AdaNF: Quantization Group Adaptive NormalFloat for Low Bit Fine-tuning of LLMs

Anonymous Authors[1]

Abstract

The integration of Quantization and Low-Rank Adaptation (LoRA) presents a promising avenue for the memory-efficient fine-tuning of large language models (LLMs) within GPU memory constraints. QLoRA, introduced by (Dettmers et al., 2024), successfully demonstrates high-fidelity 4-bit fine-tuning using an information-theoretically optimal datatype, NormalFloat. However, challenges arise with lower-bit fine-tuning, such as 2-bit, where QLoRA often struggles with convergence due to significant information loss from quantization. In this study, we address these challenges by adjusting the cumulative distribution function (CDF) offset of NormalFloat, which significantly reduces information loss through improved NormalFloat initialization. Furthermore, we introduce quantization group **Ada**ptive **N**ormal**F**loat (AdaNF), a technique that dynamically adjusts the NormalFloat CDF offset based on the statistical characteristics of each quantization group in the parameters. This adaptive approach minimizes the Lp norm of the quantization error through a grid search, allowing for customized quantization that preserves more information. Our empirical investigations across various models and downstream tasks in the low-bit fine-tuning regime confirm that our method achieves performance comparable to existing methods, effectively mitigating the limitations of prior approaches.

1. Introduction

The emergence of Large Language Models (LLMs) has brought about a paradigm shift in AI technology, demonstrating remarkable performance across a wide range of Natural Language Processing (NLP) applications (Brown et al., 2020; Achiam et al., 2023; Touvron et al., 2023a;b; Chowdhery et al., 2023; Team et al., 2023; Jiang et al., 2023). LLMs excel as few-shot learners, performing various downstream tasks through in-context learning with just a few examples (Dong et al., 2022). However, in specialized domains requiring detailed knowledge not typically covered in general training corpora, LLMs benefit significantly from fine-tuning to enhance accuracy (Liu et al., 2022). Despite its effectiveness, full fine-tuning has become impractical due to the substantial GPU resources required for the massive parameters. To address these constraints, reducing memory usage for optimizer states, gradients, and model weights has been a focus. Low-Rank Adaptation (LoRA) (Hu et al., 2021) is a widely adopted method that achieves this by significantly reducing the number of trainable parameters, representing the difference between frozen pre-trained weights and fully fine-tuned weights using only trainable low-rank matrices.

Further reductions in memory usage can be achieved through the quantization of model weights. QLoRA (Dettmers et al., 2024) successfully combines LoRA with quantization for the first time, demonstrating high performance in 4-bit quantized fine-tuning while significantly reducing GPU memory requirements without incurring additional costs. Nevertheless, when it comes to extremely low bit fine-tuning regimes, such as 2-bit, QLoRA often fails to converge on many downstream tasks (Li et al., 2023). This suggests that LoRA fine-tuning alone is insufficient to recover from the substantial information loss caused by low-bit model quantization. Even though some efforts have been made to mitigate this information loss through the strategic initialization of LoRA components (Li et al., 2023; Guo et al., 2023), a fundamental redesign of the NormalFloat quantization data type used in QLoRA is necessary to make low-bit fine-tuning more practical.

The motivation for redesigning the NormalFloat data type stems from understanding why the original NormalFloat experiences significant information loss in 2-bit fine-tuning. As illustrated in Figure 1, when an outlier is present in a quantization group, the original NormalFloat bases its dequantization on the group's maximum value, which cor-

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

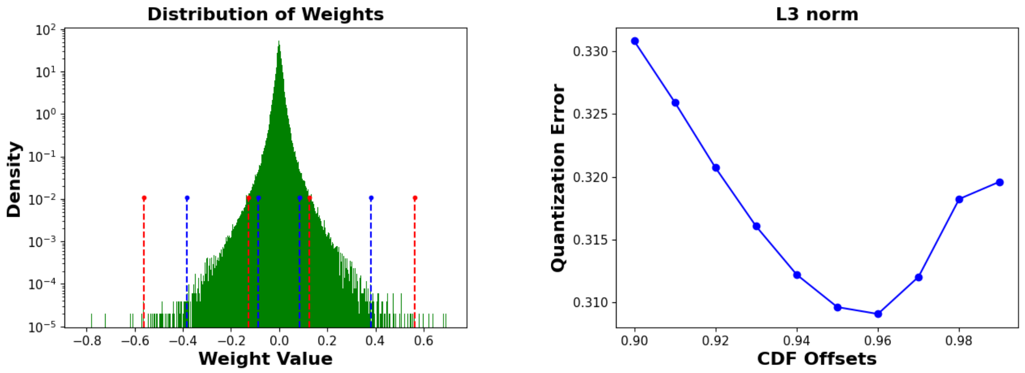

Figure 1. The left figure shows the log-scale weight distribution of a specific layer in LLAMA-2-7B. The red points represent the four dequantized values within a particular quantization group using the original 2-bit NormalFloat. With our redesigned NormalFloat, we obtain four blue points that are closer to the center. The right figure illustrates how the L3 norm of the quantization error varies with the CDF offset (See section 2 and Appx. B.1) within the same quantization group. The four dequantized blue points in the left figure are obtained from an offset of 0.96, resulting in minimal quantization error in this group. By adjusting the offset, we can find the optimal value for each group that minimizes the quantization error.

responds to the outlier. Consequently, the four dequantized values (red points in the left figure of Figure 1) are not representative, as 50% of them are outliers, leading to high quantization error. Therefore, in the 2-bit regime, it is crucial to bring these dequantized values closer to the center (blue points in the left figure of Figure 1), as this adjustment can significantly reduce the quantization error.

To address this issue, we first introduce an updated version of the NormalFloat data type, which we call Dynamic NormalFloat. This new version adjusts the dequantized values based on the ratio of the quantile output of the reference CDF offset to the quantile output of our chosen CDF offset. By selecting a lower CDF offset, the adjustment from the red points to the blue points in Figure 1 is achieved through this redesigned data type. Additionally, since each quantization group has unique statistical characteristics, we propose the quantization group **Ada**ptive **N**ormal**F**loat (AdaNF), which identifies the optimal CDF offset for each quantization group through grid search by minimizing the Lp norm of the quantization error (see the right figure of Figure 1). We evaluate our quantization framework through experiments on various models and downstream tasks in the low-bit fine-tuning regime. Our method outperforms existing approaches for 2-bit fine-tuning and shows comparable performance for 3-bit and 4-bit fine-tuning.

2. Method

In this section, we propose quantization group **Ada**ptive **N**ormal**F**loat (AdaNF), an advantageous quantization data type for low-bit LLM fine-tuning. We begin by introducing the concept of Dynamic NormalFloat with a single offset, which involves adjusting the NormalFloat initialization based on the ratio between the quantile output, determined by a specific CDF offset, and a reference quantile value. This adjustment aims to reduce information loss from quantized weight parameters. Additionally, we describe how to dynamically determine the improved NormalFloat initialization for each quantization group using this newly defined Dynamic NormalFloat.

2.1. Dynamic NormalFloat with a Single Offset

In the original NormalFloat (Dettmers et al., 2024), they set a default CDF offset c_{offset} to 0.9677083. Then, they obtain the normalized quantization map within the range $[-1, 1]$ by dividing all quantile values from symmetric NormalFloat (see Definition B.1) or asymmetric NormalFloat (see Definition B.2) by the maximum quantile value $Q(c_{\text{offset}})$, as explained in Appx. B.1. Using this k-bit normalized NormalFloat quantization map q^k_{NF} (see (4)), the quantized output of the i-th group tensor W^{flat}_i is obtained as shown in (5)[1]. The dequantized weight tensor $W^{\text{flat}}_{\text{deq},i}$ is then calculated from this quantized tensor $W^{\text{flat}}_{q,i}$ as follows:

$$\begin{aligned} W^{\text{flat}}_{\text{deq},i} &= \text{absmax}(W^{\text{flat}}_i) \cdot W^{\text{flat}}_{q,i} \\ &= \text{absmax}(W^{\text{flat}}_i) \cdot Q^k_{\text{NF}}\left(\frac{W^{\text{flat}}_i}{\text{absmax}(W^{\text{flat}}_i)}\right) \end{aligned}$$

Since the maximum and minimum outputs of the function Q^k_{NF} are 1 and -1, respectively, $\text{absmax}(W^{\text{flat}}_i)$ and $-\text{absmax}(W^{\text{flat}}_i)$ will be the maximum and minimum among the possible dequantized values for the i-th quantization group. In low-bit fine-tuning, such as 2-bit, all four possible

[1] The detailed formulation of the original NormalFloat, with respect to a CDF offset, and the group quantization process are elaborated in Appx. B.1.

dequantized values should be representative within the i-th quantization group. However, if there is an extreme outlier in the group, $\text{absmax}(W_i^{\text{flat}})$ and $-\text{absmax}(W_i^{\text{flat}})$ may not be representative, leading to significant quantization errors.

To address this issue, we propose Dynamic NormalFloat, which adjusts the maximum and minimum dequantized values inward (i.e., reduces their absolute values) compared to $\text{absmax}(W_i^{\text{flat}})$ and $-\text{absmax}(W_i^{\text{flat}})$. This ensures that all possible dequantized values are more representative within the group. This adjustment is achieved by further reducing the quantization output range of Q^k_{NF} from $[-1, 1]$. The updated quantization data type is defined as follows:

Definition 2.1. (Dynamic NormalFloat) Let Q be the quantile function of the standard normal distribution $N(0, 1)$. For a CDF offset $c_{\text{offset}} \in (0.5, 1.0)$, each i-th quantized value of the k-bit NormalFloat data type, denoted as q_i, is derived from either Definition B.1 or Definition B.2. The reference offset is c_{ref}. The k-bit Dynamic NormalFloat data type q_{DNF} is then represented as follows:

$$q^k_{\text{DNF}}|_{c_{\text{offset}},c_{\text{ref}}} = \frac{q|_{c_{\text{offset}}}}{Q(c_{\text{ref}})} = \frac{[q_1, q_2, \cdots, q_{2^k}]}{Q(c_{\text{ref}})} \tag{1}$$

The range of q_{DNF} is $\left[-\frac{Q(c_{\text{offset}})}{Q(c_{\text{ref}})}, \frac{Q(c_{\text{offset}})}{Q(c_{\text{ref}})}\right]$. Therefore, if c_{offset} is less than c_{ref}, the range of q_{DNF} becomes smaller than $[-1, 1]$. This means that the absolute value of a dequantized weight parameter must be less than $\text{absmax}(W_i^{\text{flat}})$. For the j-th element of the dequantized weight tensor in group i, we have:

$$\begin{aligned}|W^{\text{flat}}_{\text{deq},i}(j)| &= \text{absmax}(W_i^{\text{flat}}) \left| Q^k_{\text{DNF}}\left(\frac{W_i^{\text{flat}}(j)}{\text{absmax}(W_i^{\text{flat}})}\right)\right| \\ &\le \text{absmax}(W_i^{\text{flat}}) \frac{Q(c_{\text{offset}})}{Q(c_{\text{ref}})} < \text{absmax}(W_i^{\text{flat}}),\end{aligned}$$

where Q^k_{DNF} is the k-bit Dynamic NormalFloat quantization function defined by q^k_{DNF}. Thus, with a properly chosen c_{offset}, q^k_{DNF} can potentially result in less quantization error compared to the original quantization map q^k_{NF}.

2.2. Quantization Group Adaptive NormalFloat

We now introduce quantization group **Ada**ptive **N**ormal**F**loat (AdaNF), which dynamically determines an appropriate CDF offset for the Dynamic NormalFloat data type in each quantization group to minimize quantization error. Since each quantization group has unique statistical characteristics, adjusting the CDF offset for each group can more effectively preserve information during low-bit quantization compared to using a single offset. We measure the quantization error between the original weight tensor and the dequantized weight tensor using the Lp norm. This error metric is then used to identify the optimal CDF offset for each group through grid search. Finding an optimal order of the norm p for each case is crucial. A p value that is too large will cause the quantization to be overly influenced by outliers, while a p value that is too small will ignore outliers entirely. Therefore, it is essential to strike a balance by selecting an appropriate p that adequately considers outliers without being dominated by them. Our algorithm addresses this by exploring and tuning p to achieve this balance. The detailed algorithm is provided in Algorithm 1.

Algorithm 1 Grid Search for AdaNF

```
Input: original weight of a group $W$, reference CDF offset $c_{\text{ref}}$, number of grids $n$, start grid for CDF offset $c_{\text{start}}$, end grid for CDF offset $c_{\text{end}}$, order of the norm $p$
Initialize $c^* = c_{\text{start}}$, $E^* = \text{inf}$
for $i = 1, \cdots, n$ do
    $c_{\text{offset}} = c_{\text{start}} + \frac{c_{\text{end}} - c_{\text{start}}}{n-1}(i-1)$
    create the $k$-bit Dynamic NormalFloat $q^k_{\text{DNF}}|_{c_{\text{offset}},c_{\text{ref}}}$
    perform nearest rounding with $q^k_{\text{DNF}}|_{c_{\text{offset}},c_{\text{ref}}}$:
    $W_q = Q^k_{\text{DNF}}|_{c_{\text{offset}},c_{\text{ref}}}\left(\frac{W}{\text{absmax}(W)}\right)$
    get the dequantized weight $W_{\text{deq}} = \text{absmax}(W)W_q$
    compute the quantization error $E = \|W - W_{\text{deq}}\|_p$
    if $E < E^*$ then
        $c^* \leftarrow c_{\text{offset}}$, $E^* \leftarrow E$
    end if
end for
return $c^*$
```

After obtaining the optimal CDF offset for each quantization group through Algorithm 1, we initialize each Dynamic NormalFloat data type with the corresponding offset. Subsequently, the weight parameters are quantized using group quantization, as detailed in Appx. B.1.1, and these quantized weights remain fixed during the fine-tuning process. For the fine-tuning, we employ the LoRA method, as described in Appx. B.2.

3. Experiments

In this section, we present experimental results for Dynamic NormalFloat (DNF) and AdaNF [2] on Natural Language Understanding (NLU) and Natural Language Generation (NLG) tasks. We compare our algorithm with QLoRA (Dettmers et al., 2024) and other low-bit fine-tuning methods, such as LoftQ (Li et al., 2023) and ApiQ (Liao & Monz, 2024). Additionally, we use full fine-tuning and full precision LoRA (Hu et al., 2021) for reference. For the NLU task, we empirically assess the performance of these algorithms by quantizing the encoder-only DeBERTaV3-base model (He et al., 2021) and fine-tuning it on the General

[2] The more accurate terminology would be QLoRA with DNF and QLoRA with AdaNF, as DNF and AdaNF refer to quantization data types. However, for simplicity, we use the terms DNF and AdaNF.

Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018). For the NLG task, we evaluate the performance by quantizing the decoder-only LLAMA-2-7B model (Touvron et al., 2023b) and fine-tuning it on two NLG datasets: WikiText-2 (Merity et al., 2016) and GSM8k (Cobbe et al., 2021). The experimental setup details about implementation, datasets, and hyperparameter choice are provided in Appx. C.1.

3.1. Experimental Results

3.1.1. NLU with DeBERTaV3-base

We begin with Natural Language Understanding (NLU) experiments utilizing the relatively smaller DeBERTaV3-base model. The outcomes of these experiments are detailed in Table 2 in Appx. C.2. We assess the 2-bit fine-tuning effectiveness of our quantization methods, DNF with a single CDF offset and AdaNF, in comparison to two baselines: QLoRA and LoftQ, across 8 different tasks in the GLUE benchmark. For AdaNF, we explore two scenarios: one employing the L2.5 norm and the other using the L3 norm to evaluate quantization error. In terms of evaluation metrics, a higher score indicates better performance across all 8 tasks.

When comparing the best results of our methods with QLoRA for each task, our methods outperform QLoRA across all 8 tasks in the GLUE benchmark. This confirms that our redesigned quantization data types, DNF and AdaNF, are indeed improved versions of the original NormalFloat. Notably, AdaNF with the L3 norm consistently surpasses QLoRA in all tasks, demonstrating that the L3 norm is particularly effective for measuring quantization error. This norm effectively balances the influence of outliers within each quantization group, leading to reduced information loss and superior performance in 2-bit fine-tuning (for more detailed insights, see section 2.2).

When comparing our best results to those of LoftQ, our methods show better performance in three tasks: MNLI, QNLI, and SST-2. This suggests that our quantization approach is already competitive with the current state-of-the-art LoftQ. Additionally, when comparing our methods internally, AdaNF generally outperforms DNF in nearly all cases, except for the MNLI task when comparing DNF with AdaNF using the L2.5 norm. This observation supports the underlying intuition of the AdaNF algorithm: adaptively finding the optimal CDF offset for each quantization group based on minimal quantization error with the Lp norm leads to better performance than applying the same offset to all quantization groups.

3.1.2. NLG with LLAMA-2-7B

To assess the scalability of our methods, we also conducted Natural Language Generation (NLG) experiments using the larger LLAMA-2-7B model. The results of these experiments are summarized in Table 1. We compare the low-bit fine-tuning performance of our quantization algorithms, DNF with a single CDF offset and AdaNF, against three baselines: QLoRA, LoftQ, and ApiQ, on WikiText-2 and GSM8k. For AdaNF, we evaluate two cases: one using the L2 norm and another using the L3 norm to measure quantization error. The evaluation metrics used are perplexity for WikiText-2 and accuracy for GSM8K.

Table 1. Quantitative results on two NLG tasks with LLAMA-2-7B. We compare our methods, DNF and AdaNF, against three other quantized fine-tuning baselines. For reference, LoRA fine-tuning without quantization, which is not included in the table, achieves a perplexity of 5.08 on WikiText-2 and an accuracy of 38.5 on GSM8K. N.A. means the model fails to converge.

	WikiText-2 (↓)			GSM8K (↑)		
	4bit	3bit	2bit	4bit	3bit	2bit
QLoRA	5.70	5.73	N.A.	**38.2**	32.1	N.A.
LoftQ	5.24	5.63	7.85	38.0	**36.2**	26.5
ApiQ	5.28	5.53	7.46	36.4	36.0	26.0
DNF	5.21	5.55	6.93	35.4	33.7	**27.6**
AdaNF (L2)	**5.19**	**5.48**	6.88	36.7	32.4	22.8
AdaNF (L3)	**5.19**	**5.48**	**6.80**	35.8	33.5	25.5

For the WikiText-2 experiments, all our methods demonstrate improved perplexity than the three other baselines across 2-bit, 3-bit, and 4-bit settings, with the exception of the DNF 3-bit case. However, even in this instance, the perplexity of DNF 3-bit is only slightly higher than that of ApiQ 3-bit, the most recent of the three baselines. Notably, in the challenging 2-bit scenario, AdaNF with L3 norm achieves the best perplexity score of 6.80, where QLoRA fails to converge. Overall, AdaNF with L3 norm consistently shows the best performance on WikiText-2, indicating that the L3 norm effectively captures quantization error and aids AdaNF in finding the optimal CDF offset for each quantization group, minimizing information loss. Additionally, it is evident that both versions of AdaNF outperform DNF in terms of perplexity across all bit settings.

For the GSM8K experiments, our method demonstrates outstanding performance in the challenging 2-bit case. Specifically, DNF with a single CDF offset achieves an accuracy of 27.6, surpassing the 26.5 achieved by LoftQ (Li et al., 2023), the current state-of-the-art for 2-bit fine-tuning on GSM8K. While DNF outperforms all baselines in the 2-bit fine-tuning scenario, its accuracy for 3-bit and 4-bit fine-tuning is lower compared to other methods. For instance, in the 3-bit setting, both DNF and AdaNF improve upon the original QLoRA but still trail behind LoftQ and ApiQ. Further optimization of our algorithm, such as finer hyperparameter tuning, could enhance these results (see Algorithm 1).

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

An, S., Li, Y., Lin, Z., Liu, Q., Chen, B., Fu, Q., Chen, W., Zheng, N., and Lou, J.-G. Input-tuning: Adapting unfamiliar inputs to frozen pretrained models. *arXiv preprint arXiv:2203.03131*, 2022.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

Chee, J., Cai, Y., Kuleshov, V., and De Sa, C. M. Quip: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems*, 36, 2024.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dagan, I., Glickman, O., and Magnini, B. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pp. 177–190. Springer, 2005.

Dettmers, T., Lewis, M., Shleifer, S., and Zettlemoyer, L. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*, 2021.

Dettmers, T., Svirschevski, R., Egiazarian, V., Kuznedelev, D., Frantar, E., Ashkboos, S., Borzunov, A., Hoefler, T., and Alistarh, D. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*, 2023.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Dolan, B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*, 2005.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022a.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Optq: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2022b.

Guo, H., Greengard, P., Xing, E. P., and Kim, Y. Lq-lora: Low-rank plus quantized matrix decomposition for efficient language model finetuning. *arXiv preprint arXiv:2311.12023*, 2023.

He, P., Gao, J., and Chen, W. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Karimi Mahabadi, R., Henderson, J., and Ruder, S. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34: 1022–1035, 2021.

Kim, S., Hooper, C., Gholami, A., Dong, Z., Li, X., Shen, S., Mahoney, M. W., and Keutzer, K. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*, 2023.

Kopiczko, D. J., Blankevoort, T., and Asano, Y. M. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Li, Y., Yu, Y., Liang, C., He, P., Karampatziakis, N., Chen, W., and Zhao, T. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*, 2023.

Liao, B. and Monz, C. Apiq: Finetuning of 2-bit quantized large language model. *arXiv preprint arXiv:2402.05147*, 2024.

Lin, J., Tang, J., Tang, H., Yang, S., Dang, X., and Han, S. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35: 1950–1965, 2022.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Nikdan, M., Tabesh, S., and Alistarh, D. Rosa: Accurate parameter-efficient fine-tuning via robust adaptation. *arXiv preprint arXiv:2401.04679*, 2024.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Qin, G. and Eisner, J. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*, 2021.

Qin, H., Ma, X., Zheng, X., Li, X., Zhang, Y., Liu, S., Luo, J., Liu, X., and Magno, M. Accurate lora-finetuning quantization of llms via information retention. *arXiv preprint arXiv:2402.05445*, 2024.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Renduchintala, A., Konuk, T., and Kuchaiev, O. Tied-lora: Enhacing parameter efficiency of lora with weight tying. *arXiv preprint arXiv:2311.09578*, 2023.

Shao, W., Chen, M., Zhang, Z., Xu, P., Zhao, L., Li, Z., Zhang, K., Gao, P., Qiao, Y., and Luo, P. Omniquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*, 2023.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Sung, Y.-L., Nair, V., and Raffel, C. A. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34:24193–24205, 2021.

Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Warstadt, A., Singh, A., and Bowman, S. R. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Xia, W., Qin, C., and Hazan, E. Chain of lora: Efficient fine-tuning of language models via residual learning. *arXiv preprint arXiv:2401.04151*, 2024.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pp. 38087–38099. PMLR, 2023.

Zaken, E. B., Ravfogel, S., and Goldberg, Y. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., and Zhao, T. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

A. Related Work

Due to the significant computational and memory demands of Large Language Models (LLMs), numerous parameter-efficient fine-tuning (PEFT) methods have been developed. These methods reduce memory and computational costs by optimizing a much smaller subset of parameters compared to the original LLMs (Hu et al., 2021; Houlsby et al., 2019; Qin & Eisner, 2021; Lester et al., 2021; Li & Liang, 2021; An et al., 2022; Liu et al., 2022; Zaken et al., 2021; Sung et al., 2021; Karimi Mahabadi et al., 2021). Among these, LoRA (Hu et al., 2021) is the most widely adopted. It trains low-rank adapter layers on top of the frozen base model, offering stable training due to the implicit regularization of low-rank adaptation. Furthermore, LoRA facilitates easy modular adaptation across different tasks. Many LoRA variants have since been introduced to closely match the performance of full fine-tuning on more challenging downstream tasks (Zhang et al., 2023; Renduchintala et al., 2023; Kopiczko et al., 2023; Xia et al., 2024; Nikdan et al., 2024).

Combining model quantization with LoRA can further reduce GPU memory consumption during the fine-tuning of LLMs. While many model quantization methods have been developed mainly for inference purposes (Frantar et al., 2022a;b; Shao et al., 2023; Xiao et al., 2023; Dettmers et al., 2023; Kim et al., 2023; Lin et al., 2023; Chee et al., 2024), QLoRA (Dettmers et al., 2024) is the first to demonstrate that fine-tuning a quantized 4-bit model with minimal performance degradation is possible by combining NormalFloat quantization with a small set of learnable low-rank adapter weights. However, in extremely low-bit scenarios like the 2-bit regime, QLoRA suffers from significant performance degradation due to substantial weight information loss. To mitigate this, (Li et al., 2023; Guo et al., 2023) explore strategic initialization of LoRA matrices. (Liao & Monz, 2024) proposes a different approach to quantized fine-tuning by focusing on minimizing activation error instead of weight error. (Qin et al., 2024) addresses information loss from low-bit quantization by calibrating a bias constant groupwisely based on information entropy maximization. Nevertheless, to the best of our knowledge, our AdaNF quantization is the first to highlight the importance of CDF offset initialization in NormalFloat for low-bit fine-tuning, further improving performance through adaptive initialization of the NormalFloat offset for each quantization group.

B. Preliminaries

B.1. NormalFloat Quantization

Given that we employ NormalFloat as the framework for our quantization in the low bit LLM fine-tuning, we first present the definition of NormalFloat (Dettmers et al., 2024).

Definition B.1. (Symmetric NormalFloat) Q is the quantile function of the standard normal distribution $N(0,1)$, also known as the inverse cumulative distribution function (CDF). Then, for the CDF offset $c_{\text{offset}} \in (0.5, 1.0)$, each ith quantized value of the k-bit symmetric NormalFloat data type is represented as

$$q_i = Q\Big(1 - c_{\text{offset}} + \frac{2c_{\text{offset}} - 1}{2^k - 1} \times (i-1)\Big) \tag{2}$$

for all $i = 1, 2, \cdots, 2^k$.

The NormalFloat (NF) data type is based on Quantile Quantization (Dettmers et al., 2021), an information-theoretically optimal data type that ensures each quantization bin contains an equal number of values from the input tensor. (2) in Definition B.1 means the 2^k equally spaced quantiles over the range of probabilities $[1 - c_{\text{offset}}, c_{\text{offset}}]$. Similarly, we can define the asymmetric NormalFloat that includes 0, $(2^{k-1} - 1)$ negative values, and 2^{k-1} positive values.

Definition B.2. (Asymmetric NormalFloat) Q is the quantile function of the standard normal distribution $N(0,1)$. Then, for the CDF offset $c_{\text{offset}} \in (0.5, 1.0)$, each ith quantized value of the k-bit asymmetric NormalFloat data type is represented as

$$q_i = \begin{cases} Q\left(1 - c_{\text{offset}} + \frac{0.5-(1-c_{\text{offset}})}{2^{k-1}-1} \times (i-1)\right), & \text{if } 1 \le i < 2^{k-1}, \\ Q(0.5)(=0), & \text{if } i = 2^{k-1}, \\ Q\left(0.5 + \frac{c_{\text{offset}}-0.5}{2^{k-1}} \times (i - 2^{k-1})\right), & \text{if } 2^{k-1} < i \le 2^k. \end{cases} \tag{3}$$

For (3) in Definition B.2, the first case means 2^{k-1} equally spaced quantiles over the range of probabilities $[1 - c_{\text{offset}}, 0.5]$, and the third case means $2^{k-1} + 1$ equally spaced quantiles over the range of probabilities $[0.5, c_{\text{offset}}]$. After obtaining discrete values from either symmetric NormalFloat or asymmetric NormalFloat, we normalize them to the range $[-1, 1]$ by dividing each value by the maximum value $Q(c_{\text{offset}})$. Thus, the exact values of the normalized k-bit NormalFloat data type are as follows:

$$q^k_{\text{NF}} = \frac{q}{Q(c_{\text{offset}})} = \frac{[q_1, q_2, \cdots, q_{2^k}]}{Q(c_{\text{offset}})} \tag{4}$$

This allows us to quantize the input weight parameters by normalizing them into the same range $[-1, 1]$ via absolute maximum rescaling.

B.1.1. Group quantization

In addition to using NormalFloat as our quantization data type, our framework relies on group quantization to effectively handle outlier issues in the weight parameters. Group quantization involves dividing the input tensor into smaller chunks that are independently quantized. This approach indirectly reduces the number of outliers in each group, leading to smaller quantization errors. Group quantization can be implemented by dividing the weight tensor $W \in \mathbb{R}^{d \times h}$ into n_g contiguous groups of size G. This is done by flattening the weight tensor into a vector $W^{\text{flat}} \in \mathbb{R}^{dh \times 1}$ and then slicing this vector into $n_g = \frac{d \times h}{G}$ quantization groups. When we define the k-bit NormalFloat quantization function as Q^k_{NF} and denote the i-th group tensor as W^{flat}_i for $1 \leq i \leq n_g$, the quantized output $W^{\text{flat}}_{q,i}$ can be expressed as

$$W^{\text{flat}}_{q,i} = Q^k_{\text{NF}}\Big(\frac{W^{\text{flat}}_i}{\text{absmax}(W^{\text{flat}}_i)}\Big) \tag{5}$$

B.2. Low-Rank Adaptation

Low-Rank Adaptation (LoRA) (Hu et al., 2021) is a Parameter Efficient Fine-Tuning (PEFT) method that reduces the memory needed for optimizer state and gradient storage by utilizing a small set of trainable parameters, while keeping the main full model parameters fixed. These finetunable parameters, known as adapters, are implemented as factorized projections that augment the original base model. This allows the forward pass to be modified through the adapted model, which can be expressed as:

$$W' = W + \alpha BA$$

where $W \in \mathbb{R}^{d \times k}$ is a pre-trained weight matrix, $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and α is a scalar. Here, we note that the rank r is much smaller than $\min(d, k)$. During backpropagation, the gradients flow through the fixed base model weights, which do not receive updates. Instead, only the small number of parameters in the low-rank adapters are updated. We use LoRA as the learnable parameters in our framework.

C. More Details about Experiments

C.1. Experimental Setup

Implementation Details We follow the implementation setup from (Li et al., 2023), with our work largely based on the HuggingFace Transformers codebase (Paszke et al., 2019). In our model implementation, we retain the original weight matrices in a frozen state and incorporate low-rank adapters into the weight matrices within the all Multi-Head Attention (MHA) and Feedforward Neural Network (FNN) layers. For the GLUE NLU task, we also quantize the embedding layer of DeBERTaV3-base. For LoRA, we use ranks of 32 for DeBERTaV3-base and 64 for LLAMA-2-7B. Model quantization is then applied to the weight matrices augmented with low-rank adapters. We perform 2-bit fine-tuning for the NLU task and 2, 3, and 4-bit fine-tuning for the NLG task. We use symmetric NormalFloat (see Definition B.1) for our DNF and AdaNF. The NVIDIA H100 80GB GPUs are used as computing resources.

Datasets For NLU, we use total 8 tasks in GLUE, which includes three natural language inference tasks: MNLI (Williams et al., 2017), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2005), two single sentence classification tasks: SST-2 (Socher et al., 2013), CoLA (Warstadt et al., 2019), and three similarity and paraphrase tasks: MRPC (Dolan & Brockett, 2005), STS-B (Cer et al., 2017), QQP. For NLG, we utilize WikiText-2, a dataset derived from Wikipedia articles, and GSM8K, also known as the Grade School Math 8K, a specialized benchmark designed to evaluate the arithmetic reasoning capabilities of language models.

C.1.1. Hyperparameter choice

When utilizing DNF quantization, we need to set two hyperparameters: the reference CDF offset c_{ref} and a specific CDF offset c_{offset}. For AdaNF quantization, five hyperparameters are required: c_{ref}, number of grids n, starting grid for CDF offset c_{start}, ending grid for CDF offset c_{end}, and the norm order p (see Algorithm 1). For all experiments, c_{ref} is set to 0.995.

In the NLU experiments with the DeBERTaV3-base model, for DNF, we set c_{offset} to 0.9 for the QNLI, SST-2, MRPC, CoLA, QQP, and STS-B tasks, and to 0.88 for the MNLI and RTE tasks. In the same NLU experiments, for AdaNF, we measure quantization error using L2.5 and L3 norms (see line 8 in Algorithm 1), meaning p can be 2.5 or 3. For $p = 2.5$, the hyperparameters $(n, c_{\text{start}}, c_{\text{end}})$ are set to (10, 0.9, 0.99) for MNLI, QNLI, SST-2, and CoLA, and (15, 0.85, 0.99) for RTE, MRPC, QQP, and STS-B. For $p = 3$, we use (15, 0.85, 0.99) only for CoLA and (10, 0.9, 0.99) for all other tasks.

In the NLG experiments with the LLAMA-2-7B model, for DNF, c_{offset} is set to 0.95, 0.98, and 0.99 for 2-bit, 3-bit, and 4-bit, respectively. For AdaNF in the same NLG experiments, we measure quantization error using L2 and L3 norms. For $p = 2$, the hyperparameters $(n, c_{\text{start}}, c_{\text{end}})$ are set to (10, 0.9, 0.99), (10, 0.9, 0.99), and (15, 0.95, 0.9967) for 2-bit, 3-bit, and 4-bit, respectively. For $p = 3$, the hyperparameters are (10, 0.9, 0.99), (15, 0.95, 0.9967), and (15, 0.95, 0.9967) for 2-bit, 3-bit, and 4-bit, respectively.

Regarding the choice of learning rate, for NLU experiments, we follow the setup in (Li et al., 2023), except for RTE, where we use 1×10^{-4}. For all NLG experiments, we use 4×10^{-4} as the learning rate.

C.2. Result Table for NLU with DeBERTaV3-base

Table 2. 2-bit fine-tuning quantitative results on the GLUE NLU tasks with the DeBERTaV3-base model. We compare our methods, DNF and AdaNF, against two other quantized fine-tuning baselines. N.A. means the model fails to converge.

	MNLI	QNLI	RTE	SST	MRPC	CoLA	QQP	STSB
	Acc(mm)	Acc	Acc	Acc	Acc	Matt	Acc	P/S Corr
Full fine-tuning	90.6	94.0	82.0	95.3	89.5/93.3	69.2	92.4/89.8	91.6/91.1
LoRA	90.5	94.6	85.1	95.1	89.9/93.6	69.9	92.0/89.4	91.7/91.1
QLoRA	78.7	80.4	56.7	86.9	73.8/82.7	N.A.	87.1/82.7	83.6/83.3
LoftQ	86.1	89.9	**61.7**	92.0	**83.6/87.2**	**47.5**	**91.0/87.9**	**87.5/87.0**
DNF	85.1	88.2	52.3	89.0	73.5/82.0	25.7	89.9/86.4	83.2/82.9
AdaNF (L2.5 norm)	31.8	**91.0**	58.1	**92.9**	75.0/83.5	39.4	90.3/87.1	85.6/85.3
AdaNF (L3 norm)	**87.0**	89.6	58.5	91.9	79.7/86.3	30.3	89.6/86.0	85.5/85.2

D. Discussion

In this paper, we introduce a novel algorithm called quantization group Adaptive NormalFloat (AdaNF) for low bit fine-tuning. We first redefine NormalFloat as Dynamic NormalFloat (DNF), which provides an improved initialization for quantization by appropriately choosing the cumulative distribution function (CDF) offset relative to a reference CDF offset. We then adaptively find the optimal CDF offset for DNF through a grid search for each quantization group, minimizing the quantization error computed using an Lp norm. Our empirical investigation demonstrates the outperforming performance of our method for 2-bit fine-tuning on natural language generation (NLG) tasks, and comparable performance on natural language understanding (NLU) tasks, corroborating our claims. Moving forward, since the strategic initialization of LoRA introduced by (Li et al., 2023) is orthogonal to our method, a careful combination of these two approaches may potentially yield further improvements in low-bit quantized fine-tuning.

Limitation Although our low-bit fine-tuning approach achieves high performance, it still lags behind full precision fine-tuning due to inherent information loss from quantization. The primary goal of low-bit fine-tuning research is to narrow this performance gap and push the limits of what is possible. We have not yet conducted comprehensive hyperparameter tuning with our method, and systematic tuning could bring our algorithm closer to its optimal performance. While our empirical results are promising, we currently lack solid theoretical guarantees to explain why our algorithm performs so well. Establishing a theoretical foundation will enhance the sustainability of our method and provide insights for further algorithmic improvements. Additionally, conducting more extensive experiments on larger language models and more challenging downstream datasets, accompanied by error bars, will strengthen our findings and demonstrate the robustness of our approach.

Broader Impacts Our work on the quantization group Adaptive NormalFloat (AdaNF) algorithm for efficient low bit fine-tuning has significant positive impacts in terms of reducing the computational and energy requirements of large language model fine-tuning, thereby mitigating the environmental impact of AI systems. It can also democratize access to state-of-the-art models for organizations with limited resources. However, quantization inherently introduces information loss, which could amplify existing biases and fairness issues in the underlying models, leading to potentially harmful and discriminatory outputs. The improved accessibility of our method also raises concerns about potential misuse for generating misinformation or hate speech. To address these negative impacts, robust bias mitigation techniques, fairness evaluation frameworks for quantized models, rigorous testing, monitoring, and ethical governance frameworks must be developed and implemented. Continued research efforts are needed to realize the efficiency benefits while proactively addressing the associated risks and potential harms.