zhao et al., 2020), to maintain compatibility with integer-only hardware. The quantization range can also be controlled via learnable clipping thresholds. TQT (Jain et al., 2020) directly optimizes clipping boundaries, while OCTAV (Sakr et al., 2022) further incorporates gradient-aware differentiation to better handle forward–backward interactions in low-precision training.

3.3.2 Integer Transformers

Research on Integer Transformers (Kim et al., 2021) views Transformer quantization as a system-level problem, where numerical fragility arises

not only from linear projections but also from core operators such as softmax, normalization, and smooth nonlinearities. Under strict integer arithmetic constraints, joint quantization therefore requires operator-aware approximations and careful scale management.

Representative Integer Transformer approaches, including IBERT (Kim et al., 2021) and IViT (Li and Gu, 2023), pursue integer-only inference pipelines by reformulating or approximating non-integer-friendly operations and explicitly controlling intermediate activation scales to maintain stability across Transformer blocks. Subsequent studies further reveal that, within this integer computation regime, weight-only quantization is insufficient: the dominant quantization errors and memory costs often originate from the activations inside attention and feed-forward sublayers. Accordingly, methods such as QViT (Li et al., 2022) emphasize joint quantization of weights and activations as a prerequisite for preserving accuracy in Integer Transformers.

3.3.3 Dynamic Adaptation

Because activation distributions vary with input, topology, or context, joint quantization can fail when treated as a globally fixed configuration. This motivates heterogeneity-aware strategies that adapt quantization behavior to structural or instance-specific variation. DegreeQuant (Tailor et al., 2021) targets graph neural networks, where degree-dependent aggregation induces systematic differences in activation magnitude and variance across nodes; it thus incorporates degree awareness into QAT to avoid overly conservative global ranges. Similarly, to address instance-specific variation in vision tasks, Liu et al. (2022) propose a dynamic framework adjusting quantization precision conditioned on individual input samples. Their method employs a lightweight bit-controller to predict optimal layer-wise bit-widths for both weights and activations, thereby reducing computational redundancy for simple inputs while preserving high precision for complex ones.

3.3.4 Quantized Generators

For generative models such as diffusion models, quantization introduces small errors at each step, which can accumulate along the diffusion time steps. For example, BiDM (Zheng et al., 2024) shows that degradation under joint quantization is not only per-layer but also per-step, because early-step errors can change the denoising trajectory and cannot be fully corrected later. EfficientDM (He et al., 2024) focuses on efficient quantization-aware fine-tuning to adapt pretrained diffusion models to low precision under constrained training budgets. Together, these works suggest that joint QAT for generative models must account for timestep distribution shifts and compounding errors, making robustness and training efficiency jointly critical.

3.4 Quantization Target: KV Cache

As the Transformer context window expands, the KV cache becomes a primary bottleneck for memory footprint and bandwidth. Quantization of the KV cache has emerged as a critical compression target, distinct from but complementary to weight or/and activation quantization. Unlike static storage compression, KV cache quantization introduces errors that recursively participate in attention, leading to accumulated perturbations along decoding steps. and training cost. To address temporal variability, outlier amplification, and training cost, we group them into two types: distillation, and efficient end-to-end QAT.

3.4.1 Distillation

This category addresses the challenge of data availability, particularly when the pretraining data are inaccessible. Data-Free Distillation (Liu et al., 2024a) replaces missing pretraining data with synthetic sequences sampled from a full-precision teacher model, and uses distillation objectives to guide a quantized student model. A key principle is to treat the KV cache as part of the quantized computational graph rather than a post-hoc target: per-token quantization for cached keys and values is enabled during the forward pass so gradients can propagate through the quantized KV path. This alignment between training and inference-time behavior facilitates simultaneous low-bit quantization of weights, activations, and the KV cache, though it does not inherently eliminate the training overhead of QAT.

3.4.2 Efficient End-to-End QAT

The second framework prioritizes training efficiency, aiming to make end-to-end QAT feasible for large-scale models without sacrificing KV cache quantization. Bondarenko et al. (2024) propose LR-QAT, which combines low-rank auxiliary weights compatible with the quantization grid, downcasting of frozen pretrained parameters, and gradi-

ent checkpointing to reduce memory overhead. Despite focusing on training feasibility, LR-QAT maintains rigorous KV-cache quantization via per-token granularity for the keys and values, showing that KV compression can be integrated into memory-constrained training pipelines without redesigning inference-time caching. This approach highlights a practical balance between memory efficiency and quantization fidelity, making it particularly suitable for large-scale transformer models deployed in resource-limited environments.

3.5 Quantization Target: Gradient

Gradient quantization discretizes tensors computed during backpropagation, including layer output gradients, activation gradients, and parameter gradients. Weight or/and activation quantization primarily affect the forward pass, while gradient quantization directly perturbs the optimization trajectory. Since gradients determine the update directions and step sizes, errors introduced here can compound through deep gradient chains. Despite these stability challenges, gradient quantization can reduce training-time memory, lower bandwidth costs, and enable integer-friendly backward propagation.

Early work on gradient QAT focuses on controlling gradient magnitudes to avoid saturation and quantization noise at very low precision. Sun et al. (2020) propose GradScale, which uses layer-wise dynamic scaling with overflow feedback to fit gradients into the FP4 range, and introduce Two-Phase Rounding (TPR) with complementary FP4 grids to reduce rounding bias and improve stability. Beyond magnitude errors, gradient direction is also critical for stable training. Zhu et al. (2020) measure directional deviation using a cosine-based metric and propose direction-sensitive gradient clipping and direction-aware learning rate scaling to mitigate large direction errors due to quantization. For Transformers, aggressive low-bit gradient quantization is more fragile than in CNNs. Xi et al. (2023) exploit structural sparsity by splitting gradients into high- and low-bit components and computing gradients only for important ones. Chitsaz et al. (2024) further show that gradients are highly noise-sensitive: 4-bit gradient quantization is often unstable, while 8-bit quantization can work when applied selectively with per-token scaling. Overall, these studies show that effective gradient quantization requires careful range control, selective application, and awareness of model structure to maintain training stability.

4 Challenges and Future Directions

We have systematically reviewed current quantization-aware training strategies, based on the quantization targets. QAT alleviates the intensive computational demands of utilizing large-scale models. It further mitigates accuracy degradation in low-precision scenarios by explicitly injecting quantization effects (e.g., clipping, rounding, and rescaling) into model training, so that model parameters and activations can easily adapt to the discrete inference arithmetic in real-world deployment (Jacob et al., 2018). QAT commonly relies on surrogate gradients, most notably the straight-through estimator (STE), to enable end-to-end optimization (Bengio et al., 2013) to escape from non-differentiable operations like rounding. Despite strong empirical success, QAT still faces the following persistent challenges.

Optimization Instability QAT is fundamentally challenged by the mismatch between the discrete, non-differentiable nature of quantization and gradient-based optimization. Since quantization functions are piecewise constant, QAT relies on surrogate gradients such as STE to enable backpropagation. However, these heuristics do not reflect the true derivatives of the quantized objective, introducing biased gradient estimations and gradient mismatch (Bengio et al., 2013; Sakr et al., 2022). This mismatch naturally degrades optimization stability and convergence. Inaccurate gradients may fail to consistently reduce the quantized loss: near the quantization thresholds, small weight updates can cause abrupt changes in quantized values, leading to weight oscillation (Nagel et al., 2022; Liu et al., 2023). These problems are exacerbated in ultra-low-bit settings, where coarse quantization produces a highly non-smooth optimization landscape. In such regimes, surrogate gradients may be too weak or misaligned to move parameters across quantization boundaries, causing frozen weights that remain trapped within a single quantization bin (Lee et al., 2021). Overall, these limitations motivate the development of optimization methods that better align backward gradients with the true discrete objective.

Outliers in Activation. While weight quantization has become relatively mature, activation quantization remains a major bottleneck, especially for Transformers and LLMs. A key challenge is the presence of strong, input-dependent outliers in acti-

vations—commonly observed in attention modules and residual streams—which significantly skew the quantization range and degrade accuracy (Bondarenko et al., 2021; Xiao et al., 2023; Zhang and Shrivastava, 2024). Moreover, under aggressive joint quantization of weights and activations (e.g., W4A4), quantization noise accumulates along the layers. This accumulated noise interacts unfavorably with LayerNorm and Softmax, further amplifying performance degradation. As a result, recovering accuracy often relies on complex algorithmic techniques, such as knowledge distillation or rotation-based transformations, which introduce additional optimization difficulty and engineering overhead (Zhang et al., 2020; Liu et al., 2024b).

The Simulation-to-Deployment Gap. A persistent practical challenge lies in the discrepancy between fake quantization—typically simulated in FP32 during training—and true integer-only inference at deployment time. QAT pipelines often overlook low-level hardware constraints, including accumulator overflow, hardware-specific rounding modes, and operator fusion behaviors (Jacob et al., 2018; Colbert et al., 2023). As a result, models converged successfully under QAT simulations may experience substantial accuracy degradation—or even fail to execute efficiently—when deployed on real-world integer-only hardware accelerators such as DSPs or NPUs (Kim et al., 2021; Li and Gu, 2023).

High Training Overhead. QAT requires a full training pipeline, including access to labeled datasets and substantial computational resources for gradient-based optimization. This overhead becomes prohibitive for billion-parameter LLMs or diffusion models (Liu et al., 2024a; Chen et al., 2025a) Moreover, QAT typically maintains full-precision master weights and optimizer states during training, which significantly increases memory consumption. As a result, there is an inherent tension trade-off between the objective of model compression and the high computational and memory costs incurred by the compression process itself (Dremov et al., 2025; Huang et al., 2024).

Homogenization. Model homogenization is an emerging challenge in model quantization. Most existing quantization methods rely on uniform quantizers, fixed bit-width allocation, and layer-agnostic design choices to simplify deployment and ensure hardware efficiency. While effective for reducing model size and inference cost, these choices can significantly constrain the expressive capacity of quantized models. As a result, models with different architectures, training objectives, or task specializations often converge to highly similar weight distributions and activation patterns after quantization, leading to diminished representational diversity (Nagel et al., 2021; Li et al., 2024a). This quantization-induced homogenization effect becomes more pronounced in large-scale models. As model size increases, quantization noise accumulates across layers, encouraging "averaged" internal representations that suppress fine-grained features and task-specific behaviors. This phenomenon can ultimately limit the performance ceilings and reduce the adaptability to downstream tasks, particularly in LLMs (Yu et al., 2024). Addressing this trade-off between compression efficiency and representational diversity remains an open research challenge in model quantization.

Evaluating quantization methods is crucial, as reductions in bit-width or model size do not necessarily translate into faster inference or consistently accurate performance. Quantization can introduce subtle changes that may not be captured by simple metrics such as compression ratio or theoretical speedup. A thorough assessment is needed to understand its real impact. In particular, evaluating across multiple dimensions helps identify trade-offs that may not be apparent when considering accuracy alone. We survey QAT evaluation methods from three key perspectives: performance, which measures task-specific accuracy; robustness, which examines the model's resilience to perturbations; and efficiency, which considers both computational cost and memory usage. For a more detailed discussion, please refer to Appendix A.

5 Conclusion

This survey reviews existing quantization-aware training (QAT) methods as an effective method of enabling efficient inference of large-scale models by simulating quantization during training. We present a unified taxonomy of the QAT methods based on various quantization targets, ranging from weight quantization to gradient quantization. We also highlight the key challenges faced by QAT, e.g., their noisy gradient estimations, high training overhead, and model homogenization. Solving the challenges is non-trivial, and we share promising directions to address them.

Limitations

This review focuses primarily on quantization-aware training, emphasizing quantization methods and the challenges they entail. It does not cover relevant datasets and benchmarks. To maintain a clear and focused discussion, we present detailed evaluations of the QAT methods in Appendix A due to the page limitations. We intend to incorporate a comprehensive discussion of relevant datasets, benchmarks, and evaluations in the final manuscript.

References

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2021. BinaryBERT: Pushing the limit of BERT quantization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4334–4348, Online. Association for Computational Linguistics.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. 2020. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 696–697.

Yelysei Bondarenko, Riccardo Del Chiaro, and Markus Nagel. 2024. Low-rank quantization-aware training for llms. *arXiv preprint arXiv:2406.06385*.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. 2025a. Efficientqat: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10081–10100.

Mengzhao Chen, Chaoyi Zhang, Jing Liu, Yutao Zeng, Zeyue Xue, Zhiheng Liu, Yunshui Li, Jin Ma, Jie Huang, Xun Zhou, and 1 others. 2025b. Scaling law for quantization-aware training. *arXiv preprint arXiv:2505.14302*.

Kamran Chitsaz, Quentin Fournier, Goncalo Mordido, and Sarath Chandar. 2024. Exploring quantization for efficient pre-training of transformer language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13473–13487, Miami, Florida, USA. Association for Computational Linguistics.

Ian Colbert, Alessandro Pappalardo, and Jakoba Petri-Koenig. 2023. A2q: Accumulator-aware quantization with guaranteed overflow avoidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16989–16998.

Ian Colbert, Alessandro Pappalardo, Jakoba Petri-Koenig, and Yaman Umuroglu. 2024. A2q+: improving accumulator-aware weight quantization. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Peiyan Dong, Lei Lu, Chao Wu, Cheng Lyu, Geng Yuan, Hao Tang, and Yanzhi Wang. 2023. Packqvit: Faster sub-8-bit vision transformers via full and packed quantization on the mobile. *Advances in Neural Information Processing Systems*, 36:9015–9028.

Aleksandr Dremov, David Grangier, Angelos Katharopoulos, and Awni Hannun. 2025. Compute-optimal quantization-aware training. *arXiv preprint arXiv:2509.22935*.

Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. 2020. Learned step size quantization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Shinya Gongyo, Jinrong Liang, Mitsuru Ambai, Rei Kawakami, and Ikuro Sato. 2024. Learning non-uniform step sizes for neural network quantization. In *Proceedings of the Asian Conference on Computer Vision*, pages 4385–4402.

Yefei He, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. 2024. Efficientdm: Efficient quantization-aware fine-tuning of low-bit diffusion models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Yuxuan Hu, Xiaodong Chen, Cuiping Li, Hong Chen, and Jing Zhang. 2025. Quad: Quantization and parameter-efficient tuning of llm with activation decomposition. *arXiv preprint arXiv:2503.19353*.

Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. 2024. Robust and efficient quantization-aware training via coreset selection. *Transactions on Machine Learning Research*.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713.

Sambhav Jain, Albert Gural, Michael Wu, and Chris Dick. 2020. Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks. *Proceedings of Machine Learning and Systems*, 2:112–128.

Hyesung Jeon, Yulhwa Kim, and Jae-Joon Kim. 2025. L4q: parameter efficient quantization-aware fine-tuning on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2002–2024.

Wenjing Ke, Zhe Li, Dong Li, Lu Tian, and Emad Barsoum. 2024. Dl-qat: Weight-decomposed low-rank quantization-aware training for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 113–119.

Hyungjun Kim, Kyungsu Kim, Jinseok Kim, and Jae-Joon Kim. 2020. Binaryduo: Reducing gradient mismatch in binary activation network by coupling binary activations. *ArXiv*, abs/2002.06517.

Minsoo Kim, Sihwa Lee, Suk-Jin Hong, Du-Seong Chang, and Jungwook Choi. 2022. Understanding and improving knowledge distillation for quantization aware training of large transformer encoders. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6713–6725, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021. I-bert: Integer-only bert quantization. In *International conference on machine learning*, pages 5506–5518. PMLR.

Changhun Lee, Jun-gyu Jin, YoungHyun Cho, and Eunhyeok Park. 2024. QEFT: Quantization for efficient fine-tuning of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13823–13837, Miami, Florida, USA. Association for Computational Linguistics.

Junghyup Lee, Dohyung Kim, and Bumsub Ham. 2021. Network quantization with element-wise gradient scaling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6448–6457.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. 2024a. Autoregressive image generation without vector quantization. *ArXiv*, abs/2406.11838.

Yanjing Li, Sheng Xu, Xianbin Cao, Xiao Sun, and Baochang Zhang. 2023. Q-dm: an efficient low-bit quantized diffusion model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. 2022. Q-vit: Accurate and fully quantized low-bit vision transformer. *Advances in neural information processing systems*, 35:34451–34463.

Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2024b. Loftq: Lora-fine-tuning-aware quantization for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yuhang Li, Xin Dong, and Wei Wang. 2020. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhikai Li and Qingyi Gu. 2023. I-vit: Integer-only quantization for efficient vision transformer inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17065–17075.

Kai Liu, Qian Zheng, Kaiwen Tao, Zhiteng Li, Haotong Qin, Wenbo Li, Yong Guo, Xianglong Liu, Linghe Kong, Guihai Chen, Yulun Zhang, and Xiaokang Yang. 2025a. Low bit model quantization for deep neural networks: A survey. *arXiv preprint arXiv:2505.05530*.

Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. 2023. Oscillation-free quantization for low-bit vision transformers. In *International conference on machine learning*, pages 21813–21824. PMLR.

Xuewen Liu, Zhikai Li, Minghao Jiang, Mengjuan Chen, Jianquan Li, and Qingyi Gu. 2025b. Dilatequant: Accurate and efficient quantization-aware training for diffusion models via weight dilation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, page 8399–8408, New York, NY, USA. Association for Computing Machinery.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2024a. Llm-qat: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 467–484.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2024b. Spinquant: Llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*.

Zhenhua Liu, Yunhe Wang, Kai Han, Siwei Ma, and Wen Gao. 2022. Instance-aware dynamic neural network quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12434–12443.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. 2021. A white paper on neural network quantization. *ArXiv*, abs/2106.08295.

Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. 2022. Overcoming oscillations in quantization-aware training. In *International Conference on Machine Learning*, pages 16318–16330. PMLR.

Jacob Nielsen, Peter Schneider-Kamp, and Lukas Galke. 2025. Continual quantization-aware pre-training: When to transition from 16-bit to 1.58-bit pre-training for BitNet language models? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13483–13493, Vienna, Austria. Association for Computational Linguistics.

Charbel Sakr, Steve Dai, Rangharajan Venkatesan, Brian Zimmer, William J. Dally, and Brucek Khailany. 2022. Optimal clipping and magnitude-aware differentiation for improved quantization-aware training. In *International Conference on Machine Learning*.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2019. Q-bert: Hessian based ultra low precision quantization of bert. In *AAAI Conference on Artificial Intelligence*.

Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. 2023. Temporal dynamic quantization for diffusion models. *Advances in neural information processing systems*, 36:48686–48698.

Xiao Sun, Naigang Wang, Chia-Yu Chen, Jiamin Ni, Ankur Agrawal, Xiaodong Cui, Swagath Venkataramani, Kaoutar El Maghraoui, Vijayalakshmi Viji Srinivasan, and Kailash Gopalakrishnan. 2020. Ultra-low precision 4-bit training of deep neural networks. *Advances in Neural Information Processing Systems*, 33:1796–1807.

Shyam A Tailor, Javier Fernandez-Marques, and Nicholas D Lane. 2021. Degree-quant: Quantization-aware training for graph neural networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Haoxuan Wang, Yuzhang Shang, Zhihang Yuan, Junyi Wu, Junchi Yan, and Yan Yan. 2025. Quest: Low-bit diffusion model quantization via efficient selective finetuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15542–15551.

Zheng Wang, Juncheng B Li, Shuhui Qu, Florian Metze, and Emma Strubell. 2022. Squat: Sharpness-and quantization-aware training for bert. *arXiv preprint arXiv:2210.07171*.

Lu Wei, Zhong Ma, Chaojie Yang, and Qin Yao. 2024. Advances in the neural network quantization: A comprehensive review. *Applied Sciences*, 14(17).

Quan Wei, Chung-Yiu Yau, Hoi To Wai, Yang Zhao, Dongyeop Kang, Youngsuk Park, and Mingyi Hong. 2025. RoSTE: An efficient quantization-aware supervised fine-tuning approach for large language models. In *Forty-second International Conference on Machine Learning*.

Haocheng Xi, ChangHao Li, Jianfei Chen, and Jun Zhu. 2023. Training transformers with 4-bit integers. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pages 38087–38099. PMLR.

Cong Xu, Wenbin Liang, Mo Yu, Anan Liu, Keqin Zhang, Lizhuang Ma, Jianyong Wang, Jun Wang, and Wei Zhang. 2025. Pushing the limits of low-bit optimizers: A focus on ema dynamics. *ArXiv*, abs/2505.00347.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. 2024. Qa-lora: Quantization-aware low-rank adaptation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jiawei Yang, Zhongbo Li, Zeqin Feng, and Yongqiang Xie. 2025. A survey on neural network quantization. In *Proceedings of the 2025 6th International Conference on Computer Information and Big Data Applications*, CIBDA '25, page 384–394, New York, NY, USA. Association for Computing Machinery.

Jeffrey Yu, Kartik Prabhu, Yonatan Urman, Robert M. Radway, Eric Han, and Priyanka Raina. 2024. 8-bit transformer inference and fine-tuning for edge accelerators. *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*.

Zhihang Yuan, Yuzhang Shang, and Zhen Dong. 2024. PB-LLM: partially binarized large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Bonan Zhang, Chia-Yu Chen, and Naveen Verma. 2024. Reshape and adapt for output quantization (raoq): quantization-aware training for in-memory computing systems. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Tianyi Zhang and Anshumali Shrivastava. 2024. Leanquant: Accurate and scalable large language model quantization with loss-error-aware grid. *arXiv preprint arXiv:2407.10032*.

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020. TernaryBERT: Distillation-aware ultra-low bit BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 509–521, Online. Association for Computational Linguistics.

Maosen Zhao, Pengtao Chen, Chong Yu, Yan Wen, Xudong Tan, and Tao Chen. 2025. Pioneering 4-bit fp quantization for diffusion models: Mixup-sign quantization and timestep-aware fine-tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18134–18143.

Xiandong Zhao, Ying Wang, Xuyi Cai, Cheng Liu, and Lei Zhang. 2020. Linear symmetric quantization of neural networks for low-precision integer hardware. In *International Conference on Learning Representations*.

Xingyu Zheng, Xianglong Liu, Yichen Bian, Xudong Ma, Yulun Zhang, Jiakai Wang, Jinyang Guo, and Haotong Qin. 2024. Bidm: Pushing the limit of quantization for diffusion models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Qihua Zhou, Song Guo, Zhihao Qu, Jingcai Guo, Zhenda Xu, Jiewei Zhang, Tao Guo, Boyuan Luo, and Jingren Zhou. 2021. Octo:{INT8} training with loss-aware compensation and backward quantization for tiny on-device learning. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 177–191.

Feng Zhu, Ruihao Gong, Fengwei Yu, Xianglong Liu, Yanfei Wang, Zhelong Li, Xiuqi Yang, and Junjie Yan. 2020. Towards unified int8 training for convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1979.

A Evaluation

Quantization-Aware Training (QAT) differs from post-training quantization (PTQ) in that quantization operators are inserted into the optimization loop. As a result, evaluating QAT is inherently multi-dimensional: a method must (i) achieve the desired task quality at a target precision, (ii) exhibit stable and well-behaved training dynamics despite non-smooth quantization and straight-through estimators (STE), and (iii) satisfy deployment constraints (e.g., integer overflow limits, accumulator precision, or analog non-idealities). To make these objectives explicit, we organize QAT evaluation into structural metrics and functional metrics, and then discuss a cross-cutting, boundary-based view enabled by recent scaling-law analyses (Chen et al., 2025b; Dremov et al., 2025).

Evaluation setup. Many metrics are only meaningful once the quantization configuration is fixed. We thus recommend stating (a) **targets** (weights/activations/gradients/KV-cache and any nonlinear approximations), (b) **granularity** (per-tensor/per-channel/group, and group size if applicable), (c) **integer format and rounding** (symmetric/asymmetric, stochastic/deterministic rounding, clipping rule), and (d) **hardware model vs. measurement** (simulator assumptions, accumulator bit-width, ADC precision, kernel backend, or on-device profiling). When reporting efficiency, also specify sequence length/batch size (for transformers), compiler/kernel stack, and whether comparisons share identical implementations.

Structural metrics vs. functional metrics. Structural metrics diagnose whether QAT behaves as intended during training: (1) quantization distortion (tensor-level and its propagation), (2) optimization mismatch induced by STE and quantization (e.g., oscillations and gradient misalignment), and (3) feasibility under numeric/device constraints (overflow/saturation/non-ideality margins). Functional metrics measure usefulness in deployment: task quality at target precision, training and inference efficiency on the intended system, and robustness to precision, data and system shifts.

A.1 Structural Evaluation

A.1.1 Quantization Error

Structural evaluation often begins with direct measures of how closely quantized tensors approximate their full-precision counterparts.

Tensor-level distortion. Given a full-precision tensor X (weights or activations) and a quantizer $Q(\cdot)$, common proxies include normalized MSE,

$$\mathrm{nMSE}(X) = \frac{\|X - Q(X)\|_F^2}{\|X\|_F^2}, \qquad (10)$$

as well as cosine similarity or layer-wise distortion profiles. Because distortion is typically non-uniform across layers, reporting *per-layer curves* (and summary statistics such as mean/percentiles) is often more informative than a single aggregate value. Many QAT recipes that tune step sizes or clipping (uniform or non-uniform) adopt such reporting to justify improved error profiles (Esser et al., 2020; Bhalgat et al., 2020; Jain et al., 2020; Zhao et al., 2020; Li et al., 2020; Gongyo et al., 2024).

Layer-output and logit drift. Small tensor-level errors can be amplified by nonlinearities and attention. A complementary structural diagnostic is output drift, measured at intermediate block outputs or final logits. Typical choices include cosine similarity/MSE between block outputs, KL divergence between output distributions of a full-precision floating-point reference model (e.g., FP32/BF16) and its quantized counterpart (e.g., INT8/INT4), or the change in evaluation loss on the same batches (e.g., Δcross-entropy or Δperplexity for language models). These drift metrics directly quantify the functional consequence of quantization noise while still serving as structural fidelity proxies. QAT methods for ViTs and LLMs frequently use such analyses to identify sensitive submodules and motivate module-specific treatments (Li et al., 2022; Liu et al., 2024a; Chen et al., 2025a).

A.1.2 Hardware Constraints

A distinctive advantage of QAT is the ability to incorporate deployment constraints into training and to evaluate feasibility explicitly.

Accumulator overflow and saturation risk. On integer hardware, feasibility often hinges on avoiding overflow in limited-precision accumulators. We recommend reporting both (i) *certified* or analytically derived safety margins (when available), and (ii) *empirical* overflow/saturation event rates under representative inputs. Accumulator-aware approaches exemplify this paradigm by deriving bounds and enforcing them through QAT, then validating the observed saturation behavior (Colbert et al., 2023, 2024).

Analog non-idealities and ADC constraints (IMC). For in-memory computing, structural evaluation should quantify discrepancies between ideal and hardware-level outputs as a function of ADC precision, dynamic range, and non-ideal device behavior. Metrics include distributional drift of array outputs, saturation frequency, and error histograms under hardware models. QAT recipes tailored to IMC report these diagnostics to justify robustness to analog constraints (Zhang et al., 2024).

Integer-only training mismatch (TinyML/on-device learning). When training itself must be integer-only, structural metrics can quantify how closely integer-gradient updates track a floating-point optimizer, e.g., by reporting the magnitude of compensation terms or mismatch statistics during training (Zhou et al., 2021).

Dynamic and conditional quantization schedules. When quantization varies across inputs, layers, or timesteps (e.g., instance-aware dynamic quantization or diffusion timestep-dependent quantization), structural reporting should include the distribution of chosen precisions/scales and how distortion varies over the controlling variable (input complexity or timestep). This makes the "moving target" of quantization explicit and comparable across methods (Liu et al., 2022; So et al., 2023; Li et al., 2023; He et al., 2024; Zheng et al., 2024; Liu et al., 2025b; Wang et al., 2025; Zhao et al., 2025).

A.2 Functional Evaluation

Structural diagnostics explain why a QAT recipe behaves as it does, but QAT methods are ultimately judged by end-task outcomes and system-level benefits. We group functional evaluation into task performance at target precision, efficiency during training/inference, and robustness under deployment variation.

A.2.1 Task Performance

Primary task metrics. The dominant functional criterion is the task metric under the intended quantization configuration. Typical reports include ImageNet Top-1/Top-5 (vision), mAP (detection), GLUE/SQuAD/BLEU (NLP), perplexity plus broad zero-/few-shot benchmark suites (LLMs), and FID/IS under a specified sampler/schedule (diffusion). Representative low-bit QAT results are reported across CNN/ViT backbones, encoder models (BERT-like), LLMs, and diffusion models (Esser et al., 2020; Bhalgat et al., 2020; Li et al., 2020; Jain et al., 2020; Zhao et al., 2020; Shen et al., 2019; Zhang et al., 2020; Bai et al., 2021; Kim et al., 2021; Wang et al., 2022; Liu et al., 2024a; Chen et al., 2025a; Li et al., 2023; He et al., 2024; Zheng et al., 2024).

Comparability across configurations. To avoid cherry-picking, we recommend reporting a small grid of standard settings: (i) multiple bit-widths (e.g., INT8, W4A4, W3A3, W2A2 as applicable), (ii) ablations on which tensors are quantized (W-only, A-only, KV-cache), and (iii) granularity choices (per-channel vs. per-tensor vs. group). When distillation or data-free supervision is used, report the teacher metric and the student–teacher gap explicitly (Liu et al., 2024a).

A.2.2 Efficiency

Algorithmic cost proxies. Common proxies include peak memory, wall-clock time per step, GPU hours, and total training FLOPs. For QAT schedules that mix precisions, report compute and the schedule itself (transition points, phase lengths). Studies on scaling and compute-optimality treat compute budget as a first-class metric and report validation loss/perplexity as a function of total FLOPs under alternative precision allocations (Chen et al., 2025b; Dremov et al., 2025; Nielsen et al., 2025).

System measurements. Ultimately, inference benefits must be measured on the intended system: latency, throughput, memory footprint, and energy (when possible). For fair comparison, report hardware model, kernel backend, and input shapes. Mobile/edge evaluations often quantify speedups from packing or integerizing operators (Dong et al., 2023; Li and Gu, 2023), while hardware-aware QAT in IMC or limited-accumulator settings reports energy/throughput under varying device constraints (Zhang et al., 2024; Colbert et al., 2023, 2024). On-device learning evaluates training-time energy and wall-clock feasibility under microcontroller constraints (Zhou et al., 2021).

A.2.3 Robustness

Precision robustness. A basic robustness check is performance across multiple precisions and configurations, revealing whether a recipe overfits to a particular bit-width or quantization target (e.g., W-only vs. W + A). Dynamic quantization methods additionally test whether conditional policies generalize across datasets (Liu et al., 2022).

Data and task shifts. Robustness should include evaluation on distribution shifts or unseen tasks/domains, especially for LLM and PEFT-based QAT where extreme low-bit regimes may disproportionately affect reasoning or safety behavior (Xu et al., 2024; Li et al., 2024b; Jeon et al., 2025; Ke et al., 2024; Lee et al., 2024; Wei et al., 2025; Hu et al., 2025; Bondarenko et al., 2024).

System shifts and implementation sensitivity. Finally, robustness must consider deployment variation: kernel/back-end changes, rounding modes, accumulator precision differences, or analog nonidealities. Reporting sensitivity to such factors helps distinguish recipes that generalize to real deployments from those that overfit a particular simulator or kernel stack (Zhang et al., 2024; Colbert et al., 2023).

Recommended minimal functional protocol. At a minimum, we recommend reporting: (i) the task performance at the target precision, (ii) at least one efficiency metric (either a proxy metric and/or a system-level metric), and (iii) one robustness slice (e.g., across different bit-widths or under a deployment-time distribution shift), with all quantization configuration details clearly specified upfront. Adhering to this reporting protocol is critical for ensuring reproducibility and enables fair and meaningful comparisons across diverse quantization methods, reducing ambiguity in evaluation and strengthening the reliability of empirical conclusions.

Approach	Venue	Centric-Target	Bits	Gran.	Model
TernaryBERT	2020 EMNLP	W	2	Layer, Row	BERT
BinaryBERT	2021 ACL	W	1	Layer, Tensor	BERT
Nagel et al.	2022 ICML	W	3, 4	Tensor	General
A2Q	2023 ICCV	W	5–8	Channel	General
OFQ	2023 ICML	W	2, 3, 4	Row	ViT
DL-QAT	2024 EMNLP	W	3, 4	Group	LLM
QEFT	2024 EMNLP Findings	W	4	Group	LLM
PB-LLM	2024 ICLR	W	1	Column	LLM
QA-LoRA	2024 ICLR	W	4	Group	LLM
LoftQ	2024 ICLR	W	2, 4	Tensor	LLM
EfficientQAT	2025 ACL	W	2	Group	LLM
L4Q	2025 ACL	W	3, 4	Group	LLM
CQPT	2025 ACL Findings	W	1.58	Layer, Token	LLM
LSQ+	2020 CVPR	A	2, 4	Layer	General
BinaryDuo	2020 ICLR	A	1	/	BNN
LLSQ	2020 ICLR	W + A	W4A4, W3	Channel, Layer	General
Jain et al.	2020 MLSys	W + A	W8A8, W4	Tensor	General
Degree-Quant	2021 ICLR	W + A	W8A8, W4	Tensor	GNN
Liu et al.	2022 CVPR	W + A	2–6	Layer, Token	General
OCTAV	2022 ICML	W + A	4–6	Tensor	General
Q-ViT	2022 NeurIPS	W + A	W4A4, W3	/	ViT
nu-LSQ	2024 ACCV	W + A	2, 3, 4	Layer	General
EfficientDM	2024 ICLR Spotlight	W + A	W4A4	Channel, Layer	DM
BiDM	2024 NeurIPS	W + A	W/A 1	Layer	DM
per-embed	2021 EMNLP	W + A	W8A8	Tensor	Transformer
I-BERT	2021 ICML	W + A	8	/	BERT
I-ViT	2023 ICCV	W + A	8	/	ViT
Zhu et al.	2020 CVPR	G	8	Layer	CNN
GradScale	2020 NeurIPS	G	4	Layer	General
Xi et al.	2023 NeurIPS	G	4	Token	Transformer
Chitsaz et al.	2024 EMNLP Findings	G	4, 8	Tensor, Channel, Token	LLM
LLM-QAT	2024 ACL Findings	KV	8, 16	Token	LLM
LR-QAT	2024 arXiv	KV	4, 8	/	LLM

Table 1: Taxonomy of quantization-aware training (QAT) frameworks. Methods are grouped by quantization centric-target (weights, activations, weights+activations, gradients, and KV cache) and summarized by bit-width, granularity, and model.