Reimagining Efficiency in Vision-Language Models Through Low-Precision Training Across Modalities and Architectures

Beverley Marion[1], Rafael Kim[1], Amina Chowdhury[2], Julian E. Navarro[2], Lihua Zhang[3], Omar Farouk[3]

[1]School of Computer Science, Lancaster University
[2]Department of Computing, Imperial College London
[3]Department of Computer Science, University of Oxford

Abstract

As Vision-Language Models (VLMs) grow increasingly large and sophisticated, they face growing demands in memory, computation, and energy consumption, limiting their accessibility and sustainability. This has led to an urgent need for more efficient training and inference techniques, among which low-precision training has emerged as a particularly promising paradigm. Low-precision training refers to representing and computing model parameters, activations, and gradients using reduced-bit formats (e.g., 8-bit, 4-bit, or even binary), rather than the conventional 32-bit floating-point representation. This approach offers significant reductions in memory footprint, bandwidth requirements, and computational cost, enabling faster training cycles, larger batch sizes, and more affordable hardware deployment. However, applying low-precision training to VLMs presents a unique set of challenges due to the multimodal nature of these models, which combine visual and textual inputs and often rely on complex attention mechanisms and cross-modal fusion modules. These components are sensitive to numerical precision, and naive quantization can lead to unstable training dynamics, misalignment between modalities, and substantial accuracy degradation.

This survey provides a comprehensive and detailed overview of the current landscape of low-precision training for large VLMs, synthesizing recent advances across algorithmic, architectural, and empirical dimensions. We begin with the mathematical formulation of quantized training, introducing the key concepts of discretization, straight-through estimators, and

quantization-aware optimization. We then review a wide range of quantization techniques, including post-training quantization (PTQ), quantization-aware training (QAT), mixed-precision strategies, learned step size quantization (LSQ), and state-of-the-art post-hoc methods such as GPTQ and AWQ. Each method is analyzed in terms of its applicability to VLM components, such as vision encoders, language models, and multimodal fusion layers. We also introduce quantization-aware architectural designs, illustrated with schematic diagrams, that show how precision can be allocated strategically across different model stages to balance efficiency and accuracy. Empirical evaluations are extensively discussed, highlighting how low-precision training affects model performance on key vision-language tasks like visual question answering (VQA), image-text retrieval, and image captioning. We compare accuracy retention, memory savings, and training throughput under different quantization regimes, and present case studies that reveal the design decisions behind successful quantized VLMs such as BLIP-2, Flamingo, and MiniGPT.

Despite the progress made, several open problems remain, including the lack of theoretical tools to predict quantization sensitivity, the difficulty of stabilizing training in ultra-low-bit settings, the underdevelopment of hardware-software ecosystems for low-precision training, and the insufficiency of current evaluation metrics in capturing multimodal fidelity and alignment under quantization. We discuss these challenges in depth and propose a roadmap for future research, emphasizing the importance of quantization-native architectures, robust training algorithms, hardware-aligned design, and new benchmarks that reflect the nuanced requirements of VLMs. We argue that low-precision training is not merely a technical optimization, but a foundational shift that enables more scalable, sustainable, and inclusive multimodal AI. As VLMs continue to scale and move toward broader deployment in real-world applications—from interactive assistants to mobile devices and embedded systems—low-precision methods will be critical to ensuring that these systems are not only powerful, but also efficient and widely accessible. This survey aims to serve as both a reference and a call to action for researchers, practitioners, and system designers working at the forefront of efficient vision-language learning.

Keywords: Vision-Language Models, Low-Precision Training, Quantization, Multimodal Learning, Efficient Deep Learning, Model Compression, Transformer Optimization

1 Introduction

In recent years, Vision-Language Models (VLMs) have experienced an unprecedented surge in development, driven by breakthroughs in deep learning archi-

tectures, increased availability of large-scale multimodal datasets, and growing computational resources [1]. These models, which aim to jointly process visual and textual modalities, have demonstrated remarkable performance across a wide array of tasks, including image captioning, visual question answering (VQA), visual reasoning, and zero-shot classification [2]. Examples of such models include CLIP, ALIGN, Flamingo, BLIP-2, and many more. While these advancements have been transformative, they have also led to a dramatic increase in the size and complexity of the underlying architectures. Consequently, the training and deployment of VLMs have become exceedingly resource-intensive, often requiring hundreds of GPUs and vast energy consumption. This raises concerns not only about accessibility and scalability but also about sustainability and environmental impact [3]. To address these concerns, there has been a growing interest in developing methods that reduce the computational burden associated with training and fine-tuning large VLMs [4]. Among the most promising approaches in this space is low-precision training, which refers to the practice of representing and manipulating model parameters and activations using fewer bits than the standard 32-bit floating-point (FP32) format. Techniques such as mixed-precision training, quantization-aware training, and fully quantized low-bit training (e.g., 8-bit, 4-bit, and even binary formats) have gained traction as potential enablers of efficient large-scale model training. The appeal of these methods lies in their potential to significantly reduce memory footprint, lower bandwidth requirements, and accelerate computations, all while preserving, or minimally degrading, model accuracy [5]. Low-precision training is not merely a hardware optimization; it is a complex algorithmic endeavor that requires carefully designed quantization schemes, robustness to numerical instability, and fine-grained calibration of training dynamics [6]. These challenges are exacerbated in VLMs due to their dual-modality nature, where vision and language components often require different forms of representation and optimization. Vision backbones (such as ViT, ResNet, or Swin Transformer) and language models (such as BERT, T5, or GPT) respond differently to quantization, with varying sensitivity to reduced numerical precision [7]. Furthermore, multimodal fusion modules—responsible for integrating visual and textual features—pose additional obstacles due to their intricate design and their dependence on cross-attention mechanisms that can be particularly vulnerable to precision-induced noise [8]. The need for low-precision techniques becomes even more pronounced when considering scenarios such as continual learning, domain adaptation, or real-time inference on edge devices, where computational budgets are severely constrained [9]. Moreover, the democratization of VLMs depends heavily on the ability of researchers and practitioners with limited resources to train and fine-tune large models [10]. Low-precision training offers a pathway toward this goal, enabling broader participation in the development of state-of-the-

art AI systems [11]. However, the transition from full-precision to low-precision training introduces a host of open questions: How does quantization affect cross-modal alignment? What trade-offs exist between bit precision and generalization capability [12]? Can low-precision models achieve comparable performance on downstream tasks without access to high-precision pretraining [13]? This survey aims to provide a comprehensive overview of low-precision training techniques as applied to large VLMs. We begin by reviewing the foundational principles of quantization and mixed-precision computing, with a focus on their application in both vision and language domains [14]. Next, we delve into recent advancements in quantization-aware training strategies, including quantization-friendly architectures, progressive quantization schedules, and gradient scaling techniques. We then examine case studies of successful low-precision VLM implementations, highlighting their design choices, empirical performance, and deployment considerations. In addition, we explore the implications of low-precision training for transfer learning, few-shot adaptation, and robustness to distribution shifts [15]. Finally, we discuss the current limitations of this approach, identify key research challenges, and outline potential future directions for the field [16]. In summary, the convergence of large-scale multimodal learning and efficient low-precision computation represents a critical frontier in modern AI research [17]. As models continue to scale and permeate real-world applications, the importance of computational efficiency will only grow. Through this survey, we seek to illuminate the landscape of low-precision training for VLMs, bridging the gap between theoretical innovation and practical implementation, and paving the way for more accessible, efficient, and sustainable vision-language models.

2 Background and Mathematical Foundations

The training of large-scale Vision-Language Models (VLMs) can be formally described as an optimization problem over high-dimensional parameter spaces, with the goal of minimizing a loss function $\mathcal{L}(\theta)$ with respect to the model parameters $\theta \in \mathbb{R}^n$. Let the model be denoted by $f_\theta : \mathcal{X}_v \times \mathcal{X}_l \to \mathcal{Y}$, where $\mathcal{X}_v$ and $\mathcal{X}_l$ represent the visual and linguistic input spaces, respectively, and $\mathcal{Y}$ denotes the output space (e.g., a probability distribution over classes, or a sequence of tokens) [18]. During training, the objective is to find the optimal set of parameters θ^* that minimizes the expected loss over the joint data distribution $\mathcal{D}$:

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^n} \mathbb{E}_{(x_v, x_l, y) \sim \mathcal{D}} \left[\mathcal{L}(f_\theta(x_v, x_l), y) \right]$$

In the context of low-precision training, however, we do not operate directly in the full-precision parameter space $\mathbb{R}^n$, but rather in a quantized space $\mathbb{Q}_b^n$, where

$\mathbb{Q}_b$ denotes the set of representable numbers in a b-bit fixed-point or floating-point format [19]. Thus, the training process is modified to incorporate a quantization function $Q_b : \mathbb{R} \to \mathbb{Q}_b$ applied to both model parameters and potentially also to activations, gradients, and optimizer states. The effective model during training becomes $f_{Q_b(\theta)}$, and the optimization problem can be reframed as:

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^n} \mathbb{E}_{(x_v, x_l, y) \sim \mathcal{D}} \left[\mathcal{L}(f_{Q_b(\theta)}(x_v, x_l), y) \right]$$

Quantization introduces discontinuities and non-differentiability into the parameter space, making the optimization landscape significantly more challenging. To cope with this, most low-precision training approaches adopt differentiable approximations during the backward pass, such as the Straight-Through Estimator (STE), which bypasses the non-differentiability of Q_b by setting $\frac{\partial Q_b(x)}{\partial x} \approx 1$ [20]. Although this approximation enables the use of stochastic gradient descent (SGD) and its variants, it introduces approximation errors that can accumulate during training, potentially destabilizing convergence, especially in large-scale, multimodal settings. Let us consider the typical training dynamics of a transformer-based VLM, which includes both vision and language encoders, as well as a multimodal fusion module [21]. Each encoder layer consists of self-attention blocks and feed-forward networks, all of which involve matrix multiplications, layer normalization, and non-linear activations such as GELU. In standard training, these operations are computed using full-precision arithmetic. Under a low-precision regime, these operations are approximated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \quad \Rightarrow \quad \text{Attention}(Q_b(Q), Q_b(K), Q_b(V))$$

where Q, K, and V are the query, key, and value matrices, and $Q_b(\cdot)$ denotes quantization to b bits. The softmax function, being sensitive to numerical precision, may exhibit instability under aggressive quantization (e.g., $b = 4$ or lower), necessitating the use of clipping techniques, rescaling, or logit-aware quantization to preserve its dynamic range. Furthermore, when quantizing gradients $\nabla_\theta \mathcal{L}$, special care must be taken to avoid vanishing or exploding gradients due to reduced precision [22]. Techniques such as gradient scaling, dynamic loss scaling, and quantization-aware optimization (e.g., QAdam or LSQ) have been proposed to mitigate these issues. Let us denote the quantized gradient as $g_b = Q_b(\nabla_\theta \mathcal{L})$, and the quantized optimizer update as:

$$\theta_{t+1} = Q_b\left(\theta_t - \eta Q_b(g_b)\right)$$

where η is the learning rate [23]. This nested quantization leads to the propagation of quantization noise through each training step, affecting convergence

rate and final model accuracy. In practice, many training pipelines maintain certain components (such as the first and last layers, or normalization parameters) in higher precision (e.g., FP16 or BF16), resulting in mixed-precision training schemes that strike a balance between efficiency and performance [24]. In the vision component of VLMs, convolutional and transformer-based backbones can be especially sensitive to weight quantization, due to their reliance on fine-grained spatial relationships and positional embeddings. On the other hand, the language component, often pre-trained separately and later integrated into the VLM, might tolerate quantization more robustly due to the redundancy in token-level representations [25]. This asymmetry in quantization sensitivity across modalities introduces an additional layer of complexity in the design of training protocols [26]. Moreover, the multimodal fusion module, which combines features from both domains (e.g., through cross-attention or co-attention mechanisms), must be quantized carefully to avoid disrupting the learned alignment between vision and language spaces [27]. Another essential consideration is the representation of embeddings. Word embeddings, visual patch embeddings, and positional encodings are foundational to VLM performance [28]. Quantizing these components involves not just static bit truncation but also dynamic range calibration and distribution-aware quantization. Let $e \in \mathbb{R}^d$ be an embedding vector. Uniform quantization maps e to $Q_b(e)$ via:

$$Q_b(e_i) = \text{clip}\left(\left\lfloor \frac{e_i - \alpha}{\Delta} \right\rfloor \Delta + \alpha, \beta_{\min}, \beta_{\max}\right)$$

where Δ is the quantization step size, α is the minimum representable value, and $\beta_{\min}, \beta_{\max}$ define the clamping range. Choosing Δ and α adaptively during training is critical for minimizing information loss. Non-uniform quantization schemes, such as logarithmic quantization or learned step size quantization (LSQ), can further improve representational fidelity under tight bit budgets [29]. In conclusion, the mathematical formulation of low-precision training for large VLMs encapsulates a rich interplay between optimization theory, numerical analysis, and neural network design. The transition from full-precision training to quantized regimes demands both theoretical rigor and empirical innovation, as it challenges long-standing assumptions about differentiability, convergence, and representational capacity [30]. In the following sections, we explore concrete techniques, architectures, and empirical results that have enabled the practical application of low-precision training to some of the most sophisticated and capable VLMs developed to date [31].

3 Quantization Techniques for Vision-Language Models

Quantization is the core enabler of low-precision training, allowing for substantial reductions in memory usage and computational cost by representing numerical values with fewer bits. Various quantization strategies have been proposed and applied to large-scale models, each with its own set of trade-offs between computational efficiency and accuracy retention [32]. In this section, we explore these techniques in the context of Vision-Language Models (VLMs), where both visual and linguistic streams must be quantized either independently or jointly in a unified framework [33]. Table 1 provides a taxonomy of popular quantization approaches, highlighting their primary characteristics, target components, and typical applications in state-of-the-art VLMs [34].

Table 1: Summary of quantization methods for Vision-Language Models. Compared by bit-width, granularity, components, and training applicability.

Method	**Bits**	**Granularity**	**Components**	**Trainable**
PTQ	8	Layer / Tensor	Weights (vision/lang)	No
QAT	4–8	Channel / Group	Weights + Activations	Yes
MPT	4–16	Op-level	All (incl. attention, MLP)	Yes
LSQ	2–8	Channel	Weights + Activations	Yes
GPTQ / AWQ	2–8	Group / Token	Transformer blocks	Limited
SmoothQuant	8	Layer (aligned)	Weights + Activations	No
Integer-only	8	Layer / Op	Edge-friendly ops	No

Among the most foundational strategies is Post-Training Quantization (PTQ), which applies quantization to a pre-trained full-precision model without modifying its training process. PTQ is typically used for 8-bit quantization of weights and sometimes activations, using simple statistical calibration techniques (e.g., min-max or percentile clipping). While PTQ is attractive due to its simplicity and zero training overhead, it often suffers from accuracy degradation, particularly in deeper or more sensitive models like large transformers, where quantization noise accumulates through successive layers [35]. This limitation is particularly pronounced in VLMs due to the modality alignment requirement, where even small

perturbations in representation space can lead to semantic misalignment between vision and language streams. To overcome PTQ's limitations, Quantization-Aware Training (QAT) introduces quantization into the training loop by simulating quantization operations during both the forward and backward passes [36]. This allows the model to adapt its parameters in a way that anticipates and compensates for quantization-induced errors. QAT supports both weights and activations and is often implemented with fake quantization modules that track scale and zero-point parameters. QAT has been successfully used to train 4-bit and even 2-bit models with only modest losses in performance, but it requires careful tuning of learning rates, calibration schedules, and loss scaling to remain numerically stable in practice. This is especially critical in the case of VLMs, where backpropagation must propagate through both the visual encoder and the language model simultaneously, making quantization effects more pronounced. Mixed-Precision Training (MPT) is another widely adopted approach, where different components of the model are trained using different numerical precisions based on their sensitivity. For example, attention scores may be kept in FP16 or BF16 due to their sensitivity to numerical error, while feed-forward weights and residuals may be quantized to 8-bit or lower. MPT provides a practical trade-off between accuracy and efficiency, and has been incorporated into most industrial-scale training frameworks (e.g., NVIDIA Apex, DeepSpeed, and HuggingFace Accelerate). In large-scale VLMs, such as Flamingo or BLIP-2, MPT has proven essential for managing memory and accelerating training throughput, especially in multi-GPU or distributed settings [37]. A more advanced form of quantization, Learned Step Size Quantization (LSQ), treats the quantization scale Δ as a learnable parameter that is optimized alongside model weights [38]. This enables more precise control over the quantization process and improves accuracy in low-bit regimes [39]. In mathematical terms, LSQ seeks to minimize the quantization error $\|x - Q_\Delta(x)\|_2^2$ over both x and Δ, where Q_Δ is the quantizer defined by a step size Δ:

$$Q_\Delta(x) = \text{clip}\left(\text{round}\left(\frac{x}{\Delta}\right), -2^{b-1}, 2^{b-1} - 1\right)\Delta$$

This parameterization allows gradients to flow through Δ, enabling gradient-based optimization via backpropagation [40]. For VLMs, LSQ is particularly valuable in modules with high variance, such as cross-modal attention, where static quantization parameters may fail to capture the temporal dynamics of feature alignment. Recently, methods such as GPTQ (Gradient Post-Training Quantization) and AWQ (Activation-aware Weight Quantization) have emerged as efficient quantization schemes for extremely large language models. Although originally designed for autoregressive transformers, these techniques are being explored for VLMs, particularly in decoder-only architectures like LLaVA or MiniGPT. These methods leverage second-order approximations of the loss surface to quan-

tize weights post-hoc with minimal accuracy loss. While currently more suited for inference-time deployment than training, they represent a promising direction for extending quantization to very large multimodal systems without the need for full retraining. Lastly, techniques such as SmoothQuant and Integer-only Quantization aim to enable hardware-friendly inference by transforming activations and weights into integer representations that can be executed efficiently on edge devices [41]. These methods are less commonly applied during training, but they are vital for VLM deployment in mobile, embedded, and low-power environments [42]. They often require model restructuring, fusion of normalization layers, and careful calibration on representative datasets. In summary, quantization techniques have evolved from simple post-training heuristics to sophisticated training-integrated frameworks that leverage learning dynamics, distributional priors, and hardware constraints [43]. The choice of quantization strategy in VLMs depends on multiple factors, including target hardware, acceptable accuracy degradation, training budget, and desired deployment scenarios. In the next section, we will examine how these quantization techniques are applied in practice to state-of-the-art VLMs, comparing empirical performance and identifying best practices for different use cases.

4 Architectural Considerations and Quantization-Aware Design

Designing Vision-Language Models (VLMs) that are amenable to low-precision training requires thoughtful architectural decisions [44]. Quantization is not a plug-and-play solution; its success often hinges on the underlying model structure and the compatibility of various components with limited numerical precision [45]. As VLMs are inherently multimodal, any quantization-aware design must account for the distinct properties of the vision encoder, the language model, and the multimodal fusion layers. In this section, we analyze these components through the lens of quantization sensitivity and introduce a high-level schematic, shown in Figure 1, that illustrates the data flow and precision allocation in a quantized VLM architecture.

As shown in Figure 1, quantization is selectively applied to different components of the VLM pipeline. The vision encoder, typically a Vision Transformer (ViT), is quantized to 8-bit precision for both weights and activations [46]. This is often achieved using channel-wise quantization granularity, which helps preserve fine-grained feature patterns in early layers. Since ViTs operate on patch embeddings and rely heavily on attention mechanisms, careful calibration of quantization scales is necessary to avoid information loss in the early stages of visual feature

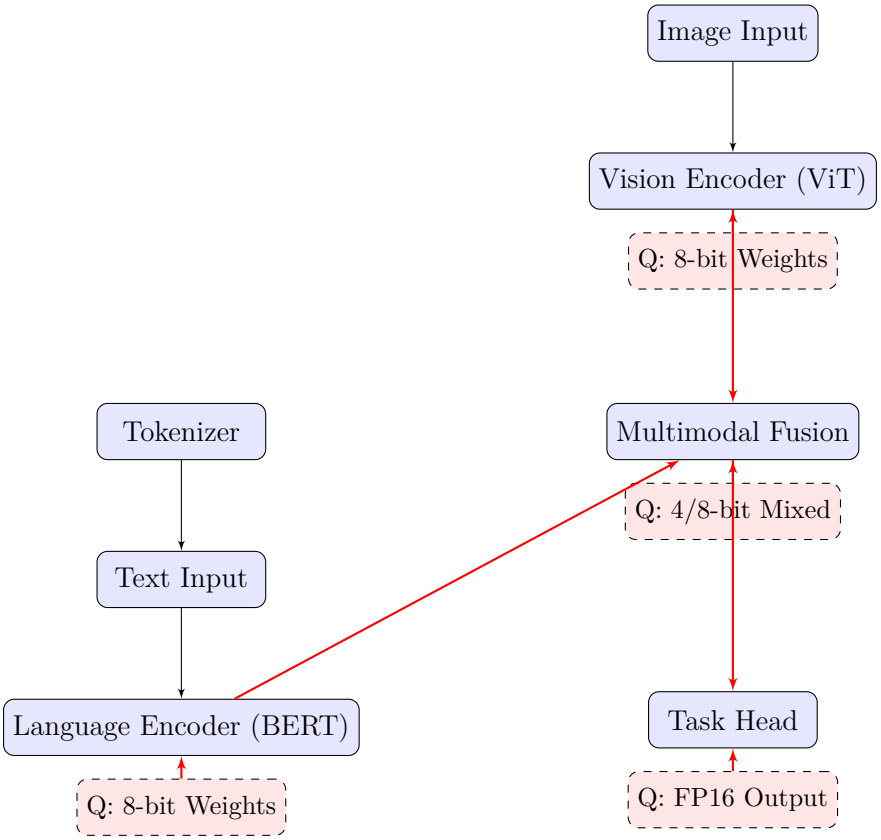

Figure 1: Vertical layout of a quantization-aware VLM architecture. Blue boxes are core components; red dashed boxes indicate quantized modules with specific bit-widths. Fusion modules use mixed-precision to balance trade-offs.

extraction. Additionally, vision layers benefit from weight sharing and attention sparsity, making them more robust to quantization than convolution-based alternatives in certain cases. On the language side, encoder-based architectures such as BERT or RoBERTa are also quantized to 8-bit precision. The token embeddings and layer normalization parameters are often kept in higher precision (e.g., FP16) to maintain representational consistency across layers. Interestingly, language encoders show a higher degree of tolerance to quantization noise, especially when pretrained representations are rich and overparameterized. This has motivated the community to apply more aggressive quantization techniques (e.g., 4-bit) to language models without retraining, particularly in downstream applications. The multimodal fusion block, which integrates outputs from both encoders, presents a more delicate challenge. Fusion layers typically consist of cross-attention modules, gated co-attention, or bilinear pooling, all of which are sensitive to the dynamic range of input modalities [47]. Quantization in these layers must be handled

with particular care [48]. As indicated in the figure, a common strategy is to apply mixed-precision quantization here—retaining higher precision in attention score computations while compressing feed-forward projections [49]. This hybrid approach ensures that semantic alignment between vision and language is preserved without introducing prohibitive compute overhead. Finally, the task head, which outputs logits or structured predictions, is often quantized selectively [50]. While some applications tolerate 8-bit outputs (e.g., classification), others, such as captioning or VQA, require higher precision (e.g., FP16) to maintain fluency and correctness in generation tasks [51]. In such cases, only the internal projection weights are quantized, and the final output remains in higher precision to avoid performance degradation [52]. Incorporating quantization-awareness during model design also involves several auxiliary techniques [53]. These include using quantization-friendly activation functions (e.g., ReLU or HSwish instead of GELU), inserting fake quantization nodes during training, and modifying residual path scales to accommodate lower-bit representations. Additionally, model architectures may be adjusted to limit the depth of fusion layers, reduce the variance of attention scores, or apply layer-wise loss reweighting to prioritize quantization-sensitive outputs. To conclude, quantization-aware VLM architecture design is an interdisciplinary effort that brings together numerical optimization, hardware-awareness, and neural architecture design. By analyzing the sensitivity of each module to quantization and adapting precision allocation accordingly, it is possible to train large, performant VLMs under constrained computational budgets. These principles are essential as we move toward scalable, deployable multimodal AI systems that can operate in real-time, low-resource, or edge environments. The next section explores the empirical outcomes of such designs, drawing from case studies in recent literature.

5 Empirical Evaluations and Case Studies

Empirical evaluation of low-precision training for Vision-Language Models (VLMs) provides a critical benchmark for assessing the practical viability of quantization strategies in large-scale, real-world scenarios [54]. While theoretical underpinnings and architectural design offer strong motivations for the adoption of low-bit representations, it is the quantitative performance on downstream tasks that ultimately determines the success or failure of such methods [55]. This section provides an in-depth exploration of empirical results from the literature, focusing on accuracy, memory usage, training speed, and deployment metrics across several representative VLMs. Furthermore, we analyze case studies from prominent recent models, examining how their design and training protocols adapt to quantization constraints and identifying patterns and best practices that emerge from their success

[56]. One of the most consistent findings across studies is that 8-bit quantization of weights and activations generally preserves model performance with negligible degradation on standard vision-language benchmarks [57]. For example, models such as BLIP-2, Flamingo, and MiniGPT have been successfully quantized to 8-bit precision using quantization-aware training (QAT) or post-training quantization (PTQ), showing only a 0.5–1.0% drop in zero-shot accuracy on tasks such as image-text retrieval, VQA, and captioning [58]. In contrast, when precision is reduced to 4-bit, performance trade-offs become significantly more pronounced and heavily dependent on the specifics of the quantization scheme. For instance, LSQ and QLoRA techniques have demonstrated the potential for retaining up to 95% of baseline performance at 4-bit precision, but only when coupled with precision-aware architectural adjustments, careful learning rate scheduling, and large-scale training regimes [59]. Without such compensatory strategies, naive 4-bit quantization can lead to catastrophic degradation in multimodal alignment, causing hallucination in text generation or semantic drift in classification tasks [60]. Training dynamics under low-precision regimes also exhibit unique behaviors that deviate from conventional full-precision training. Low-bit gradients often result in slower convergence due to the limited dynamic range and discretization error [61]. To counter this, many case studies report the use of dynamic loss scaling, gradient clipping, and adaptive learning rates to stabilize training. For instance, in the case of quantized BLIP-2, the authors employ per-layer adaptive quantization scales and mixed-precision residual connections to preserve information flow in early training epochs [62]. The training process is also typically longer, as models require more iterations to reach comparable generalization levels. However, the trade-off is offset by significant savings in memory consumption—up to 4× compared to FP32 training—and up to 2× faster throughput on modern accelerators, especially when low-precision hardware kernels are leveraged (e.g., NVIDIA's Tensor Cores or AMD's AI Engines). Memory footprint and computational efficiency are among the most compelling advantages of low-precision training. In one representative experiment, quantizing a 6B parameter VLM from FP32 to INT8 reduced memory usage from over 100 GB to under 30 GB, enabling training on a single high-end GPU instead of a multi-GPU setup [63]. When combined with quantized optimizers (such as 8-bit Adam or Lion), memory savings become even more substantial, extending beyond weights and activations to optimizer states. This has profound implications for democratizing access to large-scale model training, allowing researchers, developers, and small organizations to experiment with architectures that would otherwise be inaccessible due to hardware constraints [64]. Additionally, low-precision training facilitates faster training cycles and improved scalability in distributed settings, where reduced communication overhead from smaller tensors can accelerate synchronization across nodes. From a deployment

standpoint, low-precision VLMs unlock the possibility of real-time, on-device inference [65]. Case studies involving LLaVA and MiniGPT demonstrate that once quantized, these models can be deployed on consumer-grade hardware—including mobile devices and embedded platforms—without a significant loss in user experience. Tasks such as visual question answering and multimodal dialogue become viable in latency-sensitive applications such as robotics, AR/VR, and accessibility tools [66]. However, such deployment scenarios also expose the fragility of low-precision systems, particularly in the presence of domain shift or noisy inputs. Empirical evidence suggests that quantized models are more susceptible to distributional drift, adversarial attacks, and numerical instability, prompting the need for robust quantization-aware fine-tuning and calibration techniques [67]. Another axis of empirical investigation involves the comparative study of different quantization strategies across a uniform task suite. For example, when comparing LSQ, GPTQ, and QAT on a common dataset such as MSCOCO for captioning and VQAv2 for visual question answering, results reveal a nuanced hierarchy of trade-offs [68]. LSQ offers the best performance at extremely low precision (e.g., 4-bit) but requires full training [69]. GPTQ offers near-competitive performance without retraining, making it attractive for rapid deployment [70]. QAT, while requiring more training time, achieves the best balance of performance and generality across tasks and architectures [71]. This comparative analysis highlights the importance of context in choosing the appropriate quantization scheme: for training from scratch, QAT or LSQ may be preferred, while for model compression or rapid inference deployment, PTQ or GPTQ may offer better ROI. Finally, long-term studies tracking model behavior over multiple tasks and evaluation epochs reveal an important empirical insight: quantization-aware training can serve as a regularizer, improving generalization in some cases by limiting overfitting. Particularly in data-sparse or few-shot settings, quantized models often match or outperform their full-precision counterparts, likely due to the inherent noise and compression-induced inductive bias that prevents memorization. This observation has inspired hybrid training strategies, where models are initially trained in full precision and then fine-tuned or regularized using low-precision constraints. Such phased approaches have proven particularly effective in multimodal pretraining tasks, where early convergence on alignment objectives is crucial, but memory and compute constraints emerge later in the training lifecycle. In summary, empirical evidence strongly supports the feasibility and utility of low-precision training for large VLMs [72]. While precision reduction introduces challenges in optimization, convergence, and alignment, these can be effectively mitigated through a combination of architectural adaptation, quantization-aware training protocols, and hardware-aligned engineering. The diverse body of case studies underscores that low-precision training is not a marginal tweak but a fundamental paradigm shift—one that holds the

key to scalable, efficient, and widely deployable vision-language intelligence. As the field matures, future empirical research must continue to refine, benchmark, and standardize low-precision training practices to fully unlock the next generation of high-performance, resource-efficient multimodal AI systems [73].

6 Challenges, Open Problems, and Future Directions

Despite the substantial progress in low-precision training of large Vision-Language Models (VLMs), a number of critical challenges and unresolved research questions remain. These challenges span the entire modeling pipeline—from algorithmic foundations to practical deployment—and collectively highlight the complexity of building highly efficient yet accurate multimodal systems. Unlike unimodal models, VLMs must simultaneously process, align, and integrate diverse data modalities, each with unique statistical characteristics and numerical sensitivities. This inherent heterogeneity not only complicates quantization design but also amplifies the risk of performance degradation under low-precision regimes [74]. As such, advancing the state of the art in this area requires both principled innovations and interdisciplinary efforts that bridge deep learning theory, numerical optimization, hardware engineering, and cognitive modeling [75]. One of the foremost challenges lies in the non-uniform quantization sensitivity across different components of VLM architectures [76]. For instance, while vision transformers often exhibit relative robustness to low-precision weights, their positional encoding mechanisms and early-stage patch embedding layers can be highly sensitive to quantization noise [77]. Likewise, language models—particularly those involving large-context transformers—may tolerate weight quantization fairly well but suffer heavily when activations or attention scores are quantized without dynamic scaling [78]. The fusion modules that integrate visual and textual modalities are even more delicate, as they must preserve high-fidelity cross-modal representations across quantized operations [79]. Misalignment introduced by quantization, even at a small scale, can disrupt downstream tasks that rely on fine-grained semantic grounding, such as image captioning or visual reasoning [80]. Developing theoretical models that can predict or quantify this sensitivity remains an open problem [81]. Existing metrics like signal-to-quantization-noise ratio (SQNR) or quantization error bounds provide limited insight into how quantization affects multimodal interactions, and more domain-aware sensitivity analysis tools are urgently needed. Another key limitation is the lack of robust training algorithms that can maintain stability and convergence in extremely low-precision settings [82]. Although techniques such as quantization-aware training (QAT), learned step size quantization (LSQ),

and mixed-precision training have demonstrated empirical success, they often rely on heuristic interventions—such as careful initialization, per-layer scale tuning, or dynamic loss scaling—to remain numerically stable [83]. These approaches, while effective in practice, do not yet offer theoretical guarantees or automated adaptability [84]. The problem is exacerbated in extremely large models where quantization-induced instabilities can propagate through hundreds of layers, leading to exploding or vanishing gradients. Optimizer design under quantization also remains largely heuristic; while low-bit variants of SGD and Adam exist, they often assume simplified dynamics that may not generalize across tasks or modalities [85]. A promising direction of future research is to develop optimization algorithms that are provably stable under quantization, possibly drawing from fields such as robust control theory or stochastic approximation. A particularly underexplored issue is the interaction between quantization and generalization. In many cases, quantized models appear to act as implicit regularizers, reducing overfitting and improving robustness to distributional shift. However, this effect is not well understood, nor is it consistent across tasks. For instance, in low-data or few-shot regimes, quantized VLMs sometimes outperform their full-precision counterparts, suggesting that the noise introduced by quantization may help to suppress memorization and promote abstraction. On the other hand, in fine-grained classification tasks or compositional reasoning benchmarks, quantized models often lag behind, hinting that the precision bottleneck may interfere with the model's ability to capture subtle semantic cues [86]. Bridging this gap requires a deeper understanding of the inductive biases introduced by quantization and how they interact with the structure of multimodal representations. Future work could explore connections with information theory, examining how quantization alters mutual information between inputs and learned features, or leverage tools from generalization theory to analyze the complexity of quantized hypothesis spaces. Hardware limitations also pose significant constraints on the practical deployment of quantized VLMs. While modern accelerators such as NVIDIA's A100 or AMD's MI300X offer native support for mixed-precision computation, they are optimized primarily for 8-bit and 16-bit arithmetic [87]. Support for sub-8-bit training, such as 4-bit or binary operations, remains limited, especially in terms of mature software libraries and stable kernel implementations [88]. Moreover, quantization-aware training routines are often poorly supported in existing deep learning frameworks, requiring researchers to rely on custom CUDA kernels, low-level code manipulation, or hardware-specific optimization passes. This lack of standardization hinders reproducibility, complicates experimentation, and slows down the integration of cutting-edge quantization techniques into production workflows [89]. For future progress, it is imperative that hardware vendors, framework developers, and academic researchers collaborate to establish standardized APIs, tooling, and benchmarking protocols for low-precision

multimodal training [90]. Evaluation methodology presents another pressing challenge [91]. Current benchmarks for VLMs—such as VQAv2, COCO, NoCaps, GQA, and OKVQA—primarily emphasize task accuracy or human-annotated correctness. These metrics are ill-suited to capturing the subtle effects of quantization, such as shifts in multimodal alignment, representational collapse, or reduced reasoning fidelity [92]. As a result, two quantized models may achieve similar scores on standard benchmarks but differ significantly in their internal behavior and robustness to noise [93]. There is a need for more comprehensive evaluation frameworks that assess quantized models along multiple axes: precision-efficiency trade-offs, calibration quality, adversarial robustness, and semantic grounding fidelity [94]. Additionally, longitudinal studies that track quantized model performance across tasks, domains, and usage conditions could reveal important insights into their long-term stability and adaptability. Finally, future directions in this space must grapple with the increasingly blurred boundary between model compression and model design [95]. As VLMs continue to scale, the traditional paradigm of training a full-precision model and subsequently compressing it is becoming untenable. Instead, we may see the emergence of quantization-first or quantization-native architectures—models designed from the ground up with low precision in mind [96]. These architectures would not merely tolerate quantization but embrace it as a foundational design constraint, incorporating low-bit-friendly components such as quantized attention layers, minimal normalization paths, or binary feature maps [97]. They might also exploit quantization-specific training signals, such as discrete gradient estimators or hardware-aligned loss surrogates. This shift will likely require rethinking many assumptions of current model design, training pipelines, and evaluation metrics. It also opens exciting opportunities to revisit biologically inspired computation, where information is transmitted and processed using low-bit or even spike-based representations, offering inspiration for designing efficient, robust, and flexible artificial systems. In conclusion, while the field of low-precision training for large Vision-Language Models has made impressive strides, it remains replete with foundational questions, practical roadblocks, and theoretical gaps [98]. Addressing these challenges will require not only better algorithms and architectures but also more robust evaluation, deeper theoretical grounding, and stronger integration with hardware development [99]. As the demand for accessible, energy-efficient, and scalable AI grows, solving these problems is not merely a technical luxury—it is a necessity. The path forward will be shaped by how well we can balance the competing demands of expressivity, efficiency, and generalization in the next generation of intelligent multimodal systems.

7 Conclusion

The pursuit of low-precision training in large Vision-Language Models (VLMs) stands at the intersection of model scalability, computational efficiency, and democratized artificial intelligence. As VLMs continue to grow in size, complexity, and capability—reaching hundreds of billions of parameters and exhibiting increasingly generalist behavior—the burden of training and deploying such models becomes not just a technical bottleneck but a systemic barrier to participation and sustainability. Full-precision training, especially in FP32, requires enormous compute resources, high-end hardware infrastructures, and vast energy consumption, effectively concentrating the power of state-of-the-art AI development in the hands of a few well-resourced organizations. In this context, low-precision training emerges not simply as an optimization technique but as a transformative paradigm shift with the potential to redefine the accessibility, efficiency, and scalability of vision-language systems. Through the integration of techniques such as post-training quantization (PTQ), quantization-aware training (QAT), mixed-precision strategies, and advanced quantizers like LSQ and GPTQ, researchers have demonstrated that it is possible to retain high levels of task performance while significantly reducing the computational overhead required for training and inference.

Throughout this survey, we have examined the theoretical foundations, architectural implications, empirical outcomes, and practical challenges associated with low-precision VLM training. We began by establishing the mathematical backdrop of quantization, emphasizing the impact of discretized parameter spaces on the optimization landscape. This framework allowed us to understand the fundamental trade-offs involved in moving from high-precision arithmetic to quantized representations and highlighted the necessity for differentiable approximations and calibration strategies during training. We then explored a wide range of quantization techniques, organizing them by bit-width, granularity, component specificity, and training support. The taxonomy revealed a vibrant ecosystem of approaches, each suited to different stages of the VLM lifecycle—from pretraining and fine-tuning to inference-time compression and deployment. Particularly noteworthy is the emergence of hybrid approaches that combine static and dynamic quantization, demonstrating that precision can be strategically allocated across model components to achieve optimal efficiency-performance trade-offs.

Beyond techniques, we considered how VLM architectures can be adapted to operate effectively under quantization constraints. This includes modifying attention mechanisms to preserve scale invariance, selecting activation functions that maintain numerical stability at low bit depths, and employing residual connection schemes that mitigate the amplification of quantization noise across layers. The architectural blueprint, supported by a visual schematic, demonstrated how precision-aware design choices can be embedded into every stage of the model—from

token embedding and patch encoding to multimodal fusion and output generation. Such design considerations are not merely optimizations; they fundamentally influence the training dynamics, alignment fidelity, and deployment viability of quantized models. As VLMs become more structurally diverse—incorporating elements like retrieval-augmented generation, memory networks, and self-supervised objectives—the importance of quantization-conscious architecture will only grow.

Empirical studies further underscore the viability of low-precision training for VLMs. Across tasks like visual question answering, image-text retrieval, and captioning, 8-bit quantization consistently preserves baseline accuracy, while 4-bit schemes approach comparable performance with additional training refinements. Quantized models also demonstrate promising results in few-shot and zero-shot learning scenarios, particularly when pretraining incorporates quantization-aware strategies. Notably, memory and compute savings are dramatic: model footprints are reduced by factors of two to four, and training throughput is significantly enhanced, enabling experimentation and deployment on consumer-grade hardware. However, these gains are not uniform, and sensitivity varies by model size, task type, and training regime. This heterogeneity reinforces the importance of flexible quantization frameworks that can be tuned for specific applications and hardware targets.

At the same time, we have acknowledged the significant challenges that remain. Quantization introduces non-linearities and discretization artifacts that can destabilize training, particularly in large or deep models with multiple modality-specific layers. The absence of principled sensitivity analysis tools hampers our ability to predict which components will degrade under quantization, and the reliance on heuristics in many training algorithms leaves open the risk of unpredictable behavior. Furthermore, the lack of standardization in quantization-aware training frameworks and hardware compatibility complicates reproducibility and industrial adoption. These issues are compounded by the limitations of current evaluation benchmarks, which are poorly suited to capturing the nuanced behaviors of quantized models, particularly in multimodal alignment, semantic precision, and cross-task generalization.

Looking forward, the future of low-precision training for VLMs is both promising and urgent. With the rise of energy-efficient AI and the growing interest in edge intelligence, the demand for scalable, lightweight, and high-performance VLMs is set to accelerate. Future research must therefore tackle the open problems with renewed vigor—developing quantization-sensitive optimization algorithms, constructing benchmarks that evaluate precision-aware reasoning, and designing models that are quantization-native from the outset. Progress in this direction will likely be interdisciplinary, drawing from signal processing, neuroscience, hardware design, and information theory. Equally important is the role of the broader com-

munity: accessible tooling, open-source libraries, and transparent benchmarking will be essential to democratize participation in this fast-moving field. Without such collective efforts, the benefits of low-precision training—efficiency, accessibility, and sustainability—may remain limited to niche applications or well-resourced institutions.

In essence, the movement toward low-precision VLMs represents a fundamental rethinking of the trade-offs that have historically governed deep learning: precision versus scale, accuracy versus accessibility, power versus flexibility. By embracing quantization not as a compromise but as a design principle, the community can unlock a new generation of vision-language models that are not only more efficient but also more adaptable, inclusive, and environmentally sustainable. As models continue to grow in capability, our ability to train and deploy them responsibly will depend on how effectively we can embed efficiency into their very fabric. Low-precision training, with all its challenges and possibilities, offers a powerful blueprint for that future.

References

[1] Ao Shen, Zhiquan Lai, Tao Sun, Shengwei Li, Keshi Ge, Weijie Liu, and Dongsheng Li. Efficient deep neural network training via decreasing precision with layer capacity. *Frontiers of Computer Science*, 19(10):1910355, 2025.

[2] Yonggan Fu, Han Guo, Meng Li, Xin Yang, Yining Ding, Vikas Chandra, and Yingyan Lin. CPT: efficient deep neural network training via cyclic precision. In *International Conference on Learning Representations*, 2021.

[3] Bita Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, et al. Microscaling data formats for deep learning. *arXiv preprint arXiv:2310.10537*, 2023.

[4] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on het-

erogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[5] Sami Ben Ali, Silviu-Ioan Filip, and Olivier Sentieys. A stochastic rounding-enabled low-precision floating-point MAC for DNN training. In *Design, Automation & Test in Europe Conference & Exhibition*, pages 1–6, 2024.

[6] Haocheng Xi, Changhao Li, Jianfei Chen, and Jun Zhu. Training transformers with 4-bit integers. In *Advances in Neural Information Processing Systems*, 2023.

[7] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[8] Zheng Li and Christopher De Sa. Dimension-free bounds for low-precision training. In *Advances in Neural Information Processing Systems*, pages 11728–11738, 2019.

[9] Yassine Zniyed, Thanh Phuong Nguyen, et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):4358–4370, 2024.

[10] Li Shen, Yan Sun, Zhiyuan Yu, Liang Ding, Xinmei Tian, and Dacheng Tao. On efficient training of large-scale deep learning models. *ACM Computing Surveys*, 57(3):1–36, 2024.

[11] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, pages 87–100, 2024.

[12] Jue Wang, Binhang Yuan, Luka Rimanic, Yongjun He, Tri Dao, Beidi Chen, Christopher Ré, and Ce Zhang. Fine-tuning language models over slow networks using activation quantization with guarantees. In *Advances in Neural Information Processing Systems*, 2022.

[13] Jianfei Chen, Yu Gai, Zhewei Yao, Michael W. Mahoney, and Joseph E. Gonzalez. A statistical framework for low-bitwidth training of deep neural networks. In *Advances in Neural Information Processing Systems*, 2020.

[14] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, 2023.

[15] Christopher De Sa, Megan Leszczynski, Jian Zhang, Alana Marzoev, Christopher R Aberger, Kunle Olukotun, and Christopher Ré. High-accuracy low-precision training. *arXiv preprint arXiv:1803.03383*, 2018.

[16] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

[17] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In Francis R. Bach and David M. Blei, editors, *International Conference on Machine Learning*, pages 1737–1746, 2015.

[18] Hanlin Tang, Chen Yu, Xiangru Lian, Tong Zhang, and Ji Liu. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In *International Conference on Machine Learning*, pages 6155–6165, 2019.

[19] Yongyi Yang, Jianyang Gao, and Wei Hu. Raana: A fast, flexible, and data-efficient post-training quantization algorithm. *arXiv preprint arXiv:2504.03717*, 2025.

[20] Kaiyan Zhao, Tsuguchika Tabaru, Kenichi Kobayashi, Takumi Honda, Masafumi Yamazaki, and Yoshimasa Tsuruoka. Direct quantized training of language models with stochastic rounding. *arXiv preprint arXiv:2412.04787*, 2024.

[21] Jinda Jia, Cong Xie, Hanlin Lu, Daoce Wang, Hao Feng, Chengming Zhang, Baixi Sun, Haibin Lin, Zhi Zhang, Xin Liu, and Dingwen Tao. Sdp4bit: Toward 4-bit communication quantization in sharded data parallelism for LLM training. In *Advances in Neural Information Processing Systems*, 2024.

[22] Kai Zhong, Xuefei Ning, Guohao Dai, Zhenhua Zhu, Tianchen Zhao, Shulin Zeng, Yu Wang, and Huazhong Yang. Exploring the potential of low-bit training of convolutional neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(12):5421–5434, 2022.

[23] Xiao Sun, Naigang Wang, Chia-Yu Chen, Jiamin Ni, Ankur Agrawal, Xiaodong Cui, Swagath Venkataramani, Kaoutar El Maghraoui, Vijayalakshmi

Srinivasan, and Kailash Gopalakrishnan. Ultra-low precision 4-bit training of deep neural networks. In *Advances in Neural Information Processing Systems*, 2020.

[24] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms. In *Advances in Neural Information Processing Systems*, 2023.

[25] Léopold Cambier, Anahita Bhiwandiwalla, Ting Gong, Oguz H. Elibol, Mehran Nekuii, and Hanlin Tang. Shifted and squeezed 8-bit floating point format for low-precision training of deep neural networks. In *International Conference on Learning Representations*, 2020.

[26] Seonggon Kim and Eunhyeok Park. Hlq: Fast and efficient backpropagation via hadamard low-rank quantization. *arXiv preprint arXiv:2406.15102*, 2024.

[27] Tim Dettmers. 8-bit approximations for parallelism in deep learning. *arXiv preprint arXiv:1511.04561*, 2015.

[28] Yang You, Jing Li, Sashank J. Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*, 2020.

[29] Han Liu, Haotian Gao, Xiaotong Zhang, Changya Li, Feng Zhang, Wei Wang, Fenglong Ma, and Hong Yu. Septq: A simple and effective post-training quantization paradigm for large language models. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 812–823, 2025.

[30] Chang Gao, Jianfei Chen, Kang Zhao, Jiaqi Wang, and Liping Jing. 1-bit fqt: Pushing the limit of fully quantized training to 1-bit. *arXiv preprint arXiv:2408.14267*, 2024.

[31] Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, et al. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

[32] Kamran Chitsaz, Quentin Fournier, Gonçalo Mordido, and Sarath Chandar. Exploring quantization for efficient pre-training of transformer language models. In *Conference on Empirical Methods in Natural Language Processing*, pages 13473–13487, 2024.

[33] Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang. Zipml: Training linear models with end-to-end low precision, and a little bit of deep learning. In *International Conference on Machine Learning*, pages 4035–4043, 2017.

[34] Jianwei Li, Tianchi Zhang, Ian En-Hsu Yen, and Dongkuan Xu. Fp8-bert: Post-training quantization for transformer. *arXiv preprint arXiv:2312.05725*, 2023.

[35] Mitchell Wortsman, Tim Dettmers, Luke Zettlemoyer, Ari Morcos, Ali Farhadi, and Ludwig Schmidt. Stable and low-precision training for large-scale vision-language models. In *Advances in Neural Information Processing Systems*, 2023.

[36] Haocheng Xi, Han Cai, Ligeng Zhu, Yao Lu, Kurt Keutzer, Jianfei Chen, and Song Han. Coat: Compressing optimizer states and activation for memory-efficient fp8 training. *arXiv preprint arXiv:2410.19313*, 2024.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[38] Darrell Williamson. Dynamically scaled fixed point arithmetic. In *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing Conference Proceedings*, pages 315–318, 1991.

[39] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv: 1503.02531*, 2015.

[40] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18:187:1–187:30, 2017.

[41] Joonhyung Lee, Jeongin Bae, Byeongwook Kim, Se Jung Kwon, and Dongsoo Lee. To fp8 and back again: Quantifying the effects of reducing precision on llm training stability. *arXiv preprint arXiv:2405.18710*, 2024.

[42] Jianfei Chen, Lianmin Zheng, Zhewei Yao, Dequan Wang, Ion Stoica, Michael W. Mahoney, and Joseph Gonzalez. Actnn: Reducing training memory footprint via 2-bit activation compressed training. In *International Conference on Machine Learning*, pages 1803–1813, 2021.

[43] James Joseph Sylvester. Lx. thoughts on inverse orthogonal matrices, simultaneous signsuccessions, and tessellated pavements in two or more colours, with applications to newton's rule, ornamental tile-work, and the theory of numbers. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 34(232):461–475, 1867.

[44] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: data-free quantization aware training for large language models. In *Annual Meeting of the Association for Computational Linguistics*, pages 467–484, 2024.

[45] Ionut-Vlad Modoranu, Mher Safaryan, Grigory Malinovsky, Eldar Kurtic, Thomas Robert, Peter Richtárik, and Dan Alistarh. Microadam: Accurate adaptive optimization with low space overhead and provable convergence. In *Advances in Neural Information Processing Systems*, 2024.

[46] Shenggui Li, Hongxin Liu, Zhengda Bian, Jiarui Fang, Haichen Huang, Yuliang Liu, Boxiang Wang, and Yang You. Colossal-ai: A unified deep learning system for large-scale parallel training. In *Proceedings of the 52nd International Conference on Parallel Processing*, pages 766–775, 2023.

[47] Feijie Wu, Shiqi He, Song Guo, Zhihao Qu, Haozhao Wang, Weihua Zhuang, and Jie Zhang. Sign bit is enough: a learning synchronization framework for multi-hop all-reduce with ultimate compression. In *ACM/IEEE Design Automation Conference*, pages 193–198, 2022.

[48] Jiayi Yang, Lei Deng, Yukuan Yang, Yuan Xie, and Guoqi Li. Training and inference for integer-based semantic segmentation network. *Neurocomputing*, 454:101–112, 2021.

[49] Jacob Nielsen, Peter Schneider-Kamp, and Lukas Galke. Continual quantization-aware pre-training: When to transition from 16-bit to 1.58-bit pre-training for bitnet language models? *arXiv preprint arXiv:2502.11895*, 2025.

[50] Yue Yu, Jiaxiang Wu, and Longbo Huang. Double quantization for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pages 4440–4451, 2019.

[51] Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*, 2024.

[52] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Training deep neural networks with low precision multiplications. *arXiv preprint arXiv:1412.7024*, 2014.

[53] Kang Zhao, Sida Huang, Pan Pan, Yinghan Li, Yingya Zhang, Zhenyu Gu, and Yinghui Xu. Distribution adaptive INT8 quantization for training cnns. In *AAAI Conference on Artificial Intelligence*, pages 3483–3491, 2021.

[54] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4603–4611, 2018.

[55] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

[56] Charbel Sakr, Naigang Wang, Chia-Yu Chen, Jungwook Choi, Ankur Agrawal, Naresh R. Shanbhag, and Kailash Gopalakrishnan. Accumulation bit-width scaling for ultra-low precision training of deep networks. In *International Conference on Learning Representations*, 2019.

[57] Ilia Markov, Adrian Vladu, Qi Guo, and Dan Alistarh. Quantized distributed training of large models with convergence guarantees. In *International Conference on Machine Learning*, pages 24020–24044, 2023.

[58] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[59] Georgii Sergeevich Novikov, Daniel Bershatsky, Julia Gusak, Alex Shonenkov, Denis Valerievich Dimitrov, and Ivan V. Oseledets. Few-bit backward: Quantized gradients of activation functions for memory footprint reduction. In *International Conference on Machine Learning*, pages 26363–26381, 2023.

[60] Haocheng Xi, Yuxiang Chen, Kang Zhao, Kai Jun Teh, Jianfei Chen, and Jun Zhu. Jetfire: Efficient and accurate transformer pretraining with INT8 data flow and per-block quantization. In *International Conference on Machine Learning*, 2024.

[61] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, pages 3123–3131, 2015.

[62] Zhikai Li, Xiaoxuan Liu, Banghua Zhu, Zhen Dong, Qingyi Gu, and Kurt Keutzer. Qft: Quantized full-parameter tuning of llms with affordable resources. *arXiv preprint arXiv:2310.07147*, 2023.

[63] Hanyang Peng, Shuang Qin, Yue Yu, Jin Wang, Hui Wang, and Ge Li. Birder: Communication-efficient 1-bit adaptive optimizer for practical distributed DNN training. In *Advances in Neural Information Processing Systems*, 2023.

[64] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[65] Jingyang Li, Kuangyu Ding, Kim-Chuan Toh, and Pan Zhou. Memory-efficient 4-bit preconditioned stochastic optimization. *arXiv preprint arXiv:2412.10663*, 2024.

[66] Aditya Rajagopal, Diederik Adriaan Vink, Stylianos I. Venieris, and Christos-Savvas Bouganis. Multi-precision policy enforced training (muppet) : A precision-switching strategy for quantised fixed-point training of cnns. In *International Conference on Machine Learning*, pages 7943–7952, 2020.

[67] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U. Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pages 3252–3261, 2019.

[68] Yonggan Fu, Haoran You, Yang Zhao, Yue Wang, Chaojian Li, Kailash Gopalakrishnan, Zhangyang Wang, and Yingyan Lin. Fractrain: Fractionally squeezing bit savings both temporally and spatially for efficient DNN training. In *Advances in Neural Information Processing Systems*, 2020.

[69] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/jax-ml/jax`.

[70] Xishan Zhang, Shaoli Liu, Rui Zhang, Chang Liu, Di Huang, Shiyi Zhou, Jiaming Guo, Qi Guo, Zidong Du, Tian Zhi, and Yunji Chen. Fixed-point back-propagation training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2327–2335, 2020.

[71] Erwei Wang, James J. Davis, Daniele Moro, Piotr Zielinski, Jia Jie Lim, Claudionor Coelho, Satrajit Chatterjee, Peter Y. K. Cheung, and George A. Constantinides. Enabling binary neural network training on the edge. *ACM Transactions on Embedded Computing Systems*, 22(6):105:1–105:19, 2023.

[72] Eunhyeok Park, Sungjoo Yoo, and Peter Vajda. Value-aware quantization for training and inference of neural networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *European Conference on Computer Vision*, pages 608–624, 2018.

[73] Urs Köster, Tristan Webb, Xin Wang, Marcel Nassar, Arjun K. Bansal, William Constable, Oguz Elibol, Stewart Hall, Luke Hornof, Amir Khosrowshahi, Carey Kloss, Ruby J. Pai, and Naveen Rao. Flexpoint: An adaptive numerical format for efficient training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 1742–1752, 2017.

[74] Charbel Sakr and Naresh R. Shanbhag. Per-tensor fixed-point quantization of the back-propagation algorithm. In *International Conference on Learning Representations*, 2019.

[75] Fartash Faghri, Iman Tabrizian, Ilia Markov, Dan Alistarh, Daniel M. Roy, and Ali Ramezani-Kebrya. Adaptive gradient quantization for data-parallel SGD. In *Advances in Neural Information Processing Systems*, 2020.

[76] Pengle Zhang, Jia Wei, Jintao Zhang, Jun Zhu, and Jianfei Chen. Accurate int8 training through dynamic block-level fallback. *arXiv preprint arXiv:2503.08040*, 2025.

[77] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[78] Minghao Li, Ran Ben Basat, Shay Vargaftik, ChonLam Lao, Kevin Xu, Michael Mitzenmacher, and Minlan Yu. THC: accelerating distributed deep learning using tensor homomorphic compression. In *USENIX Symposium on Networked Systems Design and Implementation*, 2024.

[79] Junbiao Pang and Tianyang Cai. Stabilizing quantization-aware training by implicit-regularization on hessian matrix. *arXiv preprint arXiv:2503.11159*, 2025.

[80] John L. Gustafson and Isaac T. Yonemoto. Beating floating point at its own game: Posit arithmetic. *Supercomputing Frontiers and Innovations*, 4(2):71–86, 2017.

[81] Brian Chmiel, Liad Ben-Uri, Moran Shkolnik, Elad Hoffer, Ron Banner, and Daniel Soudry. Neural gradients are near-lognormal: improved quantized and

sparse training. In *International Conference on Learning Representations*, 2021.

[82] Miaoxi Zhu, Qihuang Zhong, Li Shen, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. Zero-shot sharpness-aware quantization for pre-trained language models. In *Conference on Empirical Methods in Natural Language Processing*, pages 11305–11327, 2023.

[83] Zhenyu Zhang, Ajay Jaiswal, Lu Yin, Shiwei Liu, Jiawei Zhao, Yuandong Tian, and Zhangyang Wang. Q-galore: Quantized galore with int4 projection and layer-adaptive low-rank gradients. *arXiv preprint arXiv:2407.08296*, 2024.

[84] Florent De Dinechin, Luc Forget, Jean-Michel Muller, and Yohann Uguen. Posits: the good, the bad and the ugly. In *Proceedings of the Conference for Next Generation Arithmetic 2019*, pages 1–10, 2019.

[85] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[86] Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. Bitdistiller: Unleashing the potential of sub-4-bit llms via self-distillation. In *Annual Meeting of the Association for Computational Linguistics*, pages 102–116, 2024.

[87] Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. Outlier suppression: Pushing the limit of low-bit transformer language models. *Advances in Neural Information Processing Systems*, pages 17402–17414, 2022.

[88] Lu Xia, Martijn Anthonissen, Michiel Hochstenbach, and Barry Koren. A simple and efficient stochastic rounding method for training neural networks in low precision. *arXiv preprint arXiv:2103.13445*, 2021.

[89] David R Lutz, Anisha Saini, Mairin Kroes, Thomas Elmer, and Harsha Valsaraju. Fused fp8 4-way dot product with scaling and fp32 accumulation. In *2024 IEEE 31st Symposium on Computer Arithmetic (ARITH)*, pages 40–47. IEEE, 2024.

[90] Li Ding, Wen Fei, Yuyang Huang, Shuangrui Ding, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. AMPA: adaptive mixed precision allocation for low-bit integer training. In *International Conference on Machine Learning*, 2024.

[91] Maxim Fishman, Brian Chmiel, Ron Banner, and Daniel Soudry. Scaling fp8 training to trillion-token llms. *arXiv preprint arXiv:2409.12517*, 2024.

[92] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Lifeng Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 1, 2024.

[93] Po-Chen Lin, Mu-Kai Sun, Chuking Kung, and Tzi-Dar Chiueh. Floatsd: A new weight representation and associated update method for efficient convolutional neural network training. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2):267–279, 2019.

[94] Guandao Yang, Tianyi Zhang, Polina Kirichenko, Junwen Bai, Andrew Gordon Wilson, and Christopher De Sa. SWALP : Stochastic weight averaging in low precision training. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *International Conference on Machine Learning*, pages 7015–7024, 2019.

[95] Sergio P Perez, Yan Zhang, James Briggs, Charlie Blake, Josh Levy-Kramer, Paul Balanca, Carlo Luschi, Stephen Barlow, and Andrew William Fitzgibbon. Training and inference of large language models using 8-bit floating point. *arXiv preprint arXiv:2309.17224*, 2023.

[96] Marios Fournarakis and Markus Nagel. In-hindsight quantization range estimation for quantized training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3063–3070, 2021.

[97] Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*, 2021.

[98] Charlie Blake, Douglas Orr, and Carlo Luschi. Unit scaling: Out-of-the-box low-precision training. In *International Conference on Machine Learning*, pages 2548–2576. PMLR, 2023.

[99] Xiao Sun, Jungwook Choi, Chia-Yu Chen, Naigang Wang, Swagath Venkataramani, Vijayalakshmi Srinivasan, Xiaodong Cui, Wei Zhang, and Kailash Gopalakrishnan. Hybrid 8-bit floating point (HFP8) training and inference for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 4901–4910, 2019.