Does Quantization impact Mathematical Reasoning in Language Models?

Vibha Masti [*1] **Aryan Singhal** [*1] **Jaydev Jangiti** [*1] **Lindia Tjuatja** [1]

Abstract

Rapid growth in language modeling and machine learning systems has substantially increased model size and complexity. This has led to a rise in computational demands, making inference costly in resource-limited environments. Model compression techniques, such as quantization, pruning, distillation, offer solutions to mitigate these costs, but can often degrade model performance on specialized tasks such as mathematical reasoning, code generation, etc. This paper investigates the impact of quantization on mathematical reasoning capabilities, focusing on how feature representations change in quantized versus non-quantized models. Using sparse autoencoders (SAEs), we explore the feature representations of both quantized and non-quantized models to identify which reasoning-related features are lost in the quantization process. Through our analysis, our goal is to understand feature retention and degradation during quantization, extracting insights into optimizing models for reasoning-intensive tasks without substantial performance loss. Our findings contribute to developing compression strategies that maintain task-specific abilities, enhancing the deployment of large language models in constrained settings. We plan to release our experimentation code if accepted.

1. Introduction

Language models are becoming increasingly larger and more complex, with both parameter count and general understanding ability expanding rapidly. Due to their exponential increase in model size, model inference has become computationally expensive, especially for deployment in resource-constrained environments. The latest generation of models contain hundreds of billions of parameters and are capable of performing a wide range of sophisticated tasks, including question answering, code generation, and multi-step reasoning. This scale, while enabling unprecedented generalization and emergent capabilities, comes at the cost of significantly increased computational and memory demands. As a result, running inference on these large models has become prohibitively expensive for many real-world applications, particularly in settings with limited hardware resources such as mobile devices, edge computing platforms, or cost-sensitive enterprise environments. To address these challenges, the research community has turned to model compression techniques that aim to reduce the size and complexity of neural networks while preserving most of their performance. Among the most widely used techniques are quantization ((Jacob et al., 2018; Li et al., 2016; Hubara et al., 2016; Rastegari et al., 2016; Han et al., 2015)), which reduces the precision of model weights and activations; pruning ((LeCun et al., 1989)) which removes redundant parameters or connections from the network, and distillation ((Hinton, 2015)), which trains a smaller model to mimic the behavior of a larger one. These approaches have proven to be effective for speeding up inference, reducing memory footprint, and enabling model deployment in production environments where latency and efficiency are critical.

However, the benefits of compression often come with trade-offs. Compressed models, particularly those that have undergone aggressive quantization or pruning, can suffer from degraded performance on specific benchmarks or task domains. For example, recent work has shown that model compression can impair multilingual understanding ((Marchisio et al., 2024)) and mathematical reasoning capabilities ((Jin et al., 2024; Feng et al., 2024)), both of which require high-fidelity internal representations and precise manipulation of semantic and symbolic information. These failures are not always captured by aggregate metrics like perplexity or average accuracy, highlighting the need for deeper analysis into how compression affects model internals.

In this paper, we focus on quantization and its effects on mathematical reasoning abilities from an interpretability perspective. Rather than evaluating performance solely based on input-output behavior, we aim to understand how quantization reshapes the internal feature representations that support reasoning. To this end, we employ Sparse Autoencoders (SAEs), a tool that has recently gained attention for its ability to disentangle and analyze meaningful directions

[*]Equal contribution [1]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA. Correspondence to: Vibha Masti <vmasti@alumni.cmu.edu>.

Pre-print, Copyright 2025 by the author(s).

in the hidden states of large language models. By applying SAEs to both quantized and non-quantized versions of a model, we can directly compare which internal features are preserved, altered, or lost during the quantization process. Through this approach, we seek to provide more granular insights into the impact of quantization on model reasoning and to identify avenues for designing compression techniques that are better suited for reasoning-intensive tasks.

2. Related Work

Transformers (Vaswani, 2017) have been widely used in language and vision related tasks now, and they have long been regarded as black-box learners. Linear probing techniques (Liu et al., 2019; Tenney et al., 2019) are one way to get an understanding of model representations by training an auxiliary linear model on hidden representations for a downstream task. Later work in language model interpretability have leveraged dictionary learning as a method to shed light on the hidden structures within neural networks, particularly focusing on isolating interpretable features and reducing the complexity caused by polysemantic neurons. Yun et al. (Yun et al., 2021) used non-negative sparse coding optimization for dictionary learning to visualize the embeddings of all the layers of a pre-trained BERT (Devlin, 2018) model. However, standard dictionary learning techniques were shown to have issues with overfitting. (Bricken et al., 2023)

Sparse autoencoders (SAEs) are weak dictionary learners, and have emerged as a crucial player in the field of mechanistic interpretability. By enforcing sparsity, SAEs make it possible to disentangle overlapping activations, allowing researchers to pinpoint specific features associated with distinct tasks or behaviors. (Cunningham et al., 2023) used SAEs to reconstruct internal activations of Pythia-70M and learn a set of sparsely activating features. They validated their approach by comparing the interpretability of these features to those obtained through PCA and ICA. This work paves the way for applying SAEs to other model layers, such as MLPs and attention heads, enhancing interpretability and potentially preserving reasoning capabilities in quantized language models. (Bricken et al., 2023) demonstrated how SAEs can model specific features in a one-layer transformer and provided comprehensive visualizations to support their analyses.

Gao et al. (2024) introduced a k-sparse autoencoder approach to enhance feature extraction in language models. By using a TopK activation function to control sparsity, they achieve superior reconstruction and minimal dead latents. Their work establishes scaling laws linking autoencoder size and sparsity, along with new metrics for feature quality evaluation. Further work (Lan et al., 2024) explored feature universality across large language models (LLMs) where the authors found that mid-layers in both Pythia-70m and Pythia-160m (Biderman et al., 2023) models shared high similarity, suggesting common feature structures. Braun et al. (2024) propose to train SAEs end-to-end, minimizing KL-divergence between the output distributions of the model and the model with reconstructed activations replaced.

Math-related abilities in LLMs have also been studied (Stolfo et al., 2023; Zhang et al., 2025), with attempts to interpret which features/parts of the model contribute the most to mathematical reasoning. The effects of quantization (Hubara et al., 2018) on feature interpretability (Kerkouri et al., 2024) has also been studied to uncover more robust analyses of performance retention and degradation.

Some work in mathematical reasoning analysis highlighted that low-precision, such as int8, face significant limitations, requiring large model sizes to handle these tasks effectively (Feng et al., 2024). Extensive studies leveraging the MathQA dataset have been conducted to investigate the impact of quantization on mathematical reasoning (Liu et al., 2024). They observe that quantizing LLaMA2-7B to 3-4 bits leads to a moderate reduction in performance compared to full-precision models, highlighting that mathematical reasoning tasks are particularly sensitive to lower bit-width quantization.

In contrast to prior studies that emphasize significant degradation in mathematical reasoning and challenging tasks due to quantization, Lee et al. (2024) find that task difficulty does not consistently exacerbate accuracy loss. Their evaluation of instruction-tuned models up to 405B across 13 benchmarks shows that even on inherently hard datasets like Math-Lvl-5, quantized models perform comparably to their full-precision counterparts. Moreover, models like Llama-3.1-70B (Dubey et al., 2024) maintain high accuracy on less demanding tasks (e.g., 88 on GSM8K). Their work highlights that quantization effects are nuanced, being more dependent on the method and model size than on task complexity alone.

Building on this nuanced understanding of quantization's effects, Zhao et al. (2024) investigate specific challenges posed by multi-step reasoning tasks, particularly in mathematical domains. They find that traditional low-precision quantization (e.g., W4A4) significantly degrades performance on datasets like MATH and HumanEval, with accuracy declines of 36.21% and 51.11%, respectively. To mitigate such losses, they introduced QSpec, a hybrid quantization approach that combines low-precision drafting with high-precision verification.

3. Methods

We conduct a comparative study of the effects of quantization on specific task-related abilities of models through a mechanistic interpretability-focused approach. Specifically, we study how model quantization affects the encoding of mathematical reasoning-related features. While quantization reduces the computational burden of large models, we hypothesize that it may selectively impact specific types of reasoning features, potentially affecting performance in tasks that require nuanced mathematical reasoning. We study the different reasoning features retained (and lost) during compression by training an SAE on the hidden features of both the unquantized and quantized models.

SAEs (Olshausen & Field, 1997) have recently seen a resurgence of popularity in the mechanistic interpretability space and have allowed for better understanding of underlying model representations (Ayonrinde et al., 2024; O'Neill et al., 2024). They are a class of autoencoders with a hidden layer dimension much greater than the input dimensions trained with a sparsity objective. Positioned at layers like the residual stream, MLP sublayers, or attention heads, SAEs learn a dictionary of feature directions that sparsely activate, effectively resolving polysemanticity—where single neurons respond to multiple unrelated features (Bricken et al., 2023). L1 regularization in SAEs promotes sparse activations by adding a penalty term to the loss function that discourages excessive neuron firing. This approach encourages the autoencoder to learn a more compact and meaningful representation of the input data, where only a small subset of neurons are activated for each input. We also experimented with alternate approaches using a TopK SAE (Gao et al., 2024) and discuss our analysis in the Section 5.5.

By mapping activation patterns to these sparse features, SAEs provide a clearer understanding of how LLMs process information, aiding mechanistic interpretability and improving transparency in their decision-making processes. In practice, sparse encoders have been shown to approximate the ground truth features of datasets. This is particularly effective in domains like image processing, where images can be decomposed into basic visual elements, and text analysis, where documents are often represented as sparse combinations of topics.

This capability makes them valuable for mechanistic interpretability, as they can decompose neural network activations into their component features, reducing polysemanticity and improving our understanding of how networks process information.

We evaluate mathematical reasoning on GSM8K (Cobbe et al., 2021) using a baseline (FP16) and a quantized (INT4, using bitsandbytes (bit)) language model, identifying failure cases unique to the quantized model. We train our Sparse Autoencoders (SAEs) on the hidden state activations from both models using a subset of the Dolma dataset (Soldaini et al., 2024). To make training more efficient, we cache the hidden state activations of the language models and then use that as a training dataset for the SAEs. We use a training objective that combines reconstruction MSE and L1 sparsity to disentangle the feature space and identify reasoning-specific features retained or lost during quantization. Through interpretability analysis of SAE activations, we examine individual neuron and token-specific activations, including L1 norms. Finally, we compare the sparsity and neuron activations between quantized and non-quantized models to assess quantization's impact on reasoning. The training procedure is shown in Figure 1.

4. Experimental Setup

We conducted experiments employing single-layer Sparse Autoencoders (SAEs) to analyze the internal representations within the Multi-Layer Perceptron (MLP) layers of a LLaMA 3.2 1B language model. Our investigation encompassed both the original, non-quantized (FP16 precision) activations and their quantized counterparts (INT4 precision). The hidden layer size of each SAE was configured to be an eightfold (8x) multiple of the dimensionality of the MLP activations. Consequently, given an MLP activation dimension of 2048, this resulted in a dictionary size for the SAE of 16384 (calculated as 2048×8). The architectural design of our SAEs incorporated both an encoder and a decoder, each comprising linear layers with learnable weight parameters. Following the encoder's linear transformation, a Rectified Linear Unit (ReLU) activation function was applied to introduce non-linearity into the encoded representations. The training objective for the SAEs was defined by a composite loss function that combined the Mean Squared Error (MSE) of the reconstruction and an L1 regularization term to promote sparsity. The specific form of this loss function is given by Equation 1:

$$\text{SAE\_loss}(a, \hat{a}, h) = \frac{1}{N}((1-\lambda) * \|a-\hat{a}\|_2^2 + \lambda * \|h\|_1) \quad (1)$$

where a represents the original MLP activation vector, $\hat{a}$ denotes the reconstructed MLP activation vector produced by the decoder, h is the sparse representation learned by the SAE's encoder (the dictionary representation), λ is the L1 sparsity coefficient, and N is the batch size. In our experiments, the sparsity coefficient (λ) was empirically set to 0.1. This particular loss formulation aimed to encourage sparsity in the encoded representations (h), effectively learning a dictionary of features, while simultaneously ensuring a high degree of accuracy in the reconstruction of the original input activations (a). To align with prior research

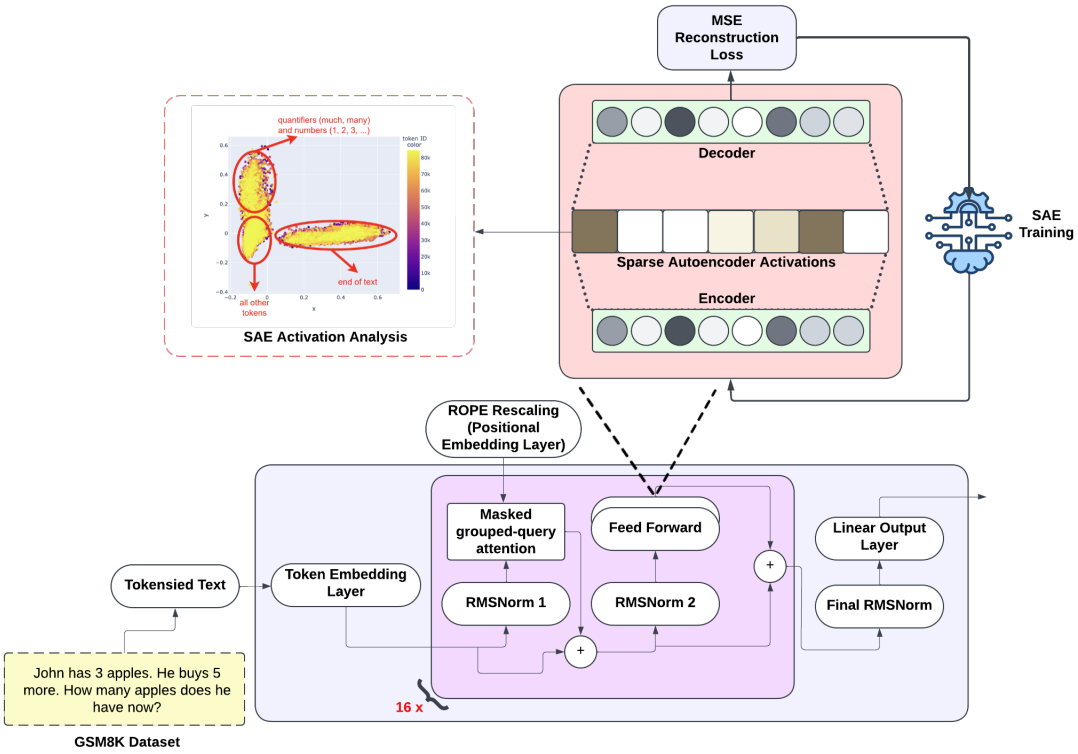


Figure 1. SAE Training Architecture

that has focused on the analysis of intermediate and later layers in transformer models ((He et al., 2024; Gao et al., 2024)), we strategically positioned our SAE for analysis at layer 12 of the LLaMA 3.2 1B model. This layer is situated approximately 3/4 of the way through the model's depth. Furthermore, we conducted supplementary experiments involving a TopK activation selection mechanism ((Gao et al., 2024)), and a detailed discussion of the findings from these ablation studies can be found in Section 5.5.

4.1. Datasets

For training the SAE, we used a subset of the Dolma dataset (Soldaini et al., 2024) containing 50 million tokens. To analyze the mathematical reasoning capabilities, the GSM8K dataset (Cobbe et al., 2021) was used in an 8-shot setting to evaluate the performance of quantized and non-quantized models. In the 8-shot setup, the model is provided with 8 example problems and their solutions before being asked to solve a new, unseen problem. This approach tests the model's ability to generalize from a limited set of examples. After running inference on both quantized models, the SAE (Sparsity-Aware Activation) activations were examined to better understand how the models processed and solved these math problems.

5. Results

5.1. Quantized vs. Non-Quantized Model Analysis

Our analysis of quantization effects on mathematical reasoning uses two configurations of the LLaMA-3.2B model: a full-precision FP16 variant and a quantized INT4 variant. Hidden layer activations were captured during inference on the GSM8K dataset to compare the two configurations. Both FP16 and INT4 model versions were evaluated with 8-shot prompting, providing eight example problems and solutions as context to guide reasoning during inference.

Table 1. Exact Match Accuracy for FP16 and INT4 Models on GSM8K Dataset (8-shot Prompting)

Model	Quantization	Exact Match
LLaMA 3.2-1B	FP16	40.51%
LLaMA 3.2-1B	INT4	39.56%

As per the results in Table 1 the FP16 model achieved an exact match accuracy of 40.51%, while the INT4 model scored slightly lower at 39.56%. This marginal difference suggests that while quantization introduces some loss in representational precision, it does not drastically impair the model's ability to perform arithmetic reasoning tasks. The

reduced accuracy in the INT4 model may be attributed to the compression process causing slight degradation in how task-relevant features are represented or retrieved during inference.

5.2. Activation Analysis

We performed an analysis to study the activations of the FP16 and INT4 models during inference to gain insights into how neurons respond to different tokens, particularly in mathematical reasoning tasks. These observations help in characterizing the models' representational behavior and assessing its capacity to encode reasoning-related features.

Figure 2 shows the token distribution based on non-zero activations in the FP16 and INT4 models. The x-axis denotes the number of non-zero activations per token, and y-axis represents the frequency of tokens with those counts. There exists a narrow peak at 0-199 activations per token for both models, indicating that there exist some tokens with a very high sparsity of features. This sparsity reflects the transformer's efficiency in selectively activating neurons to encode features. The distribution has its main peak for tokens with 1800-2000 activations and an overall sparsity of around 87%. A more detailed analysis is presented in Appendix (Table 2).

5.3. Token-based Visualizations

To gain insights into the behavior of hidden state activations, we examine the activations of the SAEs trained on both INT4 and FP16 models. This analysis is performed across all tokens in the GSM8K dataset. Using Principal Component Analysis (PCA), we reduce the dimensionality of the hidden states and visualize the token distributions in a scatterplot. The resulting clusters of tokens are depicted in Figure 3, highlighting distinct groupings.

Notably, for the INT4 model, the clusters corresponding to numerical tokens and other tokens are observed to be closer in proximity compared to their counterparts in the FP16 model. This suggests a subtle difference in representational separation between the two models. Despite this variation, the overall shapes of the distributions remain largely consistent across both models. This consistency indicates that the general representational capacity and token comprehension abilities of the models are comparable, aligning with expectations given their training configurations.

5.4. Neuron-based Visualizations

To investigate the relationship between individual neurons and specific abilities or token types, we analyzed the activation patterns of neurons for specific tokens. Specifically, we examined the frequency with which each neuron was activated by a given token across the LLaMA INT4 and FP16 models. This analysis revealed the presence of a primary neuron responsible for most numerical tokens in both model configurations.

Table 2. Comparison of Neuron Activations in INT4 and FP16 Models

Category	INT4	FP16
Dead Neurons (0 activations)	10597	7085
Percentage of Dead Neurons	64.68%	43.24%
Mean L1 norm of activations	94.39	114.67
Total sparsity of activations	89%	88%

Figure 4 illustrates the distribution of activations in the identified "numbers" neuron. The findings suggest a degree of specialization among neurons, particularly in the handling of numerical information, which appears to be consistent between the INT4 and FP16 precision settings. This observation provides evidence of the models' capacity to internally structure and encode semantic token types through distinct neuron activations.

5.5. Ablations with Top-k SAE

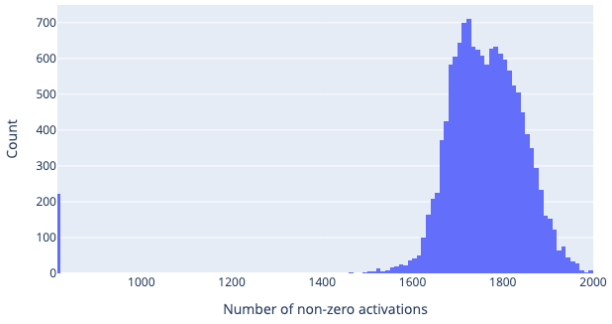
We experimented with an alternative to using L1 regularization, and only training the SAE with L0 loss, by using the Top-K activation function as described in (Gao et al., 2024). We trained a top-32 SAE on the same 50m tokens of Dolma and show our results in Table 3. We observe that while the number of dead neurons drastically decreased, the visualizations of tokens in the dictionary showed almost no meaning in the clusters formed (Figure 5), save for a large cluster for <eos> tokens. The mean L1 norm of activations was also greater by a factor of ~ 450. The distribution of number of activations is shown in Figure 6.

Table 3. Comparison of Neuron Activations in INT4 and FP16 Models - TopK SAE

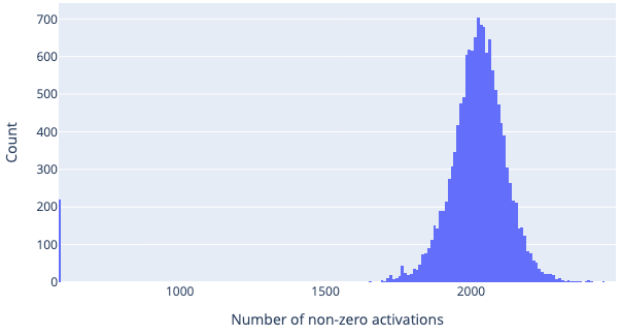
Category	INT4	FP16
Dead Neurons (0 activations)	90	62
Percentage of Dead Neurons	0.55%	0.38%
Mean L1 norm of activations	45787.48	43038.83
Total sparsity of activations	77%	76%

6. Conclusion

In this study, we investigated the activation patterns of SAEs trained on INT4 and FP16 quantized versions of the LLaMA 3.2 1B Instruct model, with a specific focus on the models' mathematical reasoning capabilities and the impact of quantization on performance. Our findings indicate that both quantization levels exhibit a comparable overall structural understanding of tokens, suggesting minimal degradation

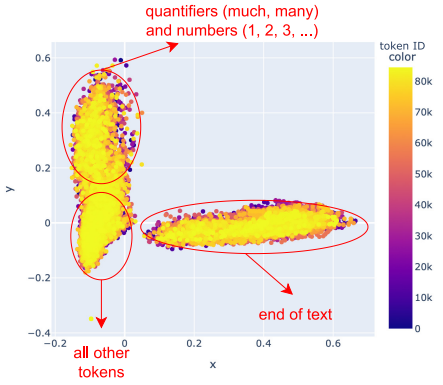


(a) Number of activations per token in INT4

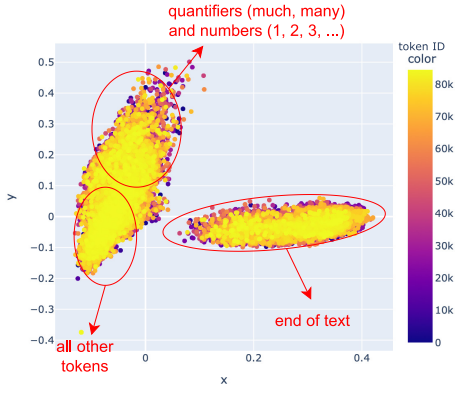


(b) Number of activations per token in FP16

Figure 2. Number of activations per token

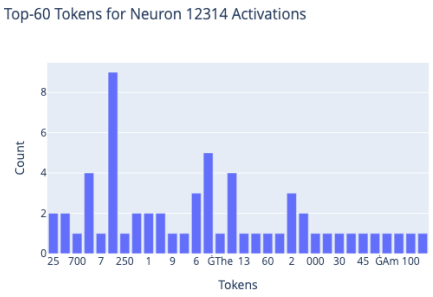


(a) Grouped activations for LLaMA FP16

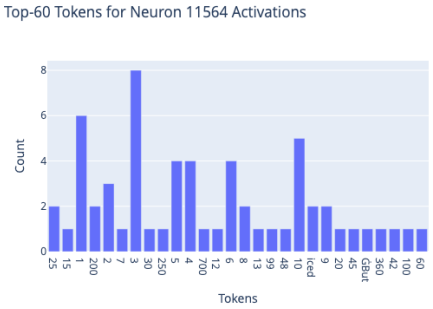


(b) Grouped activations for LLaMA INT4

Figure 3. PCA of hidden state activations for LLaMA models.



(a) FP16 model

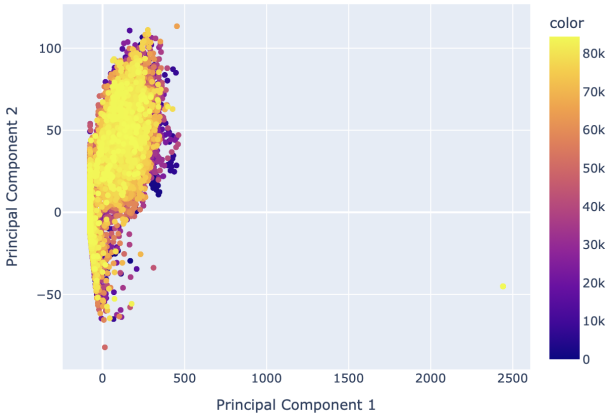


(b) INT4 model

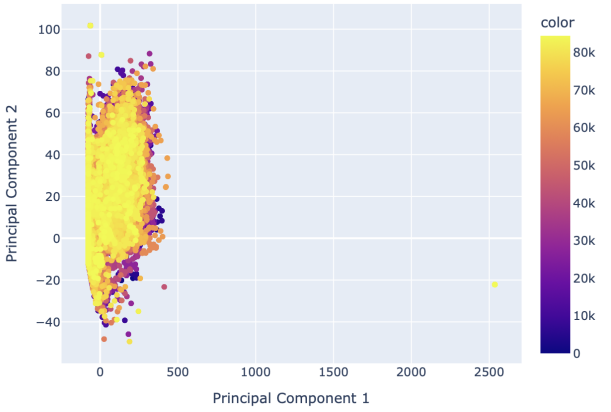
Figure 4. Neuron identified for numbers in FP16 and INT4 models

in high-level representations due to quantization. Furthermore, we observed that the SAE demonstrated significant sparsity and successfully identified monosemantic features associated with specific token subgroups, such as numerical tokens and quantifiers. Additionally, we conducted a detailed analysis of questions incorrectly answered by the INT4 model, comparing the activation patterns of both SAEs to gain insights into potential sources of error.

Future research could extend this work by training the SAE on a larger dataset, with billions of tokens, and by analyzing activations across different layers of the language model. Such investigations could further elucidate the effects of quantization and provide a deeper understanding of the internal representations learned by these models.

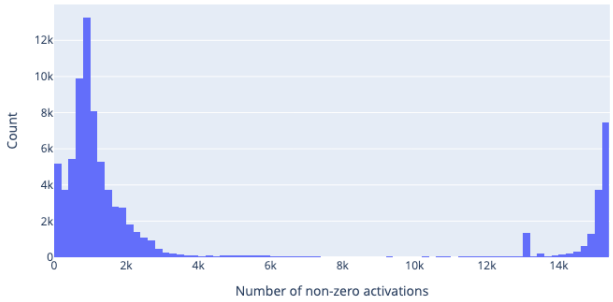


(a) INT4 model activations of GSM8K in Top-32 SAE

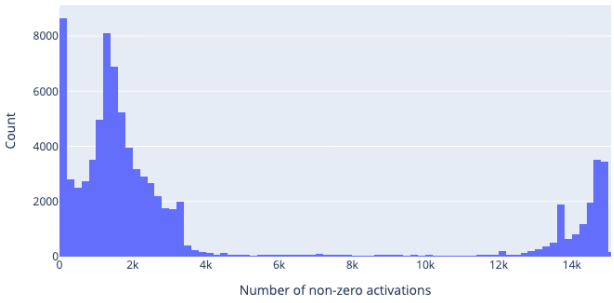


(b) FP16 model activations of GSM8K in Top-32 SAE

Figure 5. PCA visualizations of model activations for GSM8K questions in Top-32 SAE.



(a) Number of activations per token in INT4 - Top-32 SAE



(b) Number of activations per token in FP16 - Top-32 SAE

Figure 6. Number of activations per token - Top-32 SAE

References

bitsandbytes. https://huggingface.co/docs/bitsandbytes/v0.42.0/en/index. Accessed: 2024-11-12.

Ayonrinde, K., Pearce, M. T., and Sharkey, L. Interpretability as compression: Reconsidering SAE explanations of neural activations. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL https://openreview.net/forum?id=hAqeEZRVSD.

Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., Skowron, A., Sutawika, L., and van der Wal, O. Pythia: A suite for analyzing large language models across training and scaling, 2023. URL https://arxiv.org/abs/2304.01373.

Braun, D., Taylor, J., Goldowsky-Dill, N., and Sharkey, L. Identifying functionally important features with end-to-end sparse dictionary learning. *arXiv preprint arXiv:2405.12241*, 2024.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

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.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models, 2023. URL https://arxiv.org/abs/2309.08600.

Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Feng, G., Yang, K., Gu, Y., Ai, X., Luo, S., Sun, J., He, D., Li, Z., and Wang, L. How numerical precision affects mathematical reasoning capabilities of llms. *arXiv preprint arXiv:2410.13857*, 2024.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

He, Z., Shu, W., Ge, X., Chen, L., Wang, J., Zhou, Y., Liu, F., Guo, Q., Huang, X., Wu, Z., et al. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*, 2024.

Hinton, G. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. Binarized neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 4114–4122, 2016.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18(187):1–30, 2018.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2704–2713, 2018.

Jin, R., Du, J., Huang, W., Liu, W., Luan, J., Wang, B., and Xiong, D. A comprehensive evaluation of quantization strategies for large language models. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12186–12215, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.726. URL https://aclanthology.org/2024.findings-acl.726/.

Kerkouri, M. A., Tliba, M., Chetouani, A., and Bruno, A. Quantization effects on neural networks perception: How would quantization change the perceptual field of vision models? *arXiv preprint arXiv:2403.09939*, 2024.

Lan, M., Torr, P., Meek, A., Khakzar, A., Krueger, D., and Barez, F. Sparse autoencoders reveal universal feature spaces across large language models, 2024. URL https://arxiv.org/abs/2410.06981.

LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Lee, J., Park, S., Kwon, J., Oh, J., and Kwon, Y. A comprehensive evaluation of quantized instruction-tuned large language models: An experimental analysis up to 405b, 2024. URL https://arxiv.org/abs/2409.11055.

Li, F., Liu, B., Wang, X., Zhang, B., and Yan, J. Ternary weight networks. *arXiv preprint arXiv:1605.04711*, 2016.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. Linguistic knowledge and transferability of contextual representations. In *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)*, pp. 1073–1094, 2019.

Liu, Y., Meng, Y., Wu, F., Peng, S., Yao, H., Guan, C., Tang, C., Ma, X., Wang, Z., and Zhu, W. Evaluating the generalization ability of quantized llms: Benchmark, analysis, and toolbox, 2024. URL https://arxiv.org/abs/2406.12928.

Marchisio, K., Dash, S., Chen, H., Aumiller, D., Üstün, A., Hooker, S., and Ruder, S. How does quantization affect multilingual llms? *CoRR*, 2024.

Olshausen, B. A. and Field, D. J. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

O'Neill, C., Ye, C., Iyer, K. G., and Wu, J. F. Towards interpretable scientific foundation models: Sparse autoencoders for disentangling dense embeddings of scientific concepts. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*, 2024. URL https://openreview.net/forum?id=mPq3R6jdtD.

Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pp. 525–542. Springer, 2016.

Soldaini, L., Kinney, R., Bhagia, A., Schwenk, D., Atkinson, D., Authur, R., Bogin, B., Chandu, K., Dumas, J., Elazar, Y., et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

Stolfo, A., Belinkov, Y., and Sachan, M. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. *arXiv preprint arXiv:2305.15054*, 2023.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Van Durme, B., Bowman, S. R., Das, D., et al. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019.

Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Yun, Z., Chen, Y., Olshausen, B., and Lecun, Y. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 1–10, 2021.

Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.-W., Qiao, Y., et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2025.

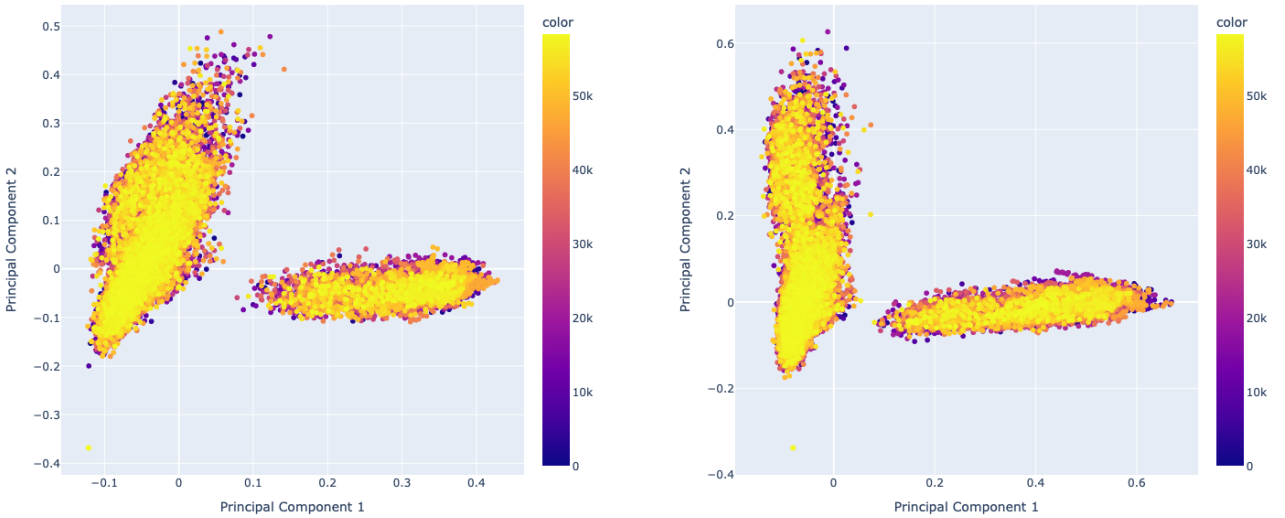
Zhao, J., Lu, W., Wang, S., Kong, L., and Wu, C. Qspec: Speculative decoding with complementary quantization schemes, 2024. URL https://arxiv.org/abs/2410.11305.

.1. Error Analysis

Building on the findings from the results, we conducted an analysis of the models' behavior by focusing on the activations for GSM8K questions that each model (quantized and non-quantized) answered incorrectly. By examining these subsets, we sought to identify patterns in token activations and neuron responses, aiming to better understand the underlying processes driving the models' decisions and the factors contributing to both correct and incorrect responses.

The INT4 model consistently gets questions wrong when they involve multiple operations or misinterpret the conditions described. These types of questions often require carefully following a step-by-step approach, as they involve various arithmetic calculations, such as combining quantities, applying rates, or adjusting for missing values.

Below figures represent PCA visualizations of the hidden state activations for questions that were consistently answered incorrectly by the INT4 model. The analysis compares the activation distributions of the quantized (INT4) and unquantized (FP16) versions of the LLaMA model to identify patterns that may underlie systematic errors.



(a) INT4 model activations for questions consistently wrong by INT4.

(b) FP16 model activations for questions consistently wrong by INT4.

Figure 7. PCA visualizations of model activations for questions consistently wrong by INT4.

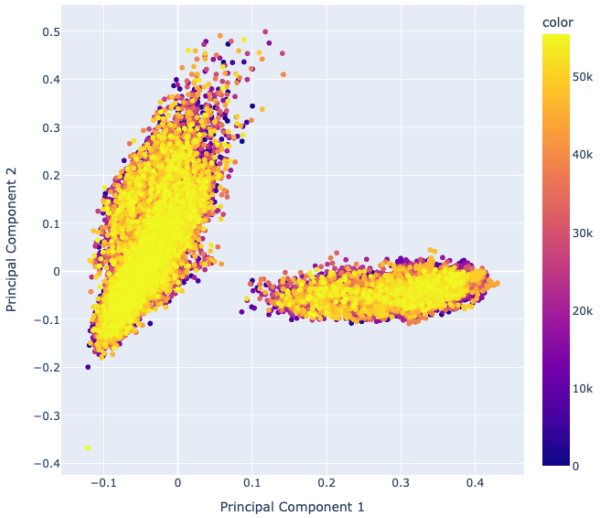
In Figure 7a, the PCA of the INT4 model activations reveals two distinct clusters: a densely populated cluster concentrated in the lower-left region and a sparse, elongated cluster extending along the first principal component. The elongated distribution, implies variability among certain tokens, possibly due to inconsistent feature encoding in the quantized model, leading to systematic errors. The cluster formation compared to the FP16 model seems to be quite poor indicating the quantized model's loss in ability to focus on specific relevant features.

In Figure 7b, the FP16 model activations show a tighter clustering pattern with fewer outliers. The dense cluster is more centralized and compact compared to the INT4 model, suggesting that the FP16 model maintains more consistent and refined feature representations. The absence of widely dispersed points indicates that the FP16 model's activations are more stable and well-separated for the error-inducing questions, allowing for better reasoning.

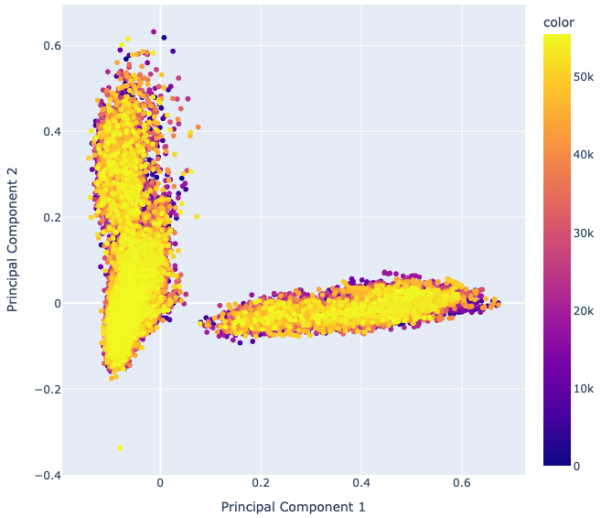
For questions where INT4 consistently fails, the PCA plots show a clear divergence: INT4 activations exhibit greater elongation and dispersion, while FP16 activations remain tightly clustered. This highlights quantization-induced instability.

In contrast, for FP16 wrong answers analyzed here, the activation patterns are more aligned, with both models showing a similar dense cluster and slight variability. Figures 8a and 8b show that the observed errors likely stem from intrinsic challenges in the questions themselves, where both models struggle to encode and separate features effectively. This indicates that the errors are not primarily caused by quantization but by limitations in the models' reasoning or representation

capabilities.



(a) INT4 model activations for questions consistently wrong by FP16.



(b) FP16 model activations for questions consistently wrong by FP16.

Figure 8. PCA visualizations of model activations for questions consistently wrong by FP16.

The comparison highlights that the INT4 model exhibits greater dispersion and inconsistency in its activations for the same set of error-prone questions. This suggests that quantization may introduce noise or reduce the model's ability to preserve fine-grained reasoning features, particularly for complex questions. The compactness of the FP16 model's activations further supports the hypothesis that unquantized models retain better representational fidelity, reducing systematic errors in reasoning tasks. This also indicates that while quantization impacts representational fidelity, some challenges in mathematical reasoning remain universal to both models, pointing to areas where model design or training methods can be further improved.