Gradient Analysis of Large Machine Learning Models for Efficiency

Siddhant Mohan[*]
New York University
Brooklyn, NY 11201
sm12766@nyu.edu

Jay Daftari[*]
New York University
Brooklyn, NY 11201
jd5829@nyu.edu

Athul Radhakrishnan[*]
New York University
Brooklyn, NY 11201
ar6316@nyu.edu

Abstract

In this project, we conduct a statistical analysis of gradients in Vision Transformer (ViT) models to understand their underlying distributional behavior. By comparing the empirical gradient distributions across multiple layers with well-known parametric distributions, we identify the Laplacian distribution as the best fit using Quantile-Quantile (Q-Q) plots as the primary tool for evaluation. Utilizing the statistical properties of the fitted Laplacian distribution, we develop quantization and pruning techniques for gradients during training. These compression techniques are critical for improving training efficiency, especially in resource-constrained environments, as they reduce memory footprint and communication overhead in distributed systems without significantly affecting model performance. Our findings suggest that understanding the probabilistic nature of gradient distributions can inform principled gradient compression strategies, thereby enabling more efficient deployment of transformer-based architectures.

1 Introduction

Training deep neural networks is computationally intensive, with gradient calculation being a major component. While model compression research has largely focused on weights and activations, gradients remain challenging due to their dynamic behavior and sensitivity to model structure. In this work, we analyze gradient distributions in Vision Transformers (ViTs), both pre-trained and randomly initialized, and find via Q-Q plots that they consistently follow a Laplacian distribution—a pattern stable across layers and training steps. This distribution's sharp peak and heavy tails offer advantages for sparsity and quantization. Building on Chmiel et al. (5), we use the Laplacian scale parameter to guide pruning and quantization, showing that understanding gradient statistics can enable principled and effective compression in modern architectures.

2 Individual contributions

Siddhant Mohan Conducted the statistical analysis of Vision Transformer gradients. Extracted gradient data during training, compared them against various parametric probability distributions, and identified the Laplacian distribution as the best fit using Q-Q plots. Further analyzed the consistency

[*] All authors contributed equally.

Preprint. Under review.

of this distribution across layers, training epochs, and initialization conditions, and estimated the corresponding distribution parameters.

Jay Daftari Focused on gradient pruning informed by the Laplacian distribution. Developed a stochastic pruning scheme using analytically derived threshold values to achieve target sparsity levels. Evaluated the impact of pruning on model performance by measuring post-training accuracy and loss degradation.

Athul Radhakrishnan Implemented gradient quantization techniques tailored to the Laplacian distribution. Explored both linear and non-linear quantization strategies, leveraging the distribution's properties to design efficient encoding schemes, and assessed their influence on quantization error.

3 Problem description

A significant portion of high compute resources during training of large models arises from the storage and communication of gradients during backpropagation. While techniques for compressing weights and activations have been widely studied, gradients remain less explored due to their dynamic and statistically less predictable nature. This project aims to address this gap by analyzing the statistical distribution of gradients in ViTs and leveraging these insights to develop principled methods for gradient pruning and quantization. By identifying a suitable probabilistic model for gradients, we aim to reduce the training footprint without substantially affecting model performance.

4 Related work

Prior work has explored various strategies for gradient compression, including quantization, sparsification, and sketching techniques. Notable among these are TernGrad (1), QSGD (2), Deep Gradient Compression (3), and PowerSGD (4), which aim to reduce the communication cost with minimal degradation in convergence or accuracy. However, many of these methods are heuristic in nature and do not rely on a principled understanding of the statistical structure of gradients.

Our work builds upon recent insights introduced by Chmiel et al. (5), who analyzed the gradient distributions in BERT and conventional CNN architectures and found them to be near-lognormal. Their statistical perspective was leveraged to guide more effective quantization and sparsification techniques. While their study focused on traditional architectures, we extend this line of inquiry to Vision Transformers (ViTs)—a newer and significantly larger class of models that operate on fundamentally different principles, such as patch-based self-attention.

5 Estimating the Gradient Distribution

5.1 Model and Dataset

We use the Vision Transformer (ViT), a Transformer-based image classifier that processes 224×224 images as 16×16 patches via self-attention (7). Specifically, we use the `google/vit-base-patch16-224` model with 85.95M parameters and 12 encoder layers. Training is performed on CIFAR-10, a 10-class dataset of 32×32 images with 50,000 training and 10,000 test samples (8). Its compact size and visual diversity make it ideal for analyzing gradients in large models.

5.2 Statistical analysis

To analyze the statistical behavior of gradients in the Vision Transformer, we designed a two-pronged approach focused on both centered and heavy-tailed distributions. Specifically, we collected raw gradients for each layer during training over 10 iterations and averaged them to reduce noise. These gradients were then compared against three zero-mean parametric distributions: Normal, Laplace, and Logistic. In parallel, we also considered the absolute values of the gradients to assess compatibility with heavy-tailed distributions—namely Lognormal, Pareto, and Inverse Gamma. The theoretical distributions can be seen in 1.

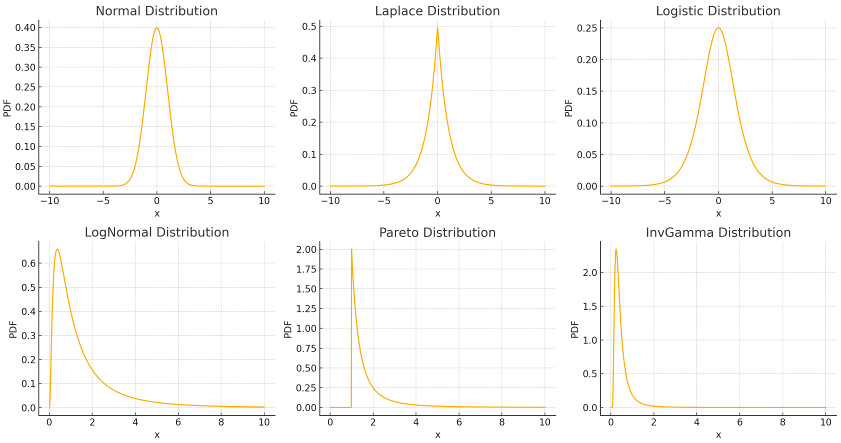


Figure 1: Theoretical distributions

For each comparison, we generated quantile-quantile (Q-Q) plots to visually evaluate how closely the empirical gradient distribution matched each theoretical model.

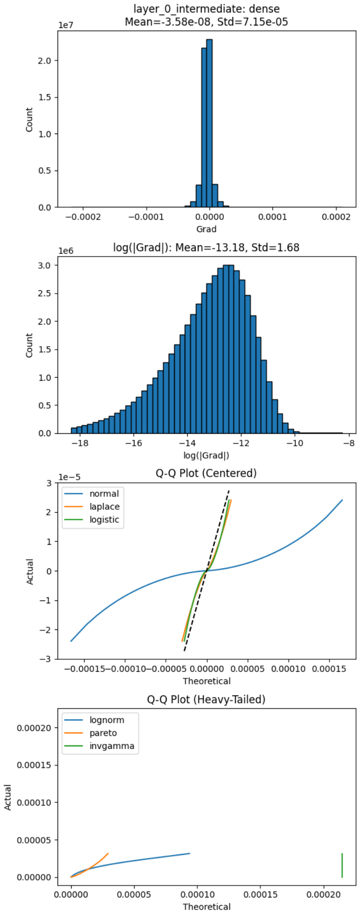


Figure 2: Layer 0 Intermediate

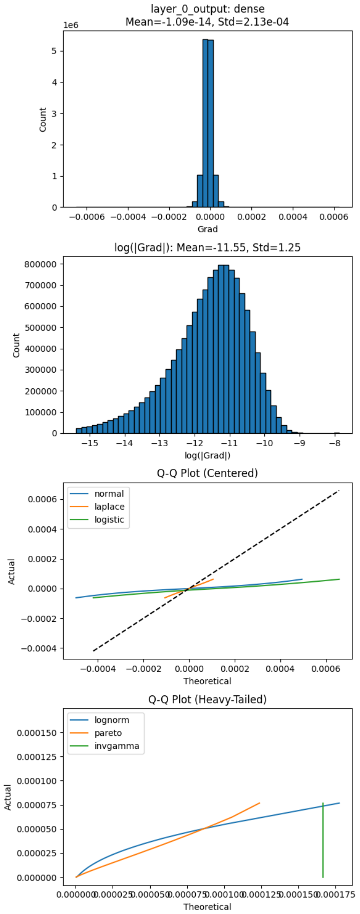


Figure 3: Layer 0 Output

Our analysis revealed that the Laplace distribution consistently provided the closest visual fit, as indicated by near-linear alignment in the Q-Q plots across all layers. These results are shown in

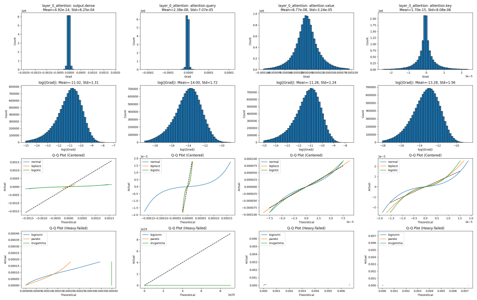


Figure 4: Layer 0 Attention

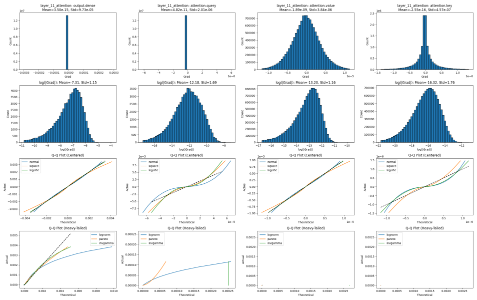


Figure 5: Layer 11 Attention

Figures 2-4 for the first and last attention blocks, as well as the intermediate and output layers of the first attention block, of a newly-initialized ViT.

We further examined layer-wise behavior and observed that while gradients in all layers followed a Laplacian shape, later layers exhibited narrower, more peaked distributions centered around zero. This reflects the fact that deeper layers typically require smaller weight updates. Notably, the Laplacian trend held for both pre-trained and randomly initialized models, and persisted over the course of training epochs. As training progressed, the gradients became increasingly sparse and spiky—suggesting that the model progressively stabilized and required fewer updates, consistent with expected convergence behavior.

5.3 Laplacian distribution

The Laplacian distribution is a continuous probability distribution characterized by a sharp peak at the mean and heavier tails than the Gaussian distribution. It is defined by two parameters: the location parameter μ, which determines the peak, and the scale parameter $b > 0$, which controls the spread of the distribution.

The probability density function (PDF) of the Laplacian distribution is given by:

$$f(x \mid \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$$

The Laplacian distribution provides a good fit due to its ability to model the sparsity and spikiness commonly observed in gradient values.

6 Pruning of gradients

6.1 Three-Way Stochastic Pruning Rule

After computing gradients g_i for each model parameter, we apply the following *stochastic three-way thresholding* map to each gradient element:

$$T_{\alpha,\varepsilon}(g) = \begin{cases} g & \text{if } |g| > \alpha \quad \text{(unchanged)} \\ \operatorname{sign}(g)\,\alpha & \text{if } \alpha\,\varepsilon \le |g| \le \alpha \quad \text{(saturated to } \pm\alpha) \\ 0 & \text{if } |g| < \alpha\,\varepsilon \quad \text{(pruned to 0)} \end{cases}$$

6.2 Expected Sparsity under a Laplace Model

Assume gradients $G \sim \text{Laplace}(0, b)$, i.e. $f(g) = \frac{1}{2b}e^{-|g|/b}$. The stochastic three-way map prunes g to zero iff $|g| < \alpha\varepsilon$, so the sparsity S can be defined as

$$S(\alpha) =_{G,\varepsilon} \big[\{\, |G| < \alpha\varepsilon\}\big] = \frac{1}{b}\int_0^{\alpha} \Big(1 - \frac{g}{\alpha}\Big) e^{-g/b}\, dg = 1 - \frac{b}{\alpha}\big(1 - e^{-\alpha/b}\big). \tag{1}$$

6.3 Closed-Form Inversion via Lambert–W

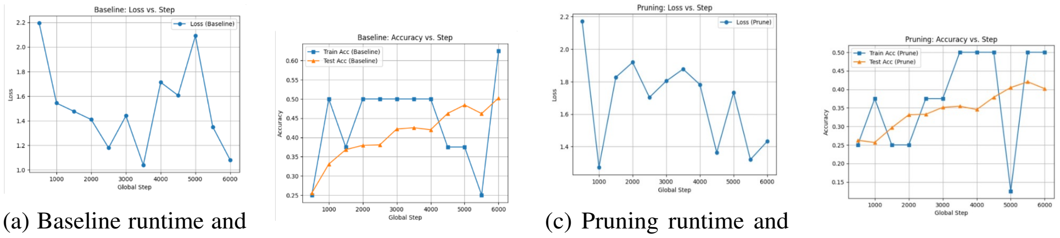
Let $\delta = 1 - S$ and $y = \alpha/b$. Then utilizing the principal branch Lambert-W function yields

$$\boxed{\alpha = b\Big[\tfrac{1}{\delta} + W\big(-\tfrac{1}{\delta}e^{-1/\delta}\big)\Big]}$$

6.4 Estimating the Scale b

We found out from empirical data analysis that the *raw* gradients G follow a zero-mean Laplace distribution with scale b: so, if $(g_i) \approx 0, \quad \hat{b} = \frac{1}{n}\sum_i |g_i|$.

6.5 Experiments and results



(a) Baseline runtime and loss

(b) Baseline accuracy

(c) Pruning runtime and loss

(d) Pruning accuracy

Figure 6: Performance comparison of baseline and pruning methods.

In the pruning experiment, we targeted 90% sparsity and computed the threshold for each layer

$$\alpha = b\Big[\frac{1}{1-S} + W\Big(-\frac{1}{1-S}\,e^{-1/(1-S)}\Big)\Big] \;\approx\; 9.9995\,b$$

from our closed-form Lambert W derivation. This threshold was used to sparsify the gradients.

During a one-epoch run, the training cross-entropy loss and the train and test accuracies of the baseline ViT model (trained from scratch) and the 90% sparsified-gradient ViT model is shown in Figure 7. As expected, the training loss steadily decreases and both training and testing accuracy improve with more optimization steps. However, the model with pruning takes longer to converge to a lower loss and high accuracy, as expected since 90% of the gradients in each iteration are not updated.

It is observed that loss with pruning is only slightly higher than the baseline, despite freezing 90% of the gradients, and accuracy with pruning is modestly lower than the baseline (Figure-4), reflecting the performance cost of aggressive sparsification.

The pruning run took 9222 seconds, over four times longer than the baseline. This slowdown arises from fitting the Laplace scale parameter b, solving for α via the Lambert W function, and applying the stochastic three-way rule for all layers before activation.

7 Laplace-Companding Quantization

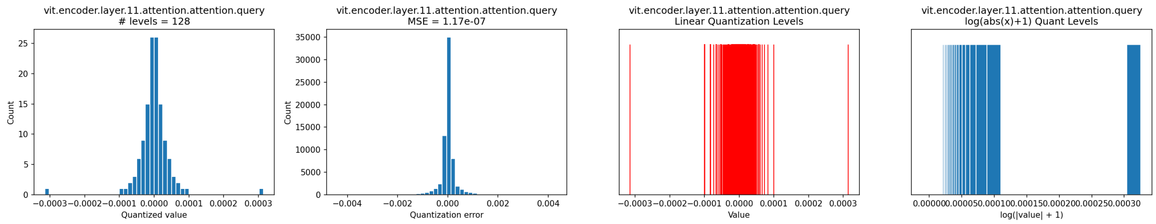


Figure 7: Quantized values and quantization levels

Assume entries of x follow a Laplace distribution:

$$p(x) = \frac{1}{2b} \exp\bigl(-|x-\mu|/b\bigr).$$

The optimal scalar quantizer is obtained by companding via the CDF [2]. Denote

$$F(x) = \begin{cases} \frac{1}{2}\exp\bigl(\frac{x-\mu}{b}\bigr), & x < \mu, \\ 1 - \frac{1}{2}\exp\bigl(-\frac{x-\mu}{b}\bigr), & x \geq \mu, \end{cases}$$

and its inverse:

$$F^{-1}(u) = \begin{cases} \mu + b\ln(2u), & u < \frac{1}{2}, \\ \mu - b\ln\bigl(2(1-u)\bigr), & u \geq \frac{1}{2}. \end{cases}$$

The companding quantizer proceeds in three stages:

1. **Fit** μ, b to $\{x_i\}$ via MLE:

$$\hat{\mu} = \mathrm{median}(x_i), \quad \hat{b} = \tfrac{1}{n}\sum_i |x_i - \hat{\mu}|.$$

2. **Compand** each x_i:

$$u_i = F(x_i; \hat{\mu}, \hat{b}) \in (0,1).$$

3. **Quantize** u_i uniformly to L levels:

$$q_i = \frac{u_i(L-1)}{L-1}, \quad \tilde{u}_i = \frac{q_i}{L-1}.$$

4. **Expand** via inverse CDF:

$$\widehat{x}_i = F^{-1}(\tilde{u}_i; \hat{\mu}, \hat{b}).$$

One can show this attains the rate-distortion bound asymptotically in L [2].

Algorithm 1 Laplace-Companding Quantization

```
Require: $x = [x_1, \dots, x_n]$, $L = 2^b$
Ensure: Quantized tensor $\widehat{x}$
0: $\hat{\mu} \leftarrow \mathrm{median}(x)$
0: $\hat{b} \leftarrow \frac{1}{n}\sum_{i=1}^{n} |x_i - \hat{\mu}|$
0: for $i = 1$ to $n$ do
0:     $u_i \leftarrow F(x_i; \hat{\mu}, \hat{b})$ {CDF mapping}
0:     $u_i \leftarrow \mathrm{clamp}(u_i, \epsilon, 1-\epsilon)$ {Avoid boundary issues}
0:     $q_i \leftarrow \mathrm{round}(u_i \cdot (L-1))/(L-1)$ {Uniform probabilistic quantization}
0:     $\widehat{x}_i \leftarrow F^{-1}(q_i; \hat{\mu}, \hat{b})$ {Inverse CDF}
0: end for
0: return $\widehat{x}$ =0
```

7.1 Experiments and results

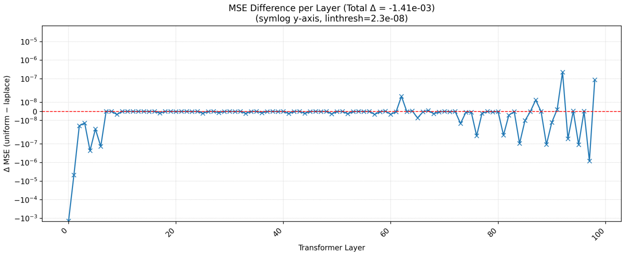


Figure 8: MSE between linear and Laplacian quantization

We conducted 7-bit gradient quantization experiments across different layers and evaluated the effectiveness of Laplacian-based quantization compared to uniform quantization (Figure 7). Our results show that Laplacian quantization achieves a low mean squared quantization error (MSE) of 1.2×10^{-7}, indicating an accurate representation of the gradient distribution. To further investigate this, we plotted the difference in MSE between linear and Laplacian quantization across all layers (Figure 8). While the MSE gap remains relatively small in the middle layers, we observe a noticeable improvement in both the initial and final layers when using Laplacian quantization. These results highlight the potential of statistically-informed quantization strategies for preserving training fidelity, particularly in layers with more dynamic gradient behavior.

8 Conclusion

This project shows that gradients in Vision Transformers follow a zero-mean Laplace distribution, which is maintained across layers and training epochs, and in both pre-trained as well as newly-initialized models. This is utilized for a principled three-way stochastic pruning rule with analytically derived thresholds via the Lambert–W function. Applying 90% sparsity yields only a slight increase in cross-entropy loss and a modest decrease in accuracy, demonstrating the effectiveness of our approach. The Laplacian distribution is also utilized for a non-linear quantization scheme that ensures minimal quantization error.

9 Future work

Building upon our work, the training and convergence of the pruned and quantized transformer models can be investigated, including parameter-efficient fine-tuning techniques such as low-rank adaptation (LoRA). Analyses of gradient distribution of transformers, as well as their pruning and quantization on larger and more complex datasets like COCO or ImageNet would set a benchmark for gradient based efficieny techniques on large transformer models.

References

[1] Wen, W., Xu, C., Yan, F., Wu, C., Wang, Y., Chen, Y., & Li, H. (2017). TernGrad: Ternary gradients to reduce communication in distributed deep learning. *Advances in Neural Information Processing Systems*, 30.

[2] Alistarh, D., Grubic, D., Li, J., Tomioka, R., & Vojnovic, M. (2017). QSGD: Communication-efficient SGD via gradient quantization and encoding. *Advances in Neural Information Processing Systems*, 30.

[3] Lin, Y., Han, S., Mao, H., Wang, Y., & Dally, W. J. (2018). Deep Gradient Compression: Reducing the communication bandwidth for distributed training. *International Conference on Learning Representations (ICLR).*

[4] Vogels, T., Karimireddy, S. P., & Jaggi, M. (2019). PowerSGD: Practical low-rank gradient compression for distributed optimization. *Advances in Neural Information Processing Systems*, 32.

[5] Chmiel, B., Ben-Uri, L., Shkolnik, M., Hoffer, E., Banner, R., & Soudry, D. (2021). Neural Gradients are Near-Lognormal: Improved Quantized and Sparse Training. *9th International Conference on Learning Representations (ICLR).*

[6] Aji, A. F., & Heafield, K. (2017). Sparse communication for distributed gradient descent. *Proceedings of EMNLP*, 440–445.

[7] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*. Available at: `https://arxiv.org/abs/2010.11929`

[8] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical Report, University of Toronto.