SineLoRAΔ: Sine-Activated Delta Compression

Cameron Gordon[*1], **Yiping Ji**[*1], **Hemanth Saratchandran**[*1], **Paul Albert** [2], **Simon Lucey** [1]

[1]Australian Institute for Machine Learning (AIML), University of Adelaide, Australia
[2]Amazon Machine Learning, Australia
{cameron.gordon, yiping.ji, hemanth.saratchandran, simon.lucey}@adelaide.edu.au

Abstract

Resource-constrained weight deployment is a task of immense practical importance. Recently, there has been interest in the specific task of *Delta Compression*, where parties each hold a common base model and only communicate compressed weight updates. However, popular parameter efficient updates such as Low Rank Adaptation (LoRA) face inherent representation limitations - which are especially pronounced when combined with aggressive quantization. To overcome this, we build on recent work that improves LoRA representation capacity by using fixed-frequency sinusoidal functions to increase stable rank without adding additional parameters. We extend this to the quantized setting and present the first theoretical analysis showing how stable rank evolves under quantization. From this, we introduce **SineLoRAΔ**, a principled and effective method for delta compression that improves the expressivity of quantized low-rank adapters by applying a sinusoidal activation. We validate SineLoRAΔ across a diverse variety of domains - including language modeling, vision-language tasks, and text-to-image generation - achieving up to 66% memory reduction with similar performance. We additionally provide a novel application of the canonical Bjøntegaard Delta metric to consistently compare adapter compression changes across the rate-distortion curve.

Introduction

Parameter-Efficient Fine-Tuning (PEFT) has emerged as a core component of modern machine learning pipelines (Houlsby et al. 2019; Han et al. 2024). Most PEFT methods adapt a frozen pre-trained backbone by learning a small set of task-specific parameters, often implemented as additive weight updates. Among these approaches, Low-Rank Adapters have become especially prominent, with a rapidly expanding literature (Hu et al. 2022; Mao et al. 2025). Recent work has sought to further reduce the number of trainable parameters by exploring alternative low-rank decompositions (Karimi Mahabadi, Henderson, and Ruder 2021; Edalati et al. 2025; Liu et al. 2024b; He et al. 2023; Ding et al. 2023; Albert et al. 2025; Kopiczko, Blankevoort, and Asano 2024; Koohpayegani et al. 2024).

Recently, a new fine-tuning paradigm has emerged that enhances the expressive power of low-rank adapters by applying rank-enhancing functions component-wise. Introduced in (Ji et al. 2025), this approach demonstrates that applying a non-linear transformation, specifically, a fixed-frequency sinusoidal function, to a low-rank adapter can significantly increase its rank. This expressivity gain comes at no additional parameter cost, preserving the memory efficiency of LoRA while yielding higher-rank representations.

In this paper, we investigate the interaction between rank-enhancing sinusoidal non-linearities and quantization, a technique that maps full-precision parameters to a smaller set of discrete values, ideally with minimal impact on model performance (Han, Mao, and Dally 2015; Gholami et al. 2022; Li et al. 2024). Quantization is a key enabler for deploying large models on resource-constrained hardware, offering improvements in memory efficiency, computational throughput, and energy consumption (Gholami et al. 2022; Dettmers et al. 2023; Xu et al. 2024; Kaushal et al. 2025).

To study this interaction, we develop a theoretical framework that characterizes how the rank of an adapter changes under quantization, showing that it is tightly controlled by the rank of the original, unquantized adapter. This leads to our key insight: when the adapter has low-rank, as is the case with LoRA, quantization preserves this structure. However, by applying a component-wise sinusoidal non-linearity after quantization, we can enrich the representational capacity of the adapter, effectively compensating for the rank limitation and enabling more expressive quantized models.

This insight is particularly relevant in the context of adapter quantization, which has emerged as one of two dominant approaches in quantized fine-tuning. The first, exemplified by QLoRA (Dettmers et al. 2023; Badri and Shaji 2024), applies quantization to the base model while maintaining high-precision adapters and activations. This approach is primarily motivated by reducing memory overhead during fine-tuning, making it feasible to adapt large language models on a single GPU (Dettmers et al. 2023). The second approach focuses on quantizing the adapters themselves (Yao, Hu, and Klimovic 2025; Liu et al. 2024a; Isik et al. 2023; Ping et al. 2024; Jie, Wang, and Deng 2023), enabling highly compact and transferable fine-tuned models. Our work follows this latter direction, showing that rank-enhancing sinusoidal functions can be easily integrated as a plug-in component into quantized adapters, significantly improving their expressivity while retaining the memory ef-

[*]These authors contributed equally.

ficiency that makes adapter quantization attractive.

To our knowledge, such rank-enhancing functions have not yet been explored in the quantization literature, and we view this work as a first step towards bridging that gap. Our main contributions are as follows:

- We provide the first theoretical analysis showing how quantization affects the stable rank of a fine-tuning adapter, and show that this rank is tightly governed by the rank of its unquantized counterpart.
- Based on our theoretical results we demonstrate that the effects of quantization on rank can be mitigated by applying rank enhancing functions in the form of sinusoids with fixed frequencies.
- In order to consistently evaluate compression effectiveness, we introduce a novel application of Bjøntegaard Delta (canonically used to compare image codecs) to compare the effect of PEFT compression approaches.

We evaluate our approach through extensive experiments on vision and language tasks, including Large Language Model adaptation, Vision-Language Model Adaptation, and Text-to-Image Generation. For evaluation on Commonsense Reasoning, we show that memory reductions of up to 66% is achievable relative to full-precision LoRA models by combining quantization and a rank-enhancing sine function.

Related Work

Parameter Efficient Adapters Parameter efficient adaptation is a common fine-tuning strategy, in which a pre-trained base model is frozen, and a minimal number of adapter weights are trained on new data (Houlsby et al. 2019). Low-Rank Adapters are a common variant, in which the adapter comprises two low-rank matrices (Hu et al. 2022). VeRA (Kopiczko, Blankevoort, and Asano 2024), RandLoRA (Albert et al. 2025), and NOLA (Koohpayegani et al. 2024) use combinations of random projections to reduce the number of parameters contained within the adapters. QA-LoRA (Xu et al. 2024) produces adapters that can be merged with the quantized base model, enabling low-precision inference.

Rank-Enhancing Functions The most relevant rank-enhancing adapter works related to our approach are Ji et al. (2025) and Li et al. (2024) who investigate the use of sine-nonlinearities in low-rank adaptation (Ji et al. 2025; Li, Song, and Hou 2024). We extend this approach by considering the effect of quantization on adapter performance.

Delta Compression and Quantization Although adapters represent a trivial proportion of the total number of parameters in a network (typically less than 1%), a recent branch of research has focused on the specific compression of these updates. Termed *Delta Compression*, this branch recognizes the practical importance of reducing the memory throughput of fine-tuned updates, which may be distributed at scale to many parties with a common base model (Isik et al. 2023; Yao, Hu, and Klimovic 2025; Brüel-Gabrielsson et al. 2025). Within this framework it is typical to quantize the adapters, by mapping values to a limited set of floating points. This is often combined with lossless entropy compression such as zip. The quantization works most related to our approach are GPT-Zip (Isik et al. 2023), Delta-Zip (Yao, Hu, and Klimovic 2025), Bit Delta (Liu et al. 2024a), and Bi-LoRA (Jie, Wang, and Deng 2023). (Ping et al. 2024) use a mixed-precision strategy, devoting higher precision to larger singular values. (Jiang et al. 2024) uses a group-wise dropout and separate quantization. (Ryu, Seo, and Yoo 2023) focus on low-rank residuals. (Liu et al. 2024a) uses binary adapters for Delta Compression. Our work differs from these models through our specific focus on the rank-increasing properties of a sine adaptation within a quantized framework.

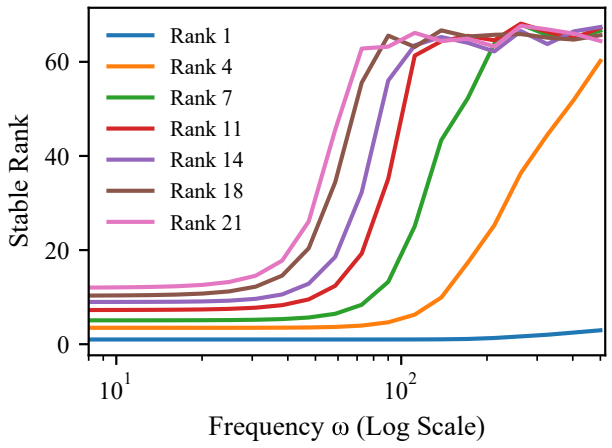

Figure 1: Effects of changing the ω term in a sine-activated low-rank matrix. Stable Rank saturates at sufficient ω.

Theoretical Framework

Preliminaries

Sine-Activated Low-Rank Adapters Recent works by (Ji et al. 2025) and (Li, Song, and Hou 2024) have explored the use of non-linear sine activations in adapter modules. Unlike common activations such as ReLU, sine functions can increase the rank of a matrix without adding additional parameters, offering a simple yet effective means of enhancing low-rank adapters. Specifically, (Ji et al. 2025) introduced a sine-activated low-rank adapter of the form:

$$\frac{\sin(\omega AB)}{\gamma} \tag{1}$$

where ω is a frequency parameter, γ is a scaling factor, and $A \in \mathbb{R}^{m\times k}$, $B \in \mathbb{R}^{k\times n}$ are low-rank matrices with bottleneck dimension k.

Stable Rank The key insight of (Ji et al. 2025) is that applying a sine function to the low-rank product AB, with large enough frequency ω, would increase the stable rank of the matrix AB which can then be used to increase the rank yielding a high rank adapter. The stable rank of a matrix A

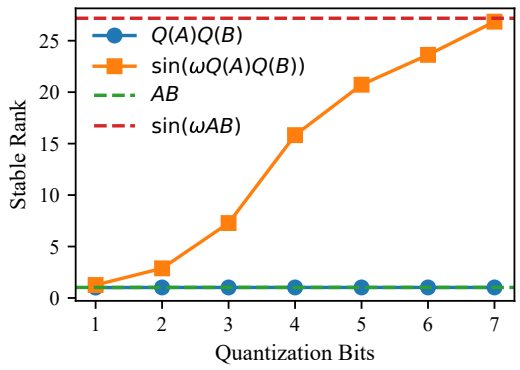

Figure 2: A sine-activated low-rank matrix $\sin(\omega AB)$ increases the stable rank relative to a low-rank matrix AB. By varying quantization level $\sin(\omega(Q(A)Q(B))$ we can interpolate the effect on stable rank between these two values.

is defined by:

$$\mathbf{SR}(A) := \frac{||A||_F^2}{(\sigma(A)_{\max})^2} \tag{2}$$

where $||A||_F^2$ denotes the Frobenius norm and $\sigma_{\max}(A)$ the maximum singular value of A. Stable rank provides a softer measure of a matrix's effective dimensionality (Martinsson and Tropp 2020). Unlike the classical rank, which counts the number of nonzero singular values, the stable rank reflects how evenly the spectral energy is distributed. For instance, two matrices with identical rank can have vastly different stable ranks depending on the decay of their singular values. This nuance is critical when aiming to enhance low-rank adapters: even without increasing the classical rank, we can improve the adapter's expressivity by boosting its stable rank. This is precisely the property exploited by sine-activated adapters in (Ji et al. 2025). Figure 1 shows how the stable rank is affected by changing the frequency term in a sine-activated matrix for different low-rank constraints.

Quantization A quantization function $Q(\cdot)$ maps values from a less restricted set to a more restricted set $\mathcal{A} \to \mathcal{B}$. Practically, this may involve explicit conversion of data-types (e.g. 16-bit precision to 4-bit precision), or maintaining the same data-type, but restricting the set of allowed values (e.g. mapping from 2^{16} discrete values to 2^4) (Gholami et al. 2022; Gray and Neuhoff 1998). This type of quantization is common for memory compression and often coupled with an integer look-up table or an entropy coder (Han, Mao, and Dally 2015; Jacob et al. 2018). It is conventional to define *quantization error* as the residual resulting from a quantization map, which can be treated as a random variable (Gersho and Gray 1991; Gray and Neuhoff 1998):

$$\epsilon = Q(A) - A \tag{3}$$

For our experiments, we use a k-means quantization scheme due to its theoretical optimality and tractable implementation (Gersho and Gray 1991; Han, Mao, and Dally 2015). This is implemented using the *k-means1d* package (Steinberg 2019), which provides an efficient wrapper for a fast k-means solver that runs in $O(kn + n \log n)$ for n 1D data and k clusters based on (Wu 1991; Grønlund et al. 2018). Further quantization experimental details are included in the Supplementary Materials.

Main Theorem

In this section, we present our main theoretical result, which establishes that the stable rank of a quantized matrix is governed by the stable rank of its unquantized counterpart. We will use the notation $\sigma_{\max}$ to denote the maximum singular value of a matrix, $\sigma_{\min}$ to denote the minimum singular value and $||\cdot||_F$ to denote the Frobenius norm.

Theorem 1. *Let A be a fixed matrix and let Q denote a quantization operator so that $Q(A) = A - \epsilon$. Assume that $\sigma_{\min}(A) \leq 1$ and $\sigma_{\max}(A) >> \sigma_{\max}(\epsilon) >> 1$, so that $\frac{\sigma_{\max}(A)}{2} \leq \sigma_{\max}(A) - \sigma_{\max}(\epsilon)$. Then:*

$$\frac{1}{2}\left(\sqrt{\mathbf{SR}(A)} - \frac{||\epsilon||_F}{\sigma_{\max}(A)}\right) \leq \sqrt{\mathbf{SR}(Q(A))} \leq 2\left(\sqrt{\mathbf{SR}(A)} + \frac{||\epsilon||_F}{\sigma_{\max}(A)}\right) \tag{4}$$

where ϵ is defined by (3).

Proof. We recall from (3) that we can write:

$$Q(A) = A - \epsilon \tag{5}$$

were ϵ is viewed as a random noise matrix. We then use the triangle inequality to obtain:

$$||A||_F - ||\epsilon||_F \leq ||Q(A)||_F \leq ||A||_F + ||\epsilon||_F. \tag{6}$$

Using inequalities for the maximum singular value of a matrix we have:

$$\sigma_{\max}(A) - \sigma_{\max}(\epsilon) \leq \sigma_{\max}(Q(A)) \tag{7}$$
$$\leq \sigma_{\max}(A) + \sigma_{\max}(\epsilon). \tag{8}$$

To prove the upper bound observe that:

$$\sqrt{\mathbf{SR}(Q(A))} = \frac{||Q(A)||_F}{\sigma_{max}(Q(A))} \tag{9}$$
$$\leq \frac{||A||_F + ||\epsilon||_F}{\sigma_{max}(Q(A))} \text{ by (6)} \tag{10}$$
$$\leq \frac{||A||_F + ||\epsilon||_F}{\sigma_{max}(A) - \sigma_{\min}(A)} \text{ by (8)} \tag{11}$$
$$\leq 2\left(\frac{||A||_F + ||\epsilon||_F}{\sigma_{max}(A)}\right) \tag{12}$$

where to get the last inequality we use the assumption in the statement of the theorem. The upper bound then follows from the definition of the stable rank.

To prove the lower bound we proceed in a similar way:

$$\sqrt{\mathbf{SR}(Q(A))} = \frac{||Q(A)||_F}{\sigma_{max}(Q(A))} \tag{13}$$

$$\geq \frac{||A||_F - ||\epsilon||_F}{\sigma_{max}(Q(A))} \text{ by (6)} \tag{14}$$

$$\geq \frac{||A||_F - ||\epsilon||_F}{\sigma_{max}(Q(A)) + \sigma_{\max}(\epsilon)} \text{ by (8)} \tag{15}$$

$$\geq \frac{1}{2}\left(\frac{||A||_F - ||\epsilon||_F}{\sigma_{max}(Q(A))}\right) \tag{16}$$

where the last inequality comes from the assumption that $\sigma_{\max}(A) \geq \sigma_{\max}(\epsilon)$. The lower bound then follows from the definition of stable rank. □

Theorem 1 presents the key insight of this work: the stable rank of a quantized adapter remains low if the original (unquantized) adapter has low stable rank, as the quantized stable rank is controlled by the unquantized one. This observation motivates applying a sinusoidal function, with a large frequency ω, after quantization. By leveraging results from (Ji et al. 2025), we note that a sine function with large frequency can increase the stable rank of the quantized adapter, effectively boosting its expressivity without sacrificing quantization efficiency. This produces a high-rank adapter while retaining the compression benefits of quantization. In particular this makes applying a sinusoidal function to a post quantization framework an effective way to yield better performance while still retaining compression benefits. Figure 2 provides an empirical illustration of our main insight. Starting with two low-rank matrices A and B, whose product AB is also low-rank, we apply quantization Q to A and B at varying bits. The figure plots the stable ranks of AB, the quantized product $Q(A)Q(B)$, the sine-activated product $\sin(\omega AB)$, and $\sin(\omega Q(A)Q(B))$. As shown, the stable rank of $\sin(\omega Q(A)Q(B))$ increases with higher quantization bits, demonstrating how sinusoidal activation can effectively restore rank after quantization.

Bjøntegaard Delta Analysis

Bjøntegaard Delta (BD) Analysis is a commonly applied evaluation technique for comparing video and image compression codecs (Bjøntegaard 2001; Herglotz et al. 2022, 2024), and has occasionally been applied for other modalities such as Point Cloud (Wang et al. 2021a,b; Herglotz et al. 2024; Barman, Martini, and Reznik 2022) or Neural Radiance Field compression (Ji et al. 2025). The metric involves evaluating two comparison codecs at two rate and performance positions. These are interpolated, with the measure evaluated as the integral between these positions. Figure 3 shows visually how the Bjøntegaard Delta can be calculated. Standard metrics include BD-Rate (compression gains at equivalent performance) and BD-PSNR (performance gains at equivalent bitrate). These can be extended without difficulty to performance evaluation metrics used for language models, such as average task accuracy (BD-Accuracy).

Mathematical Description Given RD points (R_i, D_i), we interpolate the RD curves using a smooth function $f(R)$. The BD-rate is computed as:

$$\Delta R = A \int_{D_{\min}}^{D_{\max}} \left(f_2^{-1}(D) - f_1^{-1}(D)\right) dD \tag{17}$$

Similarly, BD-Accuracy is inversely defined:

$$\Delta D = B \int_{R_{\min}}^{R_{\max}} \left(f_2(R) - f_1(R)\right) dR \tag{18}$$

Where $A = \frac{1}{D_{\max} - D_{\min}}$ and $B = \frac{1}{R_{\max} - R_{\min}}$.

Typically, a cubic polynomial is used for $f(\cdot)$, however recent works have argued for the use of Akima interpolation due to its increased stability (Herglotz et al. 2022, 2024). For our evaluation we use the *Bjontegaard* Python library accessible at https://github.com/FAU-LMS/bjontegaard.

Applicability to Delta Compression Model compression is an area of relatively recent interest in contrast to more established modalities such as images and video (Zhu et al. 2024; Hadish et al. 2024). As a result evaluating compression changes has yet to be standardised. It is therefore common to present model gains visually to show Pareto improvements; or to compare performance changes at an (approximately) equivalent parameter or performance position (e.g. '. . . performance with 25% fewer parameters'). This leads to difficulties where parameters are not directly comparable or there is variation in algorithm performance at different memory levels. In contrast, the advantage of BD analysis is that it accounts for small inconsistencies in parameters, and accounts for the natural rate-performance trade-off that occurs during compression. We suggest that BD analysis can be used to evaluate Delta Compression performance.

Results

Large Language Model Adaptation

Configurations We fine-tune LLaMA 3-8B on a commonsense reasoning tasks, training the 15k dataset for 1 epoch. Following training we apply Post-Training Quantization at different bits-per-parameter, with each target tensor quantized independently. We then evaluate on each of the test sets directly, without further fine-tuning. We evaluate on a standard suite of benchmarks including BoolQ (Clark et al. 2019), PIQA (Bisk et al. 2020), SIQA (Sap et al. 2019), HellaSwag (HS) (Zellers et al. 2019), WinoGrande (WG) (Sakaguchi et al. 2021), ARC-c, ARC-e (Clark et al. 2018) and OBQA (Mihaylov et al. 2018). We use a frozen LLAMA-3-8B base model from Hugging Face. Each base experiment is run on one H100 GPU using a batch size of 128, and re-used for quantizing to different levels of precision. Low-rank adaptors are applied to the weight matrices $\mathbf{W}_{\text{q}}, \mathbf{W}_{\text{k}}, \mathbf{W}_{\text{v}}, \mathbf{W}_{\text{up}}$, and $\mathbf{W}_{\text{down}}$. We use the ω values in (Ji et al. 2025), who apply larger ω for low-rank models. Following (Ji et al. 2025) we set $\gamma = \sqrt{n}$, where n is the row dimension of the weight matrix.

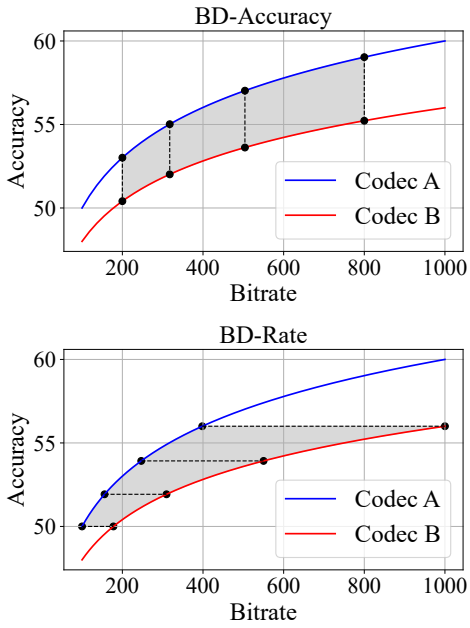

Figure 3: Bjøntegaard Delta is calculated by taking (bitrate, accuracy) pairs for two codecs, and evaluating the (horizontal or vertical) integral between interpolated performance.

Analysis Results are shown in Table 1. We can note that the SineLoRAΔ model achieves significant memory compression and consistently outperforms LoRA. The memory reduction is such that the Rank 8 SineLoRAΔ at 5-bits outperforms the full-precision LoRA, with only 33.5% of the memory (9.1MB to 27.1MB). Table 2 further validates this by examining the average performance improvements through Bjøntegaard Delta Analysis at each quantization level. At 2-bit quantization the SineLoRAΔ model shows an average 41.6% memory improvement over LoRA, and an average accuracy improvement of 1.29%. Full experimental results are recorded in Supplementary Tables 2 and 3, which ablates the effect of using a quantized base mode. Results are broadly comparable under this setting. We additionally compare results to DoRA (Liu et al. 2024b), and find that while both LoRA and SineLoRAΔ are robust to low (2-bit) quantization, the performance of DoRA degrades significantly until adapters are quantized to higher than 5-bit precision.

Vision-Language Model Adaptation

Data We fine-tune CLIP (Radford et al. 2021) on 11 standard image classification datasets, obtained by following (Zhang et al. 2024). These include: Cars (Krause et al. 2013), DTD (Cimpoi et al. 2014), EuroSAT (Helber et al. 2018), Food101 (Bossard, Guillaumin, and Van Gool 2014), Caltech101 (Fei-Fei, Fergus, and Perona 2006), Sun397 (Xiao et al. 2016), FGVCAircraft (Maji et al. 2013), Flowers102 (Nilsback and Zisserman 2008), ImageNet (Russakovsky et al. 2015), Oxford Pets (Parkhi et al. 2012), and UCF101 (Soomro, Zamir, and Shah 2012). We compare the performance of LoRA (Hu et al. 2022), SineLoRAΔ (Ji et al. 2025) for few-shot adaptation using a ViT-B/32 backbone following Post-Training Quantization.

	Rank				
Method	1	2	4	8	16
LoRA (2-bit)	69.7	71.0	74.7	75.2	77.3
SineLoRAΔ (2-bit)	**70.0**	**73.7**	**75.1**	**76.4**	**77.9**
Memory (MB)	0.6	1.1	2.2	4.3	8.6
LoRA (3-bit)	70.0	73.1	75.5	76.5	78.4
SineLoRAΔ (3-bit)	**70.5**	**74.4**	**75.9**	**77.7**	**78.6**
Memory (MB)	0.8	1.5	3.0	6.0	11.9
LoRA (5-bit)	69.4	73.1	75.6	76.7	78.6
SineLoRAΔ (5-bit)	**69.8**	**74.4**	**76.1**	**78.1**	**78.8**
Memory (MB)	1.2	2.3	4.5	9.1	18.1
LoRA (Full)	**73.7**	74.8	76.5	78.0	**79.0**
SineLoRAΔ (Full)	72.8	**75.1**	**78.5**	**78.8**	78.9
Memory (MB)	3.4	6.8	13.5	27.1	54.0
Parameters (M)	1.8	3.5	7.1	14.2	28.3

Table 1: Commonsense Reasoning performance for LoRA and SineLoRAΔ under different quantization rates. Averaged across tasks. Full refers to the typical 16-bit precision.

Quantization Level	BD-Rate ↓	BD-Accuracy ↑
2	-41.60%	1.29%
3	-28.51%	0.88%
5	-28.04%	0.96%
16	-30.46%	0.69%

Table 2: Bjøntegaard Delta Analysis for Commonsense Reasoning experiments in Table 1, with the respective LoRA model as the baseline codec. Rate-distortion pairs are generated by keeping the quantization level fixed and varying the number of parameters through rank. SineLoRAΔ demonstrates improved performance at each quantization level.

Quantization Level	BD-Rate ↓	BD-Accuracy ↑
2	19.89%	-0.90%
3	-15.65%	0.41%
5	-42.81%	0.83%
16	-44.38%	0.87%

Table 3: Bjøntegaard Delta Analysis for the Vision-Language Model Adaptation results in Table 4, with the respective LoRA model as the baseline codec. Rate-distortion pairs are generated by keeping each quantization level fixed and varying the number of parameters through rank.

Configurations Experiments are run on a NVIDIA GeForce RTX 4090 GPU with 24GB VRAM. Batch size 64, base model ViT-B/32, learning rate 0.001, weight decay 0.1, 10 epochs, AdamW optimizer (Loshchilov and Hutter 2019). Fine-tuning is conducted on attention layers($\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$) only. We finetune on few-shot tasks using 1 and 16 examples, employing different rank lev-

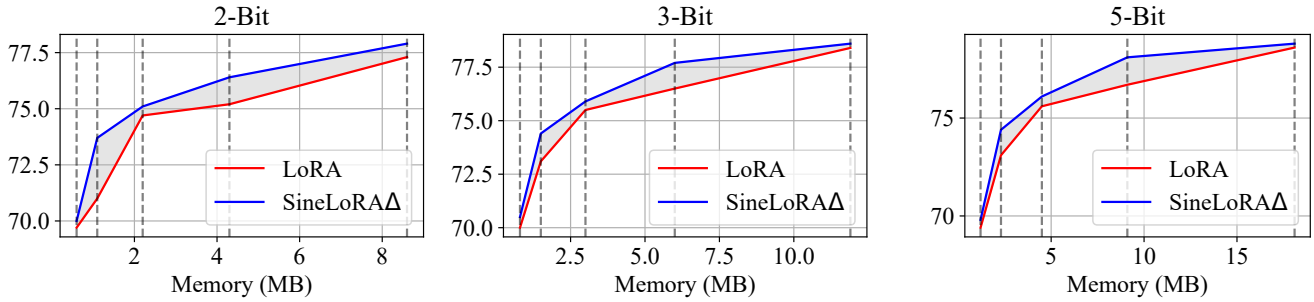

Figure 4: Commonsense Reasoning performance (average) for SineLoRAΔ and LoRA with a frozen non-quantized LLAMA-3-8B base model. SineLoRAΔ exceeds the benchmark LoRA performance across all evaluated rank and quantization levels.

els. We use $\omega = 200$ for all experiments, and $\gamma = \sqrt{n}$ where n is the weight row dimension.

Analysis Tables 3 and 4 shows our results on 1-shot classification averaged over 11 vision tasks. Consistent with the language model experiments, we observe that the SineLoRAΔ model outperforms the baseline LoRA at a similar rank and quantization. We observe that the SineLoRAΔ model only outperforms LoRA at 3-bits and higher, after which consistent compression improvements are found. This may be potentially explained by recalling Figure 2, in which lower stable rank improvements are observed for very low precision (1 and 2 bit) quantization. In the Supplementary Materials we include additional ablations and comparison with DoRA (Liu et al. 2024b).

Text-to-Image Generation

Training Details To investigate how SineLoRAΔ performs on a text-to-image generation task, we adopt a DreamBooth fine-tuning pipeline (Ruiz et al. 2023). DreamBooth is a method for adapting text-to-image diffusion models using just a few reference images of a target object. Our experiments are performed on Stable Diffusion 3 Medium (Esser et al. 2024), using the official Hugging Face implementation[1]. For data, we use the DreamBooth dataset comprising 30 objects with 5-6 images per instance. For each object, we train a separate adapter. Following training, we quantize adapters to 1, 2, 3, and 5 bits using k-means quantization. These are evaluated using standard generative text prompts with 2 seeds each. For both LoRA and SineLoRAΔ we train rank 4 adapters for 300 epochs using the AdamW optimizer using a learning rate of 4×10^{-4} (Loshchilov and Hutter 2019). For SineLoRAΔ we use a frequency $\omega = 200$ and $\gamma = 2\sqrt{n}$. All experiments are run on NVIDIA H100 GPUs, with each fine-tuning run taking around 7 minutes.

Analysis Figure 5 shows a qualitative evaluation of SineLoRAΔ and LoRA trained using Dreambooth. Results show increased object fidelity for the SineLoRAΔ models, which is maintained at lower quantization levels than LoRA. Quantitatively, we follow (Ruiz et al. 2023) and report the average cosine similarities between CLIP/DINO embeddings of generated images and subject images (CLIP-I and DINO), and of generated images and the text prompt (CLIP-T) (Radford et al. 2021; Caron et al. 2021). Table 5 shows results averaged over all 30 categories evaluated at epoch 300. We find consistent performance improvements at each quantization level for CLIP-I and DINO, which measure the similarity to the target object. Evaluating the BD-Rate between LoRA and SineLoRAΔ we find a memory improvement of -34.84% on CLIP-I and -29.48% for DINO. Comparable performance between the two models is found CLIP-T, with a small 5% improvement to the baseline LoRA. This is consistent with our qualitative results as CLIP-I and DINO measure how accurately the adapter has managed include the target object in the scene, while CLIP-T indicates how closely the overall scene matches the text prompt. We observe that 1-bit for both models has less fidelity to the fine-tuned target image, and appears dominated by the prompt. We attribute this to the increased dominance of the base model weights for generation. We provide additional qualitative results and analysis on individual category performance in the Supplementary Materials.

Discussion and Limitations

Quantization Aware Training

Experimentally we have applied a Post-Training Quantization pipeline, which compresses weights following training. This has practical computational advantages as it allows evaluation of full rate-distortion curves without retraining at individual bit-rates. It is worth noting that improvements in performance are often possible by using Quantization Aware Training, which apply quantization during the training procedure (Gholami et al. 2022; Rastegari et al. 2016). While a systematic exploration of this is independent of our research question we note this as a direction for future research.

Inference Precision

The quantization scheme we have employed maps tensors to a restricted set of float values (e.g. 2^4 values for 4-bit quantization), without recasting tensor data-types (Gholami et al. 2022). This is commonly employed in memory compression

[1]https://github.com/huggingface/diffusers/tree/main/examples/dreambooth

Model	Rank	1-Bit	2-Bit	3-Bit	4-Bit	5-Bit	8-Bit	Full	Params
LoRA	2	**67.3**	**70.0**	74.1	76.0	76.4	76.4	76.5	123K
SineLoRAΔ	2	63.5	68.1	**74.2**	**76.3**	**76.9**	**77.0**	**77.0**	123K
LoRA	5	**70.5**	**74.7**	77.0	77.5	77.8	77.8	77.9	307K
SineLoRAΔ	5	66.9	74.1	**77.5**	**78.6**	**78.7**	**78.9**	**78.9**	307K
LoRA	10	**71.6**	**77.2**	78.3	78.8	78.7	78.8	78.9	614K
SineLoRAΔ	10	68.7	76.3	**78.8**	**79.4**	**79.6**	**79.8**	**79.8**	614K
LoRA	16	**72.9**	**78.1**	79.2	79.4	79.5	79.4	79.5	983K
SineLoRAΔ	16	68.3	77.4	**79.5**	**80.0**	**80.3**	**80.2**	**80.3**	983K

Table 4: Vision-Language Model Adaptation (Averaged Over 11 Tasks) ↑

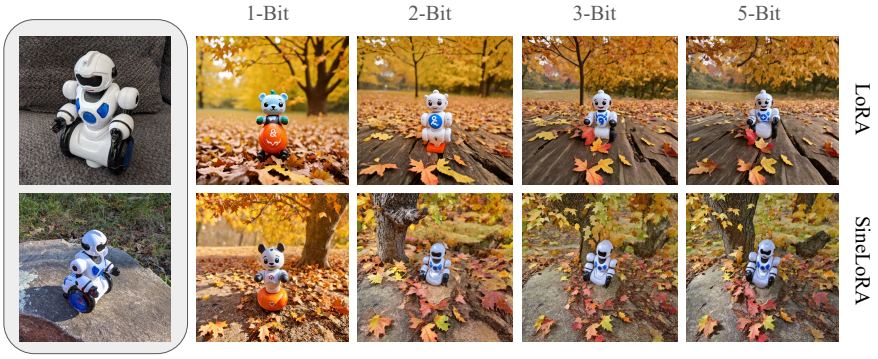

Figure 5: Dreambooth Stable Diffusion for the prompt **A toy with tree and autumn leaves in the background** for the category **robot toy**. SineLoRAΔ exhibits greater consistency with target images (left) than LoRA even at low levels of quantization.

Bits	**Model**	**CLIP-I ↑**	**CLIP-T ↑**	**DINO ↑**
1	LoRA	0.729	**0.219**	0.515
	SineLoRAΔ	**0.746**	**0.219**	**0.554**
2	LoRA	0.768	0.218	0.599
	SineLoRAΔ	**0.780**	**0.219**	**0.616**
3	LoRA	0.780	0.218	0.621
	SineLoRAΔ	**0.785**	**0.219**	**0.625**
5	LoRA	0.783	**0.219**	0.626
	SineLoRAΔ	**0.787**	**0.219**	**0.629**
Full	LoRA	0.784	**0.321**	0.626
	SineLoRAΔ	**0.790**	0.317	**0.632**

Table 5: Comparison of LoRA and SineLoRAΔ for Text-to-Image Generation. Best scores for each bit-width group and metric are highlighted in **bold**.

for efficient data transfer. As both inference and training are conducted in the original data-type, it can be easily applied without modified memory types. However, this does not exploit GPU-level optimizations available for alternative data-types (Gholami et al. 2022; Dettmers et al. 2023). Combining our approach with methods such as QA-LoRA which enable INT-4 inference may lead to additional efficiency improvements (Xu et al. 2024).

Conclusion

In this work we have presented SineLoRAΔ, a simple and effective enhancement for quantized low-rank adapters that significantly improves expressivity without introducing additional parameters. Our key theoretical insight is that stable rank under quantization is bounded by that of the original adapter - highlighting an inherent limitation of LoRA in compressed regimes. By applying fixed-frequency sinusoidal functions post-quantization, we demonstrate both analytically and empirically that these rank limitations can be mitigated. Across a diverse set of tasks - including large language model tuning, few-shot vision classification, and text-to-image generation - SineLoRAΔ delivers consistent improvements in accuracy at significantly reduced memory cost. We further propose the use of the Bjøntegaard Delta metric to evaluate compression-performance trade-offs in PEFT settings, providing a principled framework for comparing adapter methods along the rate-distortion curve. While our experiments focus on post-training quantization, the proposed method is modular and readily compatible with quantization-aware training or low-precision inference schemes. We view this work as a step toward scalable and bandwidth-efficient model adaptation, with potential applications in federated or multi-tenant deployment settings.

Acknowledgments

This research publication was supported in part by the CommBank Centre for Foundational AI Research.

References

Albert, P.; Saratchandran, H.; Zhang, F. Z.; Rodriguez-Opazo, C.; van den Hengel, A.; and Abbasnejad, E. 2025. RandLoRA: Full Rank Parameter-Efficient Fine-Tuning of Large Models. In *The Thirteenth International Conference on Learning Representations*.

Badri, H.; and Shaji, A. 2024. Towards 1-Bit Machine Learning Models. https://mobiusml.github.io/1bit_blog/.

Barman, N.; Martini, M. G.; and Reznik, Y. 2022. Revisiting Bjontegaard Delta Bitrate (BD-BR) Computation for Codec Compression Efficiency Comparison. In *Proceedings of the 1st Mile-High Video Conference*, MHV '22, 113–114. New York, NY, USA: Association for Computing Machinery.

Bisk, Y.; Zellers, R.; Le bras, R.; Gao, J.; and Choi, Y. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05): 7432–7439.

Bjøntegaard, G. 2001. Calculation of Average PSNR Differences Between RD-Curves. Technical report, VCEG-M33, Austin, TX, USA.

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101 – Mining Discriminative Components with Random Forests. In *European Conference on Computer Vision*.

Brüel-Gabrielsson, R.; Zhu, J.; Bhardwaj, O.; Choshen, L.; Greenewald, K.; Yurochkin, M.; and Solomon, J. 2025. Compress then Serve: Serving Thousands of LoRA Adapters with Little Overhead. arXiv:2407.00066.

Caron, M.; Touvron, H.; Misra, I.; Jegou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9630–9640.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing Textures in the Wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 3606–3613.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2924–2936. Minneapolis, Minnesota: Association for Computational Linguistics.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think You Have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv preprint arXiv:1803.05457*.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. QLORA: Efficient Finetuning of Quantized LLMs. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Red Hook, NY, USA: Curran Associates Inc.

Ding, N.; Lv, X.; Wang, Q.; Chen, Y.; Zhou, B.; Liu, Z.; and Sun, M. 2023. Sparse Low-rank Adaptation of Pre-trained Language Models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Edalati, A.; Tahaei, M.; Kobyzev, I.; Nia, V. P.; Clark, J. J.; and Rezagholizadeh, M. 2025. KronA: Parameter-Efficient Tuning with Kronecker Adapter. In *Enhancing LLM Performance: Efficacy, Fine-Tuning, and Inference Techniques*, 49–65. Springer.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *Forty-first international conference on machine learning*.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2006. One-shot Learning of Object Categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4): 594–611.

Gersho, A.; and Gray, R. M. 1991. *Vector Quantization and Signal Compression*. USA: Kluwer Academic Publishers.

Gholami, A.; Kim, S.; Dong, Z.; Yao, Z.; Mahoney, M. W.; and Keutzer, K. 2022. A Survey of Quantization Methods for Efficient Neural Network Inference. In *Low-Power Computer Vision*, 291–326. Chapman and Hall/CRC.

Gray, R.; and Neuhoff, D. 1998. Quantization. *IEEE Transactions on Information Theory*, 44(6): 2325–2383.

Grønlund, A.; Larsen, K. G.; Mathiasen, A.; Nielsen, J. S.; Schneider, S.; and Song, M. 2018. Fast Exact k-Means, k-Medians and Bregman Divergence Clustering in 1D. arXiv:1701.07204.

Hadish, S.; Bojković, V.; Aloqaily, M.; and Guizani, M. 2024. Language Models at the Edge: A Survey on Techniques, Challenges, and Applications. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, 262–271.

Han, S.; Mao, H.; and Dally, W. J. 2015. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv preprint arXiv:1510.00149*.

Han, Z.; Gao, C.; Liu, J.; Zhang, J.; and Zhang, S. Q. 2024. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. *Transactions on Machine Learning Research*.

He, X.; Li, C.; Zhang, P.; Yang, J.; and Wang, X. E. 2023. Parameter-Efficient Model Adaptation for Vision Transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 817–825.

Helber, P.; Bischke, B.; Dengel, A.; and Borth, D. 2018. Introducing Eurosat: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 204–207.

Herglotz, C.; Kränzler, M.; Mons, R.; and Kaup, A. 2022. Beyond Bjøntegaard: Limits of Video Compression Performance Comparisons. In *International Conference on Image Processing (ICIP)*.

Herglotz, C.; Och, H.; Meyer, A.; Ramasubbu, G.; Eichermüller, L.; Kränzler, M.; Brand, F.; Fischer, K.; Nguyen, D. T.; Regensky, A.; and Kaup, A. 2024. The Bjøntegaard Bible Why Your Way of Comparing Video Codecs May Be Wrong. *IEEE Transactions on Image Processing*, 33: 987–1001.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-Efficient Transfer Learning for NLP. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 2790–2799. PMLR.

Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Isik, B.; Kumbong, H.; Ning, W.; Yao, X.; Koyejo, S.; and Zhang, C. 2023. GPT-Zip: Deep Compression of Finetuned Large Language Models. In *Workshop on Efficient Systems for Foundation Models @ ICML2023*.

Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; and Kalenichenko, D. 2018. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ji, Y.; Saratchandran, H.; Gordon, C.; Zhang, Z.; and Lucey, S. 2025. Efficient Learning with Sine-Activated Low-Rank Matrices. In *The Thirteenth International Conference on Learning Representations*.

Jiang, Y.; Yang, Z.; Chen, B.; Li, S.; Li, Y.; and Li, T. 2024. DeltaDQ: Ultra-High Delta Compression for Fine-Tuned LLMs via Group-wise Dropout and Separate Quantization. arXiv:2410.08666.

Jie, S.; Wang, H.; and Deng, Z.-H. 2023. Revisiting the Parameter Efficiency of Adapters from the Perspective of Precision Redundancy. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 17171–17180.

Karimi Mahabadi, R.; Henderson, J.; and Ruder, S. 2021. Compacter: Efficient Low-Rank Hypercomplex Adapter Layers. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 1022–1035. Curran Associates, Inc.

Kaushal, A.; Vaidhya, T.; Mondal, A. K.; Pandey, T.; Bhagat, A.; and Rish, I. 2025. Surprising Effectiveness of pretraining Ternary Language Model at Scale. In *The Thirteenth International Conference on Learning Representations*.

Koohpayegani, S. A.; L, N. K.; Nooralinejad, P.; Kolouri, S.; and Pirsiavash, H. 2024. NOLA: Compressing LoRA using Linear Combination of Random Basis. In *The Twelfth International Conference on Learning Representations*.

Kopiczko, D. J.; Blankevoort, T.; and Asano, Y. M. 2024. VeRA: Vector-Based Random Matrix Adaptation. In *The Twelfth International Conference on Learning Representations*.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3D Object Representations for Fine-Grained Categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, 554–561.

Li, M.; Huang, Z.; Chen, L.; Ren, J.; Jiang, M.; Li, F.; Fu, J.; and Gao, C. 2024. Contemporary Advances in Neural Network Quantization: A Survey. In *2024 International Joint Conference on Neural Networks (IJCNN)*, 1–10.

Li, Y.; Song, L.; and Hou, H. 2024. LoRAN: Improved Low-Rank Adaptation by a Non-Linear Transformation. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 3134–3143. Miami, Florida, USA: Association for Computational Linguistics.

Liu, J.; Xiao, G.; Li, K.; Lee, J. D.; Han, S.; Dao, T.; and Cai, T. 2024a. BitDelta: Your Fine-Tune May Only Be Worth One Bit. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Liu, S.-Y.; Wang, C.-Y.; Yin, H.; Molchanov, P.; Wang, Y.-C. F.; Cheng, K.-T.; and Chen, M.-H. 2024b. DoRA: Weight-Decomposed Low-Rank Adaptation. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 32100–32121. PMLR.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Maji, S.; Kannala, J.; Rahtu, E.; Blaschko, M.; and Vedaldi, A. 2013. Fine-Grained Visual Classification of Aircraft. arXiv:1306.5151.

Mao, Y.; Ge, Y.; Fan, Y.; Xu, W.; Mi, Y.; Hu, Z.; and Gao, Y. 2025. A Survey on LoRA of Large Language Models. *Frontiers of Computer Science*, 19(7): 197605.

Martinsson, P.-G.; and Tropp, J. A. 2020. Randomized Numerical Linear Algebra: Foundations and Algorithms. *Acta Numerica*, 29: 403–572.

Mihaylov, T.; Clark, P.; Khot, T.; and Sabharwal, A. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2381–2391. Brussels, Belgium: Association for Computational Linguistics.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated Flower Classification over a Large Number of Classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 722–729.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. V. 2012. Cats and Dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ping, B.; Wang, S.; Wang, H.; Han, X.; Xu, Y.; Yan, Y.; Chen, Y.; Chang, B.; Liu, Z.; and Sun, M. 2024. Delta-CoMe: Training-Free Delta-Compression with Mixed-Precision for Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A. 2016. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In *ECCV*.

Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2023. Dreambooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 22500–22510.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252.

Ryu, S.; Seo, S.; and Yoo, J. 2023. Efficient Storage of Fine-Tuned Models via Low-Rank Approximation of Weight Residuals. arXiv:2305.18425.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. *Communications of the ACM*, 64(9): 99–106.

Sap, M.; Rashkin, H.; Chen, D.; LeBras, R.; and Choi, Y. 2019. SocialIQA: Commonsense Reasoning about Social Interactions. *arXiv preprint arXiv:1904.09728*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. arXiv:1212.0402.

Steinberg, D. 2019. KMeans1D: Globally Optimal Efficient 1D k-Means. Accessed: 2025-02-18.

Wang, D.; Zhu, W.; Xu, Y.; Xu, Y.; and LeYang. 2021a. Visual Quality Optimization for View-Dependent Point Cloud Compression. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5.

Wang, J.; Zhu, H.; Liu, H.; and Ma, Z. 2021b. Lossy Point Cloud Geometry Compression via End-to-End Learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(12): 4909–4923.

Wu, X. 1991. Optimal Quantization by Matrix Searching. *Journal of Algorithms*, 12(4): 663–673.

Xiao, J.; Ehinger, K. A.; Hays, J.; Torralba, A.; and Oliva, A. 2016. SUN Database: Exploring a Large Collection of Scene Categories. *International Journal of Computer Vision*, 119(1): 3–22.

Xu, Y.; Xie, L.; Gu, X.; Chen, X.; Chang, H.; Zhang, H.; Chen, Z.; Zhang, X.; and Tian, Q. 2024. QA-LoRA: Quantization-Aware Low-Rank Adaptation of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Yao, X.; Hu, Q.; and Klimovic, A. 2025. DeltaZip: Efficient Serving of Multiple Full-Model-Tuned LLMs. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, 110–127. New York, NY, USA: Association for Computing Machinery.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4791–4800. Florence, Italy: Association for Computational Linguistics.

Zhang, F. Z.; Albert, P.; Rodriguez-Opazo, C.; van den Hengel, A.; and Abbasnejad, E. 2024. Knowledge Composition using Task Vectors with Learned Anisotropic Scaling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhu, X.; Li, J.; Liu, Y.; Ma, C.; and Wang, W. 2024. A Survey on Model Compression for Large Language Models. *Transactions of the Association for Computational Linguistics*, 12: 1556–1577.