# OH! WE FREEZE: IMPROVING QUANTIZED KNOWLEDGE DISTILLATION VIA SIGNAL PROPAGATION ANALYSIS FOR LARGE LANGUAGE MODELS

**Kartikeya Bhardwaj***, **Nilesh Prasad Pandey***, **Sweta Priyadarshi**, **Kyunggeun Lee**, **Jun Ma**, **Harris Teague**
Qualcomm AI Research†
{kbhardwa,nileshpr,swetpriy,kyunggeu,jma,hteague}@qti.qualcomm.com

## ABSTRACT

Large generative models such as large language models (LLMs) and diffusion models have revolutionized the fields of NLP and computer vision respectively. However, their slow inference, high computation and memory requirement makes it challenging to deploy them on edge devices. In this study, we propose a light-weight quantization aware fine tuning technique using knowledge distillation (KD-QAT) to improve the performance of 4-bit weight quantized LLMs using commonly available datasets to realize a popular language use case, on device chat applications. To improve this paradigm of finetuning, as main contributions, we provide insights into stability of KD-QAT by empirically studying the gradient propagation during training to better understand the vulnerabilities of KD-QAT based approaches to low-bit quantization errors. Based on our insights, we propose *ov-freeze*, a simple technique to stabilize the KD-QAT process. Finally, we experiment with the popular 7B LLaMAv2-Chat model at 4-bit quantization level and demonstrate that ov-freeze results in near floating point precision performance, i.e., less than 0.7% loss of accuracy on Commonsense Reasoning benchmarks.

## 1 INTRODUCTION

The increasing popularity of large generative neural networks, LLaMA (Touvron et al. (2023), OPT Zhang et al. (2022)) and diffusion models (Dhariwal & Nichol (2021); Ho et al. (2020)), has revolutionized the field of machine learning exhibiting exceptional capabilities in generating realistic human-like text and images. However, due to large compute and memory requirements of these models, deploying these models on resource-constrained devices is challenging. Among various compression methods available, quantization (Nagel et al. (2021); Krishnamoorthi (2018); Jacob et al. (2018)) has proven to be a promising technique to optimize various vision and language models for deployment on resource-constrained devices.

In this paper, we focus on INT4 uniform, static, channelwise weight and tensorwise activation quantization of LLMs. Most quantization literature uses dynamic or blockwise methods for 4-bit or lower quantization to recover accuracy. However, such quantization schemes may not be supported on Neural Processing Units (NPU) hardware. Channelwise weight and tensorwise activation quantization is significantly more easy to support in terms of hardware and software implementation in commercial NPUs. Keeping such strong constraints in mind, when we perform INT4 quantization of weights, quantization noise increases significantly which risks destabilizing the entire training dynamics. Therefore, we need to analyze vulnerabilities in forward and backward passes within quantized LLMs. Hence, improving accuracy for uniform, static, channelwise weight and tensorwise activation quantization is extremely important for deployment use cases of LLMs.

To this end, we propose to *empirically* analyze the signal propagation through a well-known chat use case LLM, LLaMAv2-Chat, to understand its vulnerabilities to quantization errors. Specifically, we focus on multi-head self-attention modules of LLaMAv2-Chat (see Fig. 1) and conduct a detailed analysis of its forward and backward pass signals to see which components of the attention modules can cause KD-QAT to destabilize. After identifying these vulnerabilities, we propose specific

*Equal Contribution. †Qualcomm AI Research is an initiative of Qualcomm Technologies Inc.

solutions to stabilize the KD-QAT process to achieve significant improvements in INT4 quantized LLaMAv2-Chat network accuracy enabling its deployment on commercially available hardware.

We make the following **key contributions** in this paper:

1. We create a simple pipeline for KD-QAT for a well-known chat use case LLM, LLaMAv2-Chat-7B model. It uses public datasets and requires less than a day to provide INT4 quantized networks on a single node containing 8 NVIDIA A100 GPUs.
2. We analyze forward and backward pass in detail for this network and find that low-bit quantization non-uniformly impacts different parts of the attention modules. Specifically, o- and v-projection layers are more susceptible to quantization errors than other weight layers in the multi-head self-attention module.
3. Based on our analysis, we propose *ov-freeze* (oh! we freeze). It shows the importance of properly analyzing layers susceptible to low-bit quantization before embarking on quantization-aware-training. We demonstrate that this improves accuracy significantly across multiple benchmarks and closes the gap between quantized and FP models.

## 2 Related Work

Many methods have been proposed in the model efficiency space to solve the problem of neural network quantization. Quantization not only reduces model size but also leverages efficient fixed-point representation over floating-point representations. Most quantization techniques can be categorized either as post-training quantization (PTQ) (Nagel et al. (2020; 2019); Dong et al. (2019)) or quantization-aware-training (QAT)(Bhalgat et al. (2020); Esser et al. (2019); Nagel et al. (2022)) methods methods. Although, PTQ are go-to compression techniques (Frantar et al. (2023); Yao et al. (2022)) for most generative models, these methods suffer huge degradation in performance post quantization for low bitwidth quantization. QAT based approaches are difficult for these models due to lack of access to high quality data and training recipes required for finetuning these models.

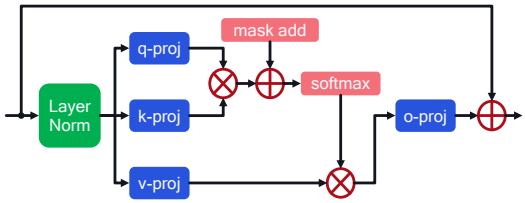


Figure 1: Multi-Head Self-Attention modules in LLaMAv2-Chat model. We focus on the forward and backward signal propagation properties after the q-, k-, v-, and o-projection layers to understand their vulnerabilities to low-bit quantization.

In this work, we propose to use knowledge distillation to finetune weight only quantized language models to achieve near floating point performance. LLM-QAT (Liu et al. (2023)) takes a similar approach but uses teacher generated dataset for finetuning, which is not always feasible due to sensitivity of generated dataset on sampling strategies and compute and memory intensive nature of the generation process. In this work, we use publicly available datasets for the distillation process, and perform ablations to understand vulnerabilities in the multi-head self-attention modules of the LLaMAv2-Chat model.

## 3 Method

In this section, we first describe our quantization and KD-QAT setup and analyze the gradient values of the LLaMAv2-Chat model to understand low-bit quantization vulnerabilities in the multi-head self-attention modules. Then, based on this analysis, we propose ov-freeze, a technique to stabilize our KD-QAT training pipeline.

### 3.1 Preliminaries

While quantizing neural networks, real valued weight $W^r$ and activations $x^r$ are quantized to low precision value. Hence, for a $b$ bitwidth uniform quantizer with scale $s$ and zero-point $z$, asymmetric quantization is defined as

$$q(W^r; s, z) = s \cdot \left[ clamp(\left\lfloor \frac{W^r}{s} \right\rceil + z, 0, 2^b - 1) - z \right] \tag{1}$$

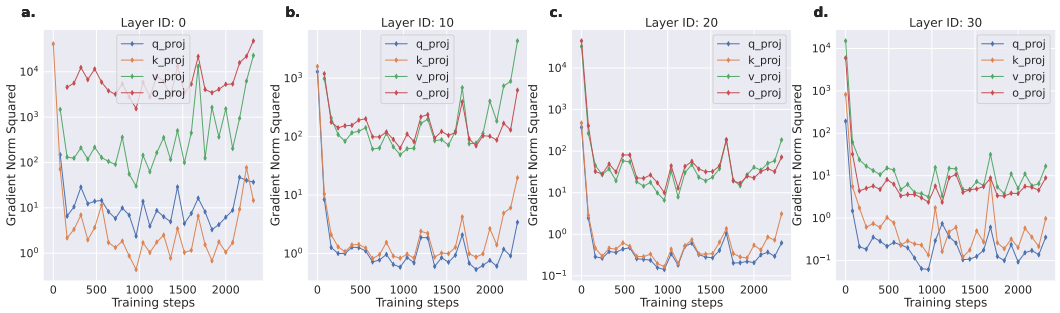


Figure 2: Relative comparison of gradient norm squared values between q-, k-, v-, o-projection outputs at various LLaMA self-attention modules during KD-QAT: (a) Layer 0, (b) Layer 10, (c) Layer 20, and (d) Layer 30. Clearly, o- and v- projection gradient norm squared values are one or two orders of magnitude higher than that of q- and k- projection layers. Such higher gradient values may destabilize o- and v-layers faster than q- and k-projections during KD-QAT, especially for low-bit quantization (e.g., INT4).

For our experiments, we use AIMET [1] (Siddegowda et al. (2022)) to quantize the models to desired bit-widths and use per-channel symmetric quantization for weights and asymmetric quantization for activations respectively. We set the quantization range of each weight quantizer using a mean squared error (MSE) based criteria minimizing the local MSE between FP and quantized tensors.

### 3.2 KD-QAT Setup

To demonstrate the effectiveness of our KD-QAT methodology on an established chat application, we experiment with the popular LLaMAv2-Chat model (Touvron et al. (2023)), and quantize it to W4A16[2]. For knowledge distillation based QAT, we use full-precision chat model as the teacher and quantized model as our student. Contrary to LLM-QAT (Liu et al. (2023)), we observe that using a weighted sum of cross-entropy and KL divergence for training leads to more stable and better convergence.

### 3.3 Gradient Properties of Quantized LLM

We investigate the gradient values of various components of the LLaMAv2-Chat multi-head self-attention modules (see Fig. 1) to understand which layers may contribute to destablization when we conduct INT4 weight quantization. For layer activations with output feature map, $\Theta$, and task loss $\mathcal{L}$, we compute the frobenius norm squared of the gradients at this layer's output:

$$||g||_2^2 = ||\nabla_\Theta \mathcal{L}(\Theta)||_2^2 \tag{2}$$

In Fig. 2, we show the comparisons among these gradient norm squared values for q-, k-, v-, and o-projection outputs at different hidden modules[3]. As evident, the gradient norm squared for o- and v-projections are one or two orders of magnitude higher than that for q- and k-layers. Moreover, as shown in Fig. 2 (a, b), the v-layers (green lines) in early hidden modules (Layer ID: 0 and 10) seem to have largest variation in gradient norm compared to other layers. Therefore, when we perform low-bit (e.g., INT4) quantization, these observations suggest that o- and v-layers are much more vulnerable to quantization errors than q- and k- layers. Specifically, results in Fig. 2 indicate that significantly higher gradient values may create bigger, more abrupt changes in o- and v- layers which would more directly impact their forward pass outputs. Therefore, forward pass of o- and v-projections may exhibit unstable behavior due to such high gradient values.

### 3.4 Oh! We Freeze

Since o- and v-projection layers can be senstive to sudden parameter changes due to high gradients, these layers can destabilize the entire KD-QAT training trajectory. Therefore, we propose *ov-freeze*

[1] AIMET is a product of Qualcomm Innovation Center, Inc., available on GitHub at `https://github.com/quic/aimet`

[2] WxAy quantization indicates quantizing all the weights and output activations to x and y bits respectively.

[3] Hidden modules are labeled as Layer IDs throughout this paper. For instance, "Layer ID: 10" refers to the 10th hidden module out of the total 32 modules containing multi-head self-attentions in LLaMAv2 model.

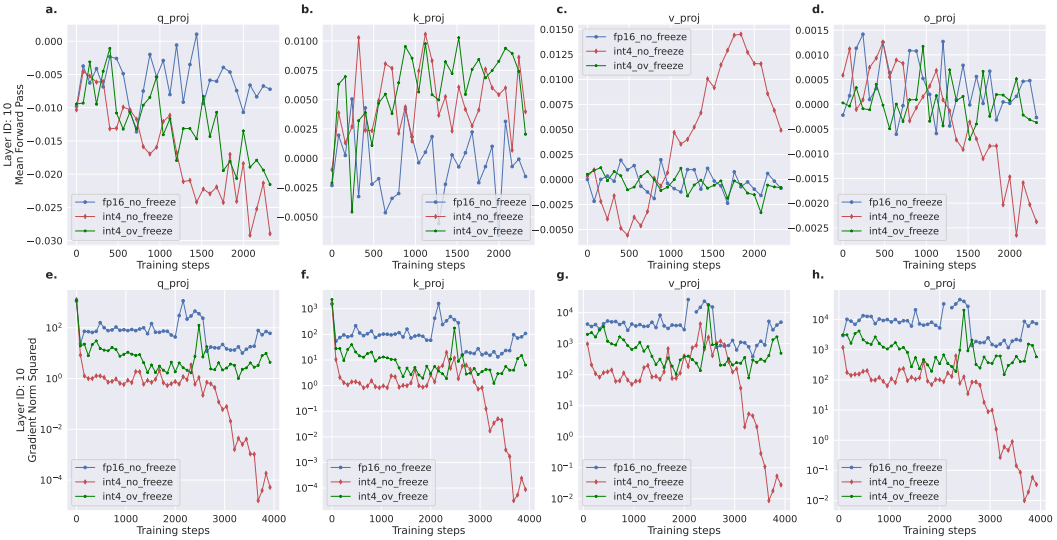


Figure 3: Forward and backward pass signal propagation analysis on a multi-head self-attention module at one of the hidden LLaMAV2-chat layers. We first analyze forward pass by tracking the mean value of the forward pass at the output of (a) q-projection, (b) k-projection, (c) v-projection, and (d) o-projection layers. Clearly, o and v layers are the most unstable as evident from the INT4 KD-QAT, no-freeze run. With ov-freeze, the forward pass behavior of the INT4 quantized network resembles that of the floating point model. Next, we analyze the backward pass behavior at the output of (e) q-projection, (f) k-projection, (g) v-projection, and (h) o-projection layers. Again, ov-freeze makes gradients more similar to the floating point training.

to stabilize the KD-QAT training process. As the name suggests, ov-freeze fixes the o- and v-projection weights to their post-training quantization values, freezes them and trains the rest of the network. This results in model adapting to quantization error without modifying the most sensitive parts of the network. Note that, v- and o-layers constitute a much smaller part of the complete LLM since majority of the parameters are contained within the MLP layers. In the next section, we will demonstrate how the proposed method significantly stabilizes the training loop and improves the final quantization results.

# 4 Experiments

In this section, we first describe the training setup and datasets. Then, we conduct extensive experiments to show that ov-freeze stabilizes both the KD-QAT training forward and backward passes. Finally, we present results demonstrating the superior W4A16 quantized accuracy obtained using ov-freeze on several benchmark tasks.

## 4.1 Training Setup and Datasets

To realize an important language use case, on device chat applications, we perform KD-QAT experiments on the popular 7B LLaMAv2-Chat. We used a 660B-token language dataset for the finetuning process. To evaluate the performance of the quantized networks, we use Wikitext and the CommonSense benchmarks - PIQA, Arc-Easy, Arc-Challenge, WinoGrande, OpenbookQA and Hellaswag.

## 4.2 Signal Propagation in LLaMAv2-Chat KD-QAT

### 4.2.1 Forward Pass Analysis

As the very first metric, we analyze the mean value of output activations of certain layers in the network. Specifically, Fig. 3 (a-d) show how the mean output activation value of q-, k-, v-, and o-projections vary across thousands of training steps for a specific LLaMAv2 hidden module. Here, we have used the FP16-no-freeze model (blue line) as a reference model. As evident from the INT4-no-freeze experiment (red line), o- and v-projection outputs deviate significantly from the FP16 reference. In contrast, when we freeze these two layers in all hidden modules (ov-freeze, green line), the mean forward pass follows the FP16 reference closely. Note that, even though we freeze only the o- and v-layers, the forward pass behavior still improves for other layers like the

q-projection output shown in Fig. 3(a). These instabilities in forward pass can easily destabilize the training trajectory during KD-QAT. Note that, although Fig. 3 shows results for hidden module (Layer ID) 10, we observed similar trends for other hidden modules as well (see Appendix A).

Next, we empirically explore the impact of ov-freeze on backward pass.

#### 4.2.2 Backward Pass Analysis

We now analyze the gradients at the outputs of q-, k-, v-, and o-projection layers at the same hidden module from the last section. We analyze gradient norm squared values as shown in Fig. 3 (e-h). Clearly, the INT4 ov-freeze (green line) again demonstrates similar gradient behavior as the reference FP16-no-freeze training run (blue line). In contrast, the INT4 no-freeze experiment (red line) has dramatically different backpropagation characteristics compared to the reference. All these instabilities result in lower final accuracy and an unstable KD-QAT.

| Model | Perplexity (ppl↓) |
|---|---|
| FP16 | 7.08 |
| PTQ: Min-Max range setting | 9.69 (+2.61) |
| PTQ: MSE-based range setting | 7.68 (+0.60) |
| KD-QAT: Baseline (no freeze) | 7.31 (+0.23) |
| KD-QAT: qkv-freeze (ours) | 7.12 (+0.04) |
| KD-QAT: v-freeze (ours) | 7.11 (+0.03) |
| KD-QAT: o-freeze (ours) | 7.09 (+0.01) |
| KD-QAT: oqkv-freeze (ours) | 6.99 (–0.09) |
| **KD-QAT: ov-freeze (ours)** | **6.98** (–0.10) |

Table 1: Wikitext Perplexity for FP16 and 4-bit weight only quantized LLaMAv2-Chat Model using context length 2048 obtained through various PTQ and KD-QAT schemes. Perplexity (↓): the lower the better. Red color denotes >0.2 point and Green denotes <0.2 point perplexity change compared to FP16.

### 4.3 Results

In this section, we compare performances of FP16 and quantized models obtained by our proposed freezing scheme on multiple benchmarks. First, Table 1 reports perplexity of PTQ and KD-QAT optimized 4-bit weight only quantized LLaMAv2-Chat on the Wikitext dataset. In our ablation, we consider different freezing schemes, o-, v-, ov-, qkv- and qkvo-, and observe freezing layers with relatively higher gradient norms values, i.e. o- and v-, along with other layers help significantly in improving both the stability of the KD-QAT training and the perplexity on Wikitext, achieving at-par or better performance than the FP model. As shown in Table 1, freezing ov- layers outperforms all other freezing schemes and PTQ baselines achieving better Wikitext perplexity than the FP16 model.

Finally, to enable on-device deployment of LLaMAv2-Chat model for practical uses, we evaluate our proposed ov-freeze W4A16 quantized model on multiple benchmarks. We use KD-QAT based weight quantized finetuned model, and quantize all activation outputs to INT16 using PTQ based min-max range setting to obtain the W4A16 model. As shown in Table 2, freezing ov-layers achieves competitive performance within 0.7% drop of FP16 model on average on CommonSense reasoning benchmarks, thereby improving significantly over both the best MSE based RTN and AdaRound baselines, both of which lose more than 2% accuracy compared to the FP16 network.

| Model | PIQA(↑) | Arc-Easy(↑) | Arc-Challenge(↑) | Winogrande(↑) | OpenbookQA(↑) | Hellaswag(↑) | Average(↑) |
|---|---|---|---|---|---|---|---|
| FP16 | 76.66 | 69.65 | 44.37 | 66.38 | 43.8 | 75.42 | 62.71 |
| | | | | PTQ | | | |
| MSE | 75.79 | 65.53 | 41.47 | 66.22 | 41.00 | 72.76 | 60.46 (–2.25%) |
| AdaRound | 74.59 | 66.62 | 40.7 | 66.54 | 40.60 | 72.28 | 60.22 (–2.49%) |
| | | | | KD-QAT | | | |
| no-freeze | 76.66 | 67.38 | 42.75 | **67.17** | 41.80 | 74.14 | 61.65 (–1.06%) |
| **ov-freeze** | **77.26** | **69.11** | **42.83** | 66.54 | **42.20** | **74.15** | **62.02** (–0.69%) |

Table 2: Evaluation of FP16 and various W4A16 Quantized LLaMAv2-Chat models using 1024 context length on Commonsense Reasoning benchmarks. (↑): the higher the better. Red color denotes >1% and Green denotes <1% loss of accuracy compared to FP16.

## 5 Conclusion

In this work, we proposed a light-weight quantization aware finetuning technique using knowledge distillation (KD-QAT) to improve the performance of popular W4A16 quantized LLM, LLaMAv2-Chat, for an important use case, on device chat applications, using commonly available datasets. We perform systematic study of the forward and backward pass of the LLaMAv2-Chat model, and analyze the output and gradient feature maps to hint towards extreme sensitivity of o- and v-layers in the multi-head attention modules. To improve the stability of the quantized finetuning, we proposed ov-freeze and experimented with popular 7B LLaMAv2-Chat model at 4-bit quantization level to show significant improvement in performance over vanilla QAT and achieve near float-point precision performance (within 0.7% drop of the FP16 model) on the CommonSense reasoning benchmarks.

## REFERENCES

Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 696–697, 2020.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 293–302, 2019.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 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.

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

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1325–1334, 2019.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pp. 7197–7206. PMLR, 2020.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. *arXiv preprint arXiv:2203.11086*, 2022.

Sangeetha Siddegowda, Marios Fournarakis, Markus Nagel, Tijmen Blankevoort, Chirag Patel, and Abhijit Khobare. Neural network quantization with ai model efficiency toolkit (aimet). *arXiv preprint arXiv:2201.08442*, 2022.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=f-fVCElZ-G1`.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.

## A Additional Results

**Forward and Backward Signal Propagation Results on Other LLaMAv2-Chat Modules.** Fig. 4, 5, and 6 show that our observations for hidden module 10 (i.e., Layer ID: 10 in Fig. 3) hold across other hidden modules. Clearly, the no-freeze mean forward pass for v-projection layer is unstable for all the hidden modules shown in Fig. 4, 5, and 6. The proposed method ov-freeze makes the forward and backward pass characteristics similar to those seen during floating point training.

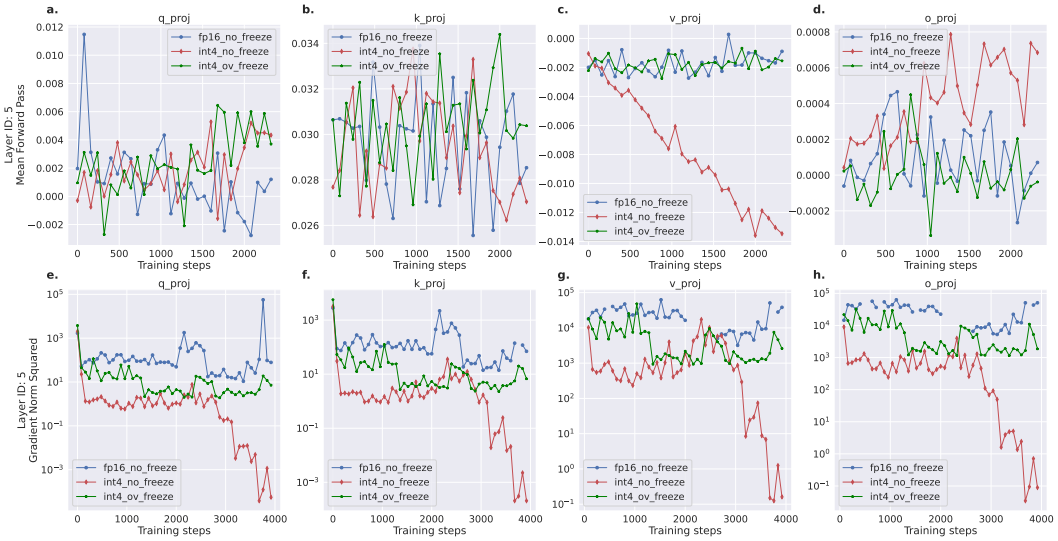


Figure 4: Layer ID: 5. Forward and backward pass signal propagation analysis on 5th hidden LLaMAV2-chat self-attention module. Our proposed ov-freeze makes the quantized model's forward pass and gradients more similar to those observed during FP16 training.

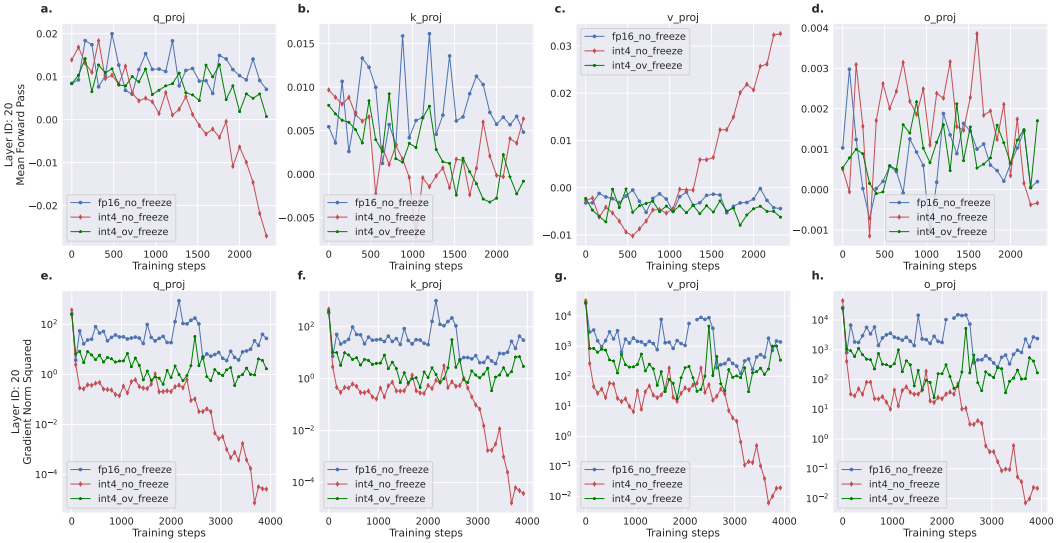


Figure 5: Layer ID: 20. Forward and backward pass signal propagation analysis on 20th hidden LLaMAV2-chat self-attention module. Our proposed ov-freeze makes the quantized model's forward pass and gradients more similar to those observed during FP16 training.

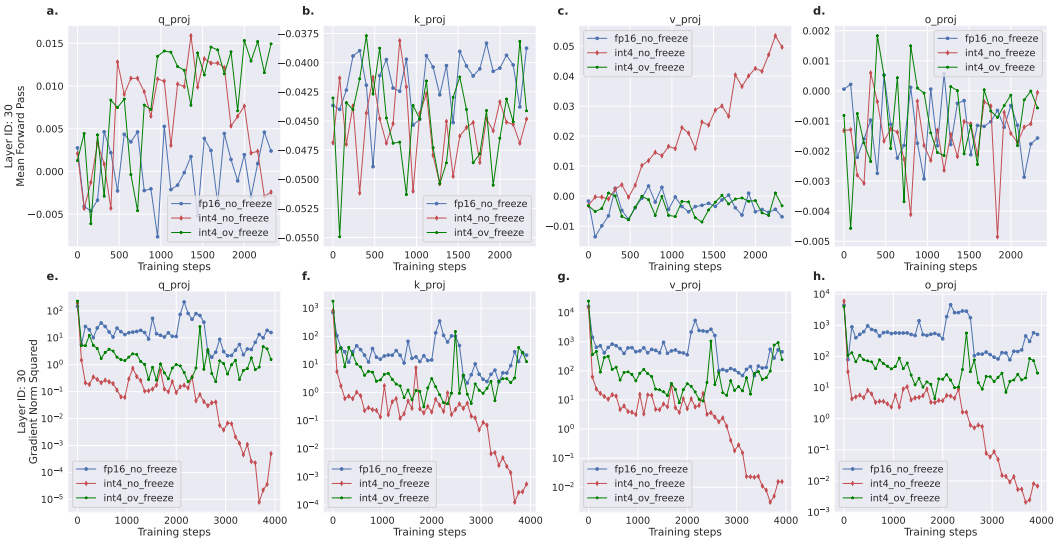


Figure 6: Layer ID: 30. Forward and backward pass signal propagation analysis on 30th hidden LLaMAV2-chat self-attention module. Our proposed ov-freeze makes the quantized model's forward pass and gradients more similar to those observed during FP16 training.