Scaling laws for post-training quantized large language models

Anonymous ACL submission

Abstract

Generalization abilities of well-trained large language models (LLMs) are known to scale predictably as a function of model size. In contrast to the existence of practical scaling laws governing pre-training, the quality of LLMs after post-training compression for efficient deployment remains highly unpredictable, often requiring case-by-case validation in practice. In this work, we attempted to close this gap for post-training weight quantization of LLMs, by conducting a systematic empirical study on multiple LLM families quantized to numerous low-precision tensor data types using popular weight quantization techniques. We identified key scaling factors pertaining to characteristics of the local loss landscape, based on which the performance of quantized LLMs can be reasonably well predicted by a statistical model.

1 Introduction

Large language models (LLMs) based on the transformer architecture (Vaswani et al., 2023) are known to obey empirical scaling laws. An LLM's generalization abilities, measured by the negative-log-likelihood (NLL) loss in next-token prediction, are predictably related to increases in parameter count, pre-training data volume, and computation cost (Kaplan et al., 2020; Dettmers and Zettlemoyer, 2023; Henighan et al., 2020; Alabdulmohsin et al., 2022; Su et al., 2024; Song et al., 2024; Muennighoff et al., 2023; Bordelon et al., 2024; Bahri et al., 2024).

Thanks to the guidance from scaling laws, pre-training of LLMs, a notoriously expensive computation in practice, enjoys a certain degree of confidence in return on investment. However, training is but half way toward model deployment in the real world. For these LLMs to run efficiently on a target accelerator for inference, they often undergo post-training compression, such as quantization (Gholami et al., 2021; Frantar et al., 2022; Park et al., 2024; Kim et al., 2023, 2024; Yao et al., 2022).

Post-training quantization (PTQ) is a process that attempts to preserve a trained LLM's generalizability, while performing its computation with low-precision data types. As such, PTQ, a process involving numerous extra factors, introduces significant additional uncertainty into the quality of the final model for deployment, in many cases completely obscuring the predictability prescribed by the pre-training scaling laws. This makes PTQ in today's practice a business of trial-and-error (Huang et al., 2024; Sharify et al., 2024; Yuan et al., 2023; Hu et al., 2022), lacking useful practical guidance from scaling laws like those governing model pre-training.

In this work, we attempted to close this gap in knowledge by systematically studying the empirical scaling of extra factors involved in PTQ, in addition to the pre-trained NLL loss. We briefly enumerate below all factors considered.

1. **Loss of pre-trained LLM.** This is a known scaling law that determines the quality of a trained LLM as the input to the PTQ procedure; intuitively, the better the trained model, the better its quantized version would be, everything else being equal. Section 2.1 is dedicated to it.
2. **Local loss landscape of pre-trained LLM.** Because quantization is a specific perturbation to the trained network, the resulting loss due to the perturbation depends not only on the converged NLL loss, but also on how steeply the loss changes in the neighborhood of convergence (Frumkin et al., 2023; Nahshan et al., 2020; Evci et al., 2020). Section 2.2 is dedicated to its scaling.
3. **Low-precision data type for quantization.** Numerous novel tensor data types for efficient inference have emerged recently (Rouhani et al., 2023; Dettmers et al., 2023; Agrawal et al., 2024; Guo et al., 2022); intuitively, both the tensor data type and its numerical precision would

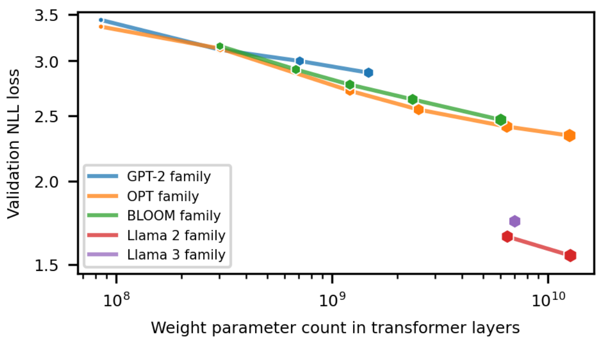

Figure 1: **Scaling of pre-trained NLL loss.** NLL losses evaluated on the validation split of the WikiText-2 dataset are plotted against the total parameter counts in the transformer layers' weight tensors. Model families are color-coded and the symbol sizes encode the weight parameter count, a convention shared by following figures.

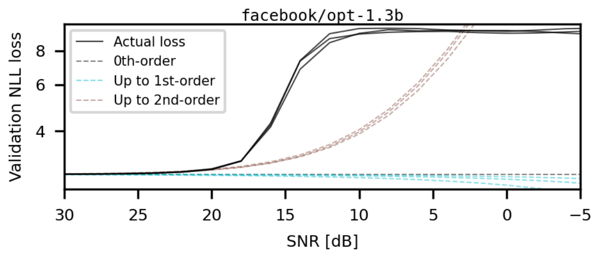

Figure 2: **Local radial loss landscape mapping.** Shown here is measurement of the *typical* loss landscape in the neighborhood of pre-trained weights, by evaluation of the loss along typical radial perturbations, 3 independent instances illustrated for `opt-1.3b`, together with their Taylor series approximations.

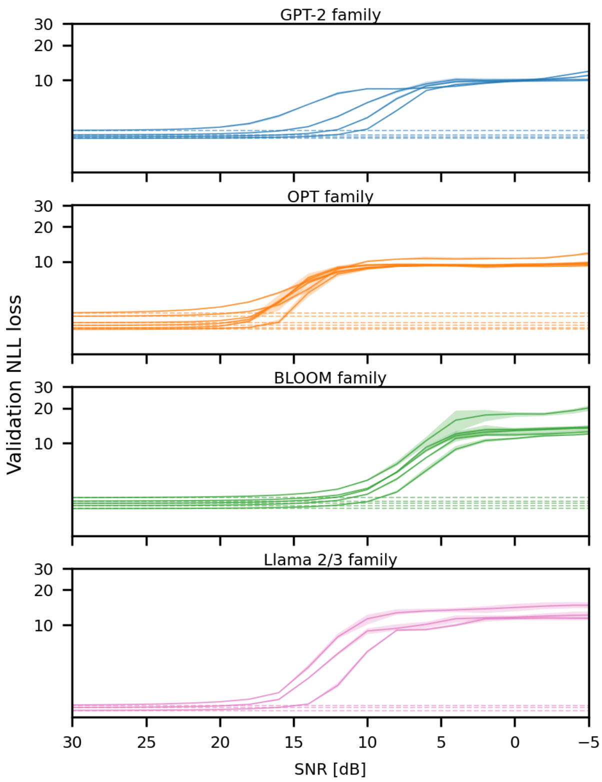

Figure 3: **Local loss landscape of LLMs grouped in families.** Shown are the mean (colored curves) and range (colored shades) of 3 independent measurements for each model. The typical characteristics are common to all models. Within a family, larger models tend to have flatter local loss landscape, in a predictable manner.

correlate with the quality of quantization. Section 2.3 is dedicated to its scaling.

4. **PTQ algorithm.** After aggressive low-precision quantization, certain PTQ optimization algorithms are commonly used to recover some model quality (Frantar et al., 2022; Xiao et al., 2024; Lin et al., 2024; Lee et al., 2024). These methods typically minimizes local quantization error as opposed to direct global loss optimization as in quantization-aware fine-tuning (*e.g.* Li et al. 2023; Jeon et al. 2024). Section 2.4 is dedicated to its scaling.

We show with concrete examples (for procedural details see Section 4), that all the above factors have underlying empirical scaling laws for certain LLM families. Incorporating these empirical rules, in Section 3, we build a predictive statistical model that takes the above factors as input and predicts the outcome of a PTQ procedure on unseen LLMs at a reasonable accuracy.

2 Factors subject to scaling for LLM PTQ

2.1 Loss of pre-trained LLM

First, we recapitulate one of the original scaling laws on well trained LLMs with no data limit (Kaplan et al., 2020).

We visualize in Figure 1 this scaling law with our experiments (see Section 4 for details). The GPT-2, OPT and BLOOM model families roughly follow one power law, whereas models in the Llama 2/3 family track a different, but qualitatively similar path.

2.2 Characteristics of local loss landscape

Next, we characterize another crucial factor intrinsic to the LLM itself, its local loss landscape.

A quantization of network weight $\boldsymbol{w}$ [1] can be

[1] Here we denote by vector $\boldsymbol{w}$ a flattened version of all weight matrices $(\boldsymbol{W}_1, \cdots, \boldsymbol{W}_L)$ of the network that are subject to quantization.

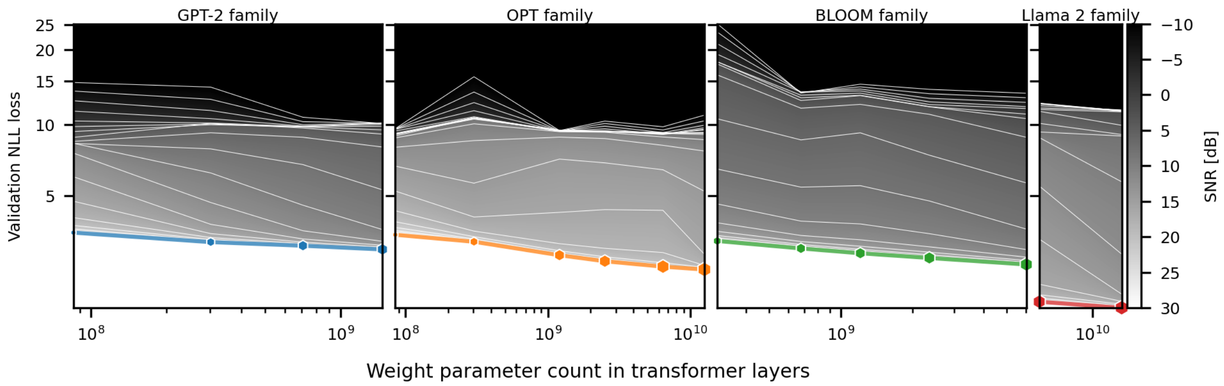

Figure 4: **Scaling of local loss landscape as a function of LLM size.** We plot NLL loss against weight parameter count, with typical perturbation SNR as a gray-scale heat map. Thin white iso-SNR curves are at 2 dB increments. With OPT family as the only exception, vertical spacing of these iso-SNR curves is shorter in large models than in small ones of the same family, suggesting flatter local minima at larger model sizes.

considered as a perturbation $\boldsymbol{w} \to \boldsymbol{w} + \Delta\boldsymbol{w} = Q(\boldsymbol{w})$, where Q is a quantizer, and the resulting loss of the quantized network becomes $\mathrm{NLL}(\boldsymbol{w} + \Delta\boldsymbol{w})$ from the pre-trained $\mathrm{NLL}(\boldsymbol{w})$. The resulting loss is a function not only of the pre-trained weight $\boldsymbol{w}$, but also of the perturbation $\Delta\boldsymbol{w}$, often approximated by Taylor expansion,

$$\begin{aligned}\mathrm{NLL}(\boldsymbol{w} + \Delta\boldsymbol{w}) = \mathrm{NLL}(\boldsymbol{w})& \\ + \boldsymbol{g}^\top \Delta\boldsymbol{w} + \frac{1}{2}\Delta\boldsymbol{w}^\top \boldsymbol{H} \Delta\boldsymbol{w}& \\ + O(\|\Delta\boldsymbol{w}\|^2).&\end{aligned}$$

Here $\boldsymbol{g}$ and $\boldsymbol{H}$ are the gradient and Hessian at $\boldsymbol{w}$, and $\|\cdot\|$ is the ℓ_2-norm.

As the absolute magnitude of $\boldsymbol{w}$ scales with dimensionality (see Appendix A), we use signal-to-noise ratio (SNR), a relative quantity to measure the magnitude of its perturbation $\Delta\boldsymbol{w}$,

$$\mathrm{SNR}(\boldsymbol{w}, \Delta\boldsymbol{w}) = 20 \log_{10} \frac{\|\boldsymbol{w}\|}{\|\Delta\boldsymbol{w}\|},$$

in decibel (dB). A higher SNR represents a smaller deviation $\Delta\boldsymbol{w}$ from $\boldsymbol{w}$. When the perturbation is due to quantization, *i.e.* $\Delta\boldsymbol{w} = Q(\boldsymbol{w}) - \boldsymbol{w}$, SNR becomes signal-to-quantization-noise ratio (SQNR),

$$\mathrm{SQNR}(\boldsymbol{w}) = 20 \log_{10} \frac{\|\boldsymbol{w}\|}{\|Q(\boldsymbol{w}) - \boldsymbol{w}\|}.$$

Intuitively, the flatter the local loss landscape is near $\boldsymbol{w}$, the less impact a same perturbation $\Delta\boldsymbol{w}$ is to exert on the loss. In Figure 2, we show with an example LLM, the *typical* local loss landscape in the neighborhood of pre-trained weights. We randomly sample a unit vector $\hat{e} \sim \mathcal{S}^D$ from the D-dimensional unit sphere, D being the dimensionality of $\boldsymbol{w}$, and measure $\mathrm{NLL}(\boldsymbol{w} + \lambda\hat{e})$ while sweeping $\lambda \in \mathbb{R}^+$. We see that the typical radial loss is very *step-like*: it stays relatively low and flat near $\boldsymbol{w}$, then rises rapidly (faster than quadratic), and finally plateaus further away from $\boldsymbol{w}$. These qualitative characteristics are shared by all LLMs of various sizes and from various families (Figure 3).

We also find that, within the same LLM family, larger models have flatter local loss landscape than smaller ones, in a systematic way (Figures 3, 4) for each family.

2.3 Low-precision data type for quantization

Now, we identify an extrinsic factor in PTQ process that is independent from the LLM itself, namely the low-precision tensor data type for quantization. Note that we consider tensorial data types, not simply scalar numerical formats. In addition to traditional integer quantization that requires calibration, emerging standards such as microscaling (MX, Rouhani et al. 2023) adopt more effective and efficient tensor data types, which we study in this work. we also present a comparative study of traditional integer quantization in Appendix B.

We first ask how the magnitude of quantization errors $\Delta\boldsymbol{w} = Q(\boldsymbol{w}) - \boldsymbol{w}$ vary across LLMs for certain data types. Despite the existence of significant scaling of $\|\boldsymbol{w}\|$ (see Appendix A for further details), the SQNRs are relatively invariant across model families and model sizes (Figure 5, left), and vary across numerical data types in a highly predictable manner. In contrast, NLL losses show

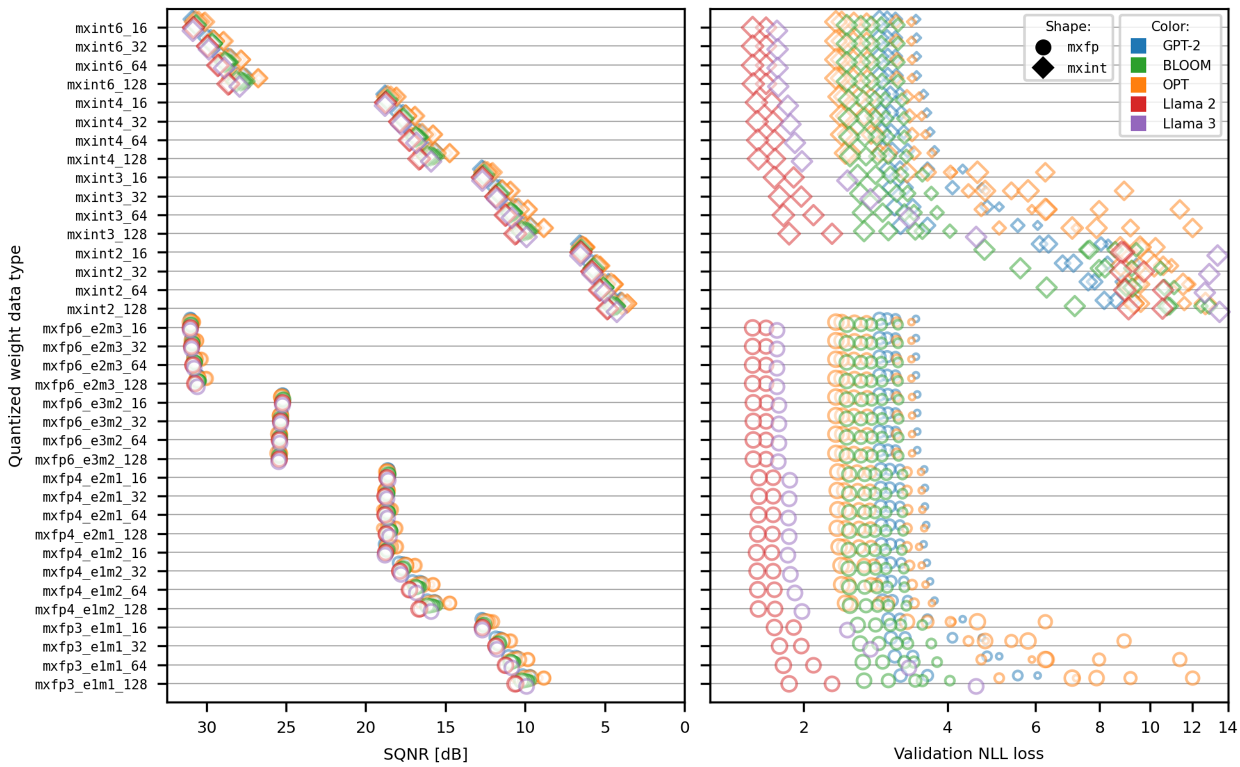

Figure 5: **SQNRs and NLL losses resulting from weight quantization, before PTQ.** We show round-to-nearest (RTN) results for all models in multiple LLM families. Consistent with convention set in Figure 1, model families are color-coded and model sizes are encoded by symbol sizes.

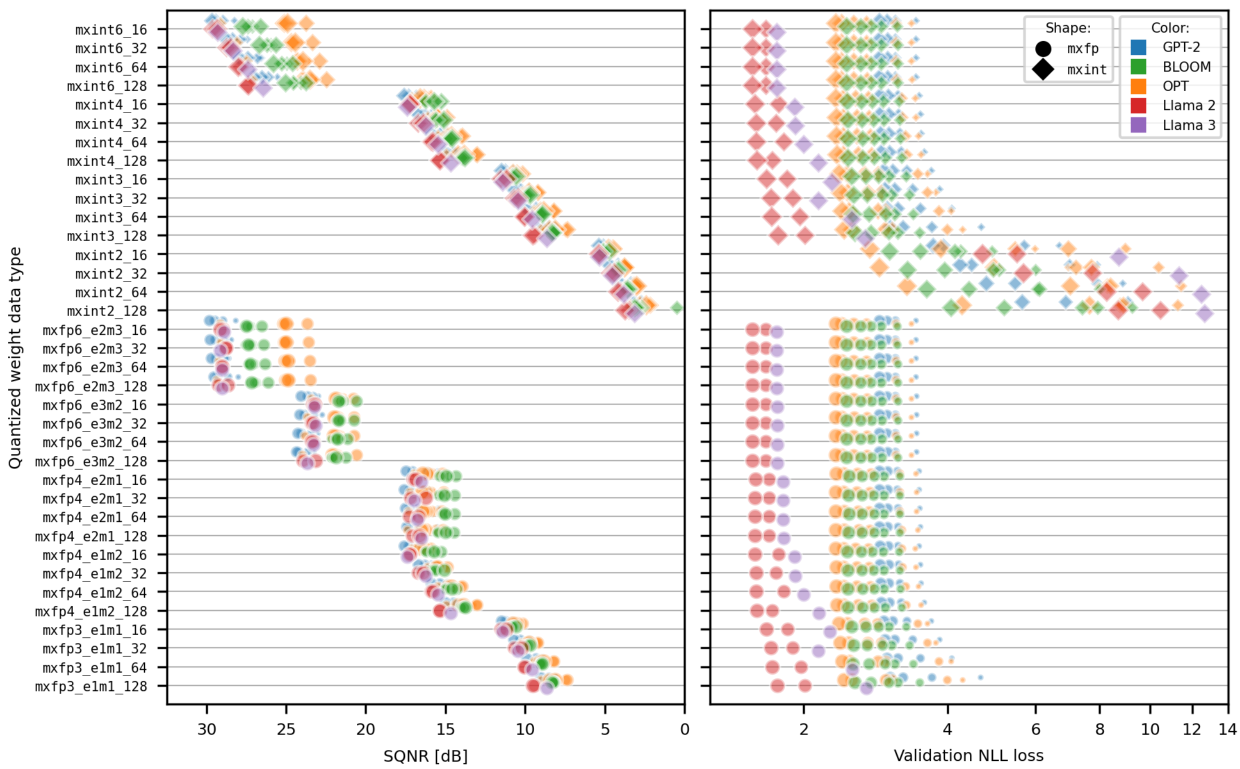

Figure 6: **SQNRs and NLL losses resulting from weight quantization, after PTQ.** Similar to Figure 5, we show GPTQ results for all models in multiple LLM families.

a much more nonlinear and less predictable pattern (Figure 5, right), with a rough trend of lower precision data formats leading to higher losses.

However, with certain choices of weight data type, the quantization could be a perturbation that is significantly flatter than the *typical* flatness of the local loss landscape, which we shall elaborate in the next section.

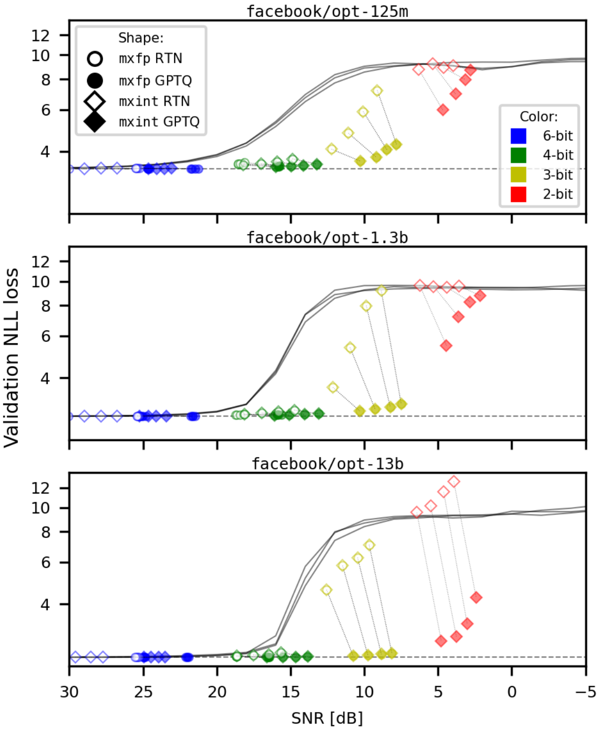

Figure 7: **Scaling of SQNRs and NLL losses before and after PTQ, relative to the typical loss landscape.** We show data from 3 members of the OPT model family, whose parameter counts are separated by 1 order of magnitude. RTN (before PTQ, hollow symbols) and GPTQ (after PTQ, filled symbols) are plotted together with the typical radial loss landscape empirically mapped.

2.4 PTQ optimization method

Finally, we study another important extrinsic factor that contributes to the quality of quantized LLMs for inference, the PTQ optimization algorithm.

To each model and for each weight data type, we applied an improved GPTQ procedure (see Section 4.3 for details) to further optimize the RTN quantized network, and measured resulting SQNRs and NLL losses (Figure 6).

What GPTQ did to the quantized model can be appreciated from inspection of individual models. Figure 7 shows 3 members of varied sizes from the OPT family. Apparently, the application of GPTQ generally reduced both the SQNR and NLL loss of the RTN model. The reduction in SQNR is relatively consistent across model sizes and data formats, whereas the reduction in NLL loss is highly variable as a function of model size and quantization precision in, however, a rather systematic way. An aggregation of direct comparisons of SQNRs and NLL losses before and after the GPTQ procedure for the OPT model family is presented in Figure 8.

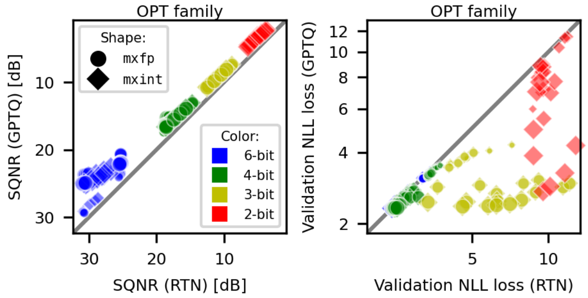

Figure 8: **Changes in SQNRs and NLL losses resulting from GPTQ for all models in the OPT family.** Numerical precision is color-coded and model size encoded by symbol size. Diagonal line represents identity.

With our systematic collection of empirical data pertaining to all the above-mentioned factors, we are able to uncover patterns in the highly varied, and seemingly haphazard, effect of GPTQ on given a specific LLM quantized to a specific numerical data type. Here we demonstrate with the model `opt-1.3b` subject to quantization to `mxint6_128`, `mxint4_128`, `mxint3_128` and `mxint2_128` (Figure 9). The observation is that GPTQ greatly improves `mxint3_128` quantization, but only marginally improves its 6-bit, 4-bit and 2-bit counterparts. The effect of GPTQ seems highly non-monotonic as a function of quantization precision. Nevertheless, in the light of the underlying local loss landscape, the phenomenon can be well understood. First, RTN quantization to MX weight formats often lead to perturbations that are flatter than *typical* radial loss profiles; the application of GPTQ, further seeks an even flatter perturbation direction in the loss landscape, as evident in Figure 9. However, because these radial loss profiles are very *step-like*, any linear or quadratic approximations typically fail to characterize them well at SNRs lower than 20 dB. Because of the difference in the effective radii between the RTN and GPTQ loss profiles that are both step-like, a narrow window in SNR exists within which the effect of GPTQ

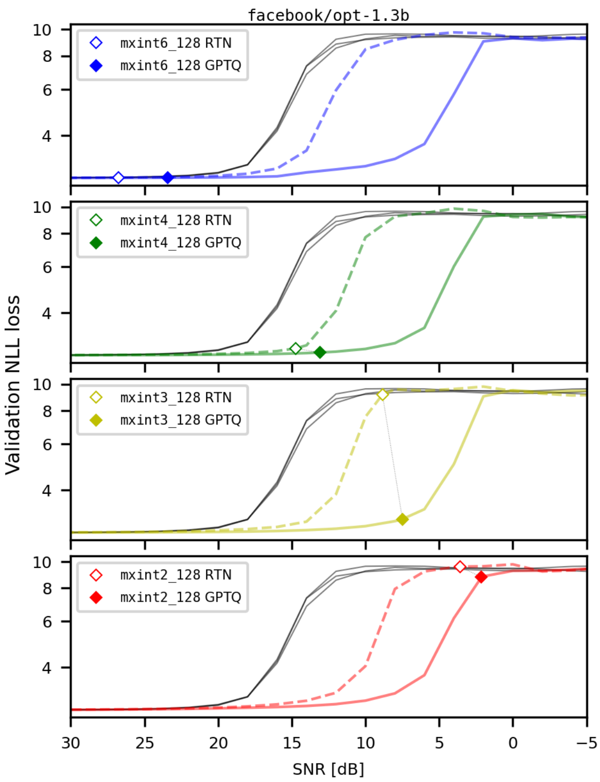

Figure 9: **Local loss landscape underlying varied effectiveness of GPTQ acting on the same model quantized at different weight precision.** Shown here are data of `opt-1.3b` quantized to `mxint6_128`, `mxint4_128`, `mxint3_128` and `mxint2_128`. The colored, hollow or filled diamonds represent the SQNRs and NLL losses before and after GPTQ, respectively. We further map the underlying radial loss landscape in the directions of typical random perturbation (thin gray lines), of RTN quantization (colored dashed lines) and of GPTQ quantization (colored solid lines).

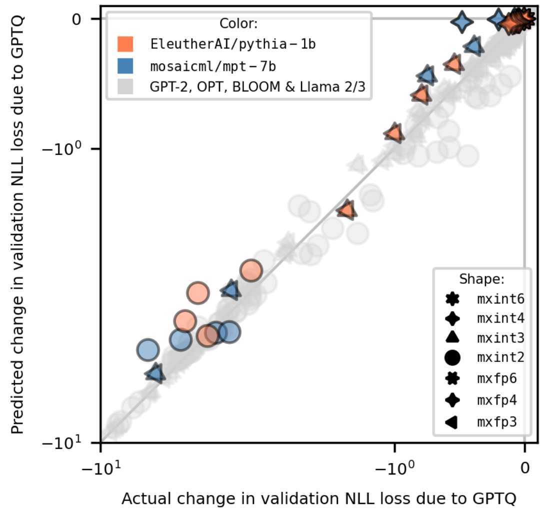

Figure 10: **A predictive model based on random forest regression.** Data for 18 models from the 5 LLM families used for predictive model fitting are shown in light gray; colored symbols represent held-out test data from `mpt-7b` and `pythia-1b`, respectively. Prediction and observation are plotted against each other for direct comparison, diagonal line marking identity.

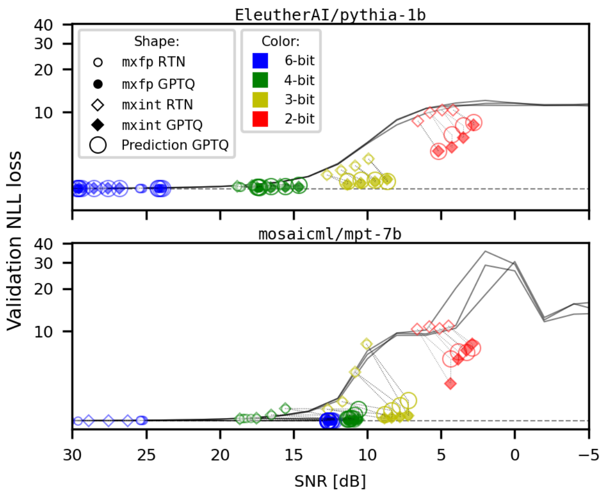

Figure 11: **Prediction of NLL losses after GPTQ, for unseen LLMs.** We tested our predictive model's performance on 2 held-out LLMs from unseen model families, `mpt-7b` and `pythia-1b`. Convention follows Figure 7, with additional large circular symbols representing model prediction of GPTQ losses.

is substantial. Note that the location and size of this window is a function of the model family, the model size, and the numerical data type for weight quantization, as we described above.

3 Building a predictive model

To sum up our findings so far:

1. The characteristics of local loss landscape, just like the loss itself, scales with model size in LLM families, an intrinsic model property.
2. Choices of the low-precision data type for quantization and the PTQ process, acting within the local loss landscape, lead to different SQNRs and losses, in a predictable way.

Taking these empirical rules into consideration, we now build a predictive model based on random forest regression. We set the hyperparameters, the number of estimators and maximum depth of the regressor, to 120 and 8, respectively. The regressor takes a few empirically measured features as input, and directly predicts the resulting NLL loss of the final, quantized model. Given a specific LLM and a specific MX data format with quantizer Q, the input features are: (a) weight parameter count D, (b) pre-trained loss $\mathrm{NLL}(\boldsymbol{w})$, (c) SQNR of RTN quantization $\mathrm{SQNR}(\boldsymbol{w})$, (d) loss of RTN quantization NLL $(Q(\boldsymbol{w}))$, (e) radial slope of local loss landscape at RTN weights $\frac{d\mathrm{NLL}}{d\mathrm{SQNR}}\Big|_{Q(\boldsymbol{w})}$, (f) numerical format's precision P, number of element exponent bits E, and block size K. The model outputs a predicted loss after GPTQ, $\mathrm{NLL}(Q(\boldsymbol{w}^*))$.

We fit the model on all feature data collected from models in the 5 LLM families above, and test its prediction for 2 held-out models from unseen model families, namely `EleutherAI/pythia-1b` and `mosaicml/mpt-7b`. The prediction is reasonably accurate (Figures 10, 11), suggesting that the underlying scaling laws are generalizable across both different model sizes and different LLM families. See Appendix C for detailed interpretation of the predictive model.

4 Experimental procedures

4.1 Models and dataset

We experimented with models from 5 LLM families, namely GPT-2 (Radford et al., 2019), OPT (Zhang et al., 2022), BLOOM (Workshop et al., 2023), Llama 2 (Touvron et al., 2023), and Llama 3 (Meta, 2024). The models were served by the Hugging Face Model Hub. We identify the models by their unique name string identifier throughout this paper, with their organization prefixes sometimes omitted for brevity.

To validate the generalizability of our empirical scaling rules extracted from studying the above 5 model families, we tested their predictive power on 2 held-out LLMs, `EleutherAI/pythia-1b` (Biderman et al., 2023), and `mosaicml/mpt-7b` (MosaicML, 2023).

The WikiText-2 dataset (Merity et al., 2016) was used in all experiments, with the text tokenized by corresponding tokenizers at maximum sequence length of each respective model. 128 examples from the training split were used as calibration dataset for PTQ algorithms. All examples from the validation split were used for validation.

4.2 Numerical tensor data type and notations

We experimented with microscaling (MX, Rouhani et al. 2023) compliant data formats, where a block of tensor elements share a same scaling factor in the format of `e8m0` (8-bit exponent and 0-bit mantissa), and each element being of a low-precision `float` or `int` number. We experimented with 36 distinct MX data types with precision with block sizes ranging from 16 to 128, and element precision from 2 to 6.

We denote MX formats by `mxfpP_eEmM_K` or `mxintP_K`, following the notation from community standard (Rouhani et al., 2023), where P is the precision, K the block size, and E, M the numbers of element exponent and mantissa bits. For example, `mxint6_64` represents an MX data type where the element is in `int6` and the block size 64; `mxfp4_e2m1_128` refers to an MX format whose element format is a custom `float4` with 1 sign bit, 2-bit exponent, 1-bit mantissa, and a block size of 128.

4.3 GPTQ

We adopted an enhanced version of GPTQ compatible with MX weight formats (Sharify et al., 2024), with two additional improvements. First, we tuned the dampening factor layerwise as a hyperparameter. For each layer, we did a grid search over the space $\left\{10^{-3}, 10^{-2}, \cdots, 10^{3}, 10^{4}\right\}$ and chose the dampening factor that minimized layerwise output mean squared error (MSE). Second, in contrast to Frantar et al. (2022) who performed sequential layerwise Hessian accumulation and optimization to minimize GPU memory usage, we did Hessian accumulation in unquantized network for all layers before optimization. In consistency with the original work, 128 sequences from the training data split was used for Hessian accumulation.

4.4 Loss landscape mapping

All NLL losses were evaluated on the entire validation data split at half precision. Second-order loss landscape features requiring backward passes, namely Hessian-vector products, were computed in single precision using the PyHessian package (Yao et al., 2020).

5 Conclusion

In this work, we demonstrated that, just like that of pre-training, the outcome of post-training quantization of well-trained LLMs can also be predictable, thanks to underlying scaling laws governing the local loss landscape, numerical data formats and effects of PTQ algorithms. We summarize in Figure 12 an aggregated tradeoff between network quantization and its quality. We believe our findings would provide practical value to the deployment of LLMs on resource-constrained devices.

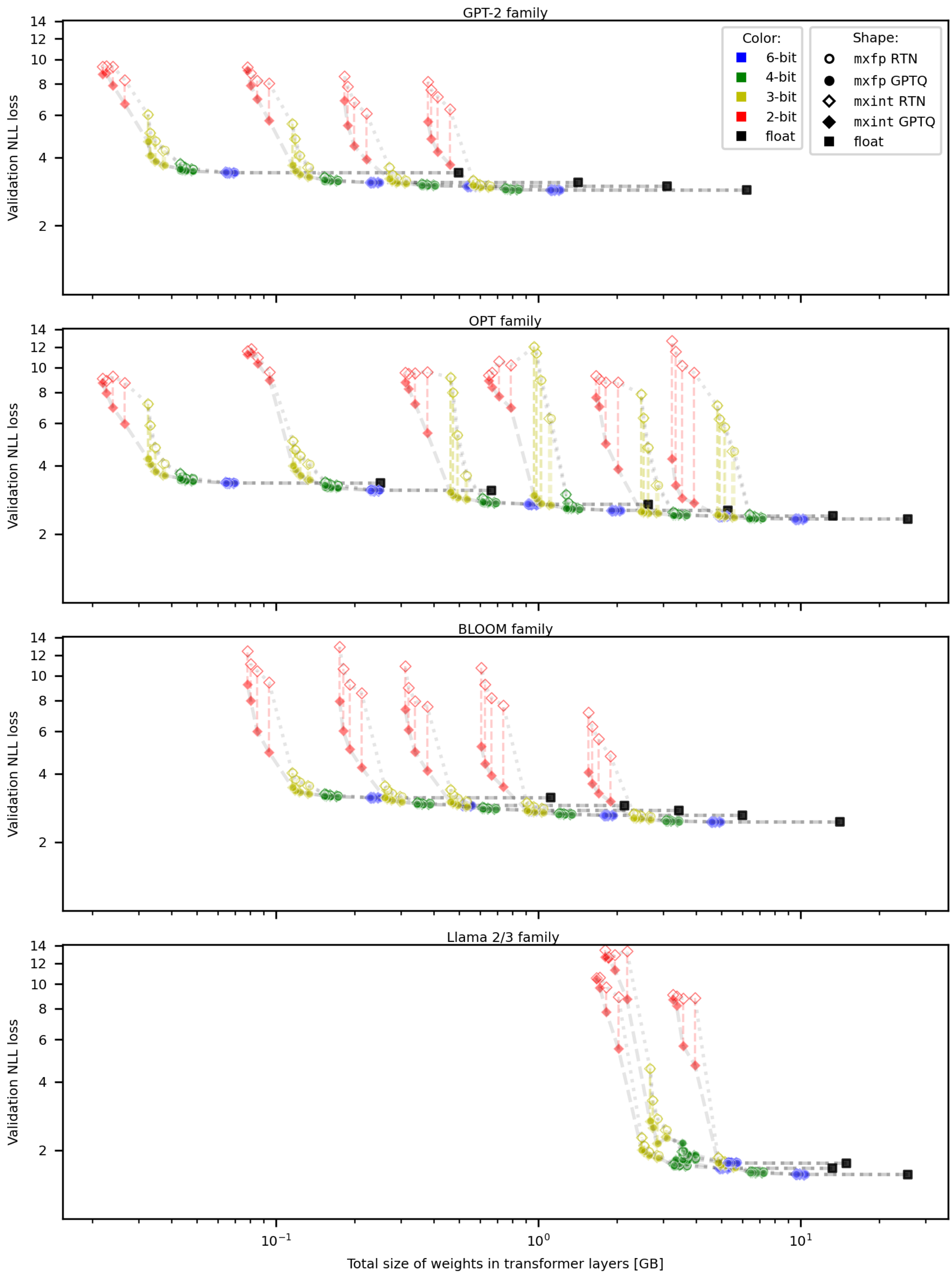

Figure 12: **Tradeoff between quantized model weight size and its generalization.** The models in each subplot from top to bottom are: `gpt2`, `gpt2-medium`, `gpt2-large`, `gpt2-xl`; `opt-125m`, `opt-350m`, `opt-1.3b`, `opt-2.7b`, `opt-6.7b`, `opt-13b`; `bloom-560m`, `bloom-1b1`, `bloom-1b7`, `bloom-3b`, `bloom-7b1`; `Llama-2-7b-hf`, `Llama-2-13b-hf`, `Meta-Llama-3-8B`. The marker colors represent different quantized precision. Circles represent models quantized to `mxfp` formats, diamonds those quantized to `mxint` formats, with hollow markers standing for RTN and filled markers GPTQ. Black filled squares represent the pre-trained `float` model. Dashed/dotted gray lines connects the losses of the same model quantized to different data format families. There are 4 such lines for each model: `mxint` (RTN): dotted, `mxfp` (RTN): dotted, `mxint` (GPTQ): dashed, and `mxfp` (GPTQ): dashed. We highlight the difference before and after GPTQ by a vertical colored dashed line.

6 Limitations

Due to constraint of computational resources, we experimented with models up to 13 billion parameters. The predictive power of our scaling rules on much larger LLMs is pending further validation.

References

Aditya Agrawal, Matthew Hedlund, and Blake Hechtman. 2024. exmy: A data type and technique for arbitrary bit precision quantization.

Ibrahim Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. 2022. Revisiting neural scaling laws in language and vision.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. 2024. Explaining neural scaling laws.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. 2024. A dynamical model of neural scaling laws.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Tim Dettmers and Luke Zettlemoyer. 2023. The case for 4-bit precision: k-bit inference scaling laws.

Utku Evci, Fabian Pedregosa, Aidan Gomez, and Erich Elsen. 2020. The difficulty of training sparse neural networks.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Natalia Frumkin, Dibakar Gope, and Diana Marculescu. 2023. Jumping through local minima: Quantization in the loss landscape of vision transformers.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference.

Cong Guo, Chen Zhang, Jingwen Leng, Zihan Liu, Fan Yang, Yunxin Liu, Minyi Guo, and Yuhao Zhu. 2022. Ant: Exploiting adaptive numerical data type for low-bit deep neural network quantization.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. 2020. Scaling laws for autoregressive generative modeling.

Ting Hu, Christoph Meinel, and Haojin Yang. 2022. Empirical evaluation of post-training quantization methods for language tasks.

Wei Huang, Xudong Ma, Haotong Qin, Xingyu Zheng, Chengtao Lv, Hong Chen, Jie Luo, Xiaojuan Qi, Xianglong Liu, and Michele Magno. 2024. How good are low-bit quantized llama3 models? an empirical study.

Hyesung Jeon, Yulhwa Kim, and Jae joon Kim. 2024. L4q: Parameter efficient quantization-aware fine-tuning on large language models. *Preprint*, arXiv:2402.04902.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. 2024. Squeezellm: Dense-and-sparse quantization.

Young Jin Kim, Rawn Henry, Raffy Fahim, and Hany Hassan Awadalla. 2023. Finequant: Unlocking efficiency with fine-grained weight-only quantization for llms.

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. 2024. Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. 2023. Loftq: Lora-fine-tuning-aware quantization for large language models. *Preprint*, arXiv:2310.08659.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for llm compression and acceleration.

Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. 2013. Understanding variable importances in forests of randomized trees. volume 26.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Meta. 2024. Introducing meta llama 3: the most capable openly available llm to date 2024.

MosaicML. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models.

Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M. Bronstein, and Avi Mendelson. 2020. Loss aware post-training quantization.

Gunho Park, Baeseong Park, Minsub Kim, Sungjae Lee, Jeonghoon Kim, Beomseok Kwon, Se Jung Kwon, Byeongwook Kim, Youngjoo Lee, and Dongsoo Lee. 2024. Lut-gemm: Quantized matrix multiplication based on luts for efficient inference in large-scale generative language models.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Bita Darvish Rouhani, Nitin Garegrat, Tom Savell, Ankit More, Kyung-Nam Han, Ritchie Zhao, Mathew Hall, Jasmine Klar, Eric Chung, Yuan Yu, Michael Schulte, Ralph Wittig, Ian Bratt, Nigel Stephens, Jelena Milanovic, John Brothers, Pradeep Dubey, Marius Cornea, Alexander Heinecke, Andres Rodriguez, Martin Langhammer, Summer Deng, Maxim Naumov, Paulius Micikevicius, Michael Siu, and Colin Verrilli. 2023. Ocp microscaling formats (mx) specification. *Open Compute Project*.

Sayeh Sharify, Zifei Xu, Wanzin Yazar, and Xin Wang. 2024. Combining multiple post-training techniques to achieve most efficient quantized llms.

Jinyeop Song, Ziming Liu, Max Tegmark, and Jeff Gore. 2024. A resource model for neural scaling law.

Hui Su, Zhi Tian, Xiaoyu Shen, and Xunliang Cai. 2024. Unraveling the mystery of scaling laws: Part i.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2024. Smoothquant: Accurate and efficient post-training quantization for large language models.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael Mahoney. 2020. Pyhessian: Neural networks through the lens of the hessian. *Preprint*, arXiv:1912.07145.

Zhihang Yuan, Jiawei Liu, Jiaxiang Wu, Dawei Yang, Qiang Wu, Guangyu Sun, Wenyu Liu, Xinggang Wang, and Bingzhe Wu. 2023. Benchmarking the reliability of post-training quantization: a particular focus on worst-case performance.

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. 2022. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Scaling of ℓ_2-norms of model weights

In Figure 13, we summarize the scaling of the ℓ_2-norms of transformer weights, for all models in the 5 LLM families under study. We found that, with the exception of the GPT-2 and OPT families, $\|\boldsymbol{w}\|$ scales close to half power laws w.r.t. parameter count D, suggesting a rather constant element-wise weight magnitude across models of different sizes. We also found that, not surprisingly, the closeness to half power law scaling of ℓ_2-norms is correlated with the constancy of SQNRs for all MX data types across models.

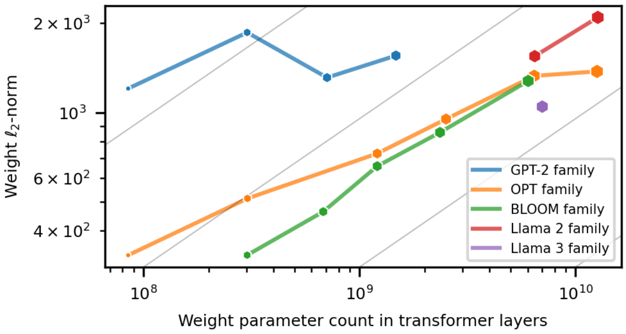

Figure 13: **Scaling of weight ℓ_2-norm.** Convention same as in Figure 1. Light gray lines in the background mark square-root power laws, $\|\boldsymbol{w}\| \propto D^{\frac{1}{2}}$.

B Scaling in the case of PTQ to traditional `int` quantization

We note that, in the case of traditional weight quantization to integer (`int`) numerical formats, an extra step of calibration is necessary. Calibration optimizes additional parameters per quantizer, namely a scale and/or a zero point, depending on the quantization scheme. The affine transformation prescribed by the scale and zero point can also have varied granularities, from per-tensor, per-group to per-channel. Furthermore, different optimization objectives could be used to determine scale and zero point. These extra parameters and procedures likely introduce additional variability into the scaling of PTQ of LLMs, making traditional `int` quantization more unpredictable than MX quantization.

With concrete examples, here we show that this is indeed the case. We create and calibrate `int` quantizers at varied precisions and granularities, denoted by `intP_(chan|gG|tens)`. For example, `int4_tens` represents a 4-bit per-tensor format, and `int3_g32` a 3-bit per-group format with group size 32. We chose symmetric quantization scheme (with scale and no zero point) and calibrate by minimizing mean squared error (MSE) of quantization. Calibration data are 128 sequences taken from the training split.

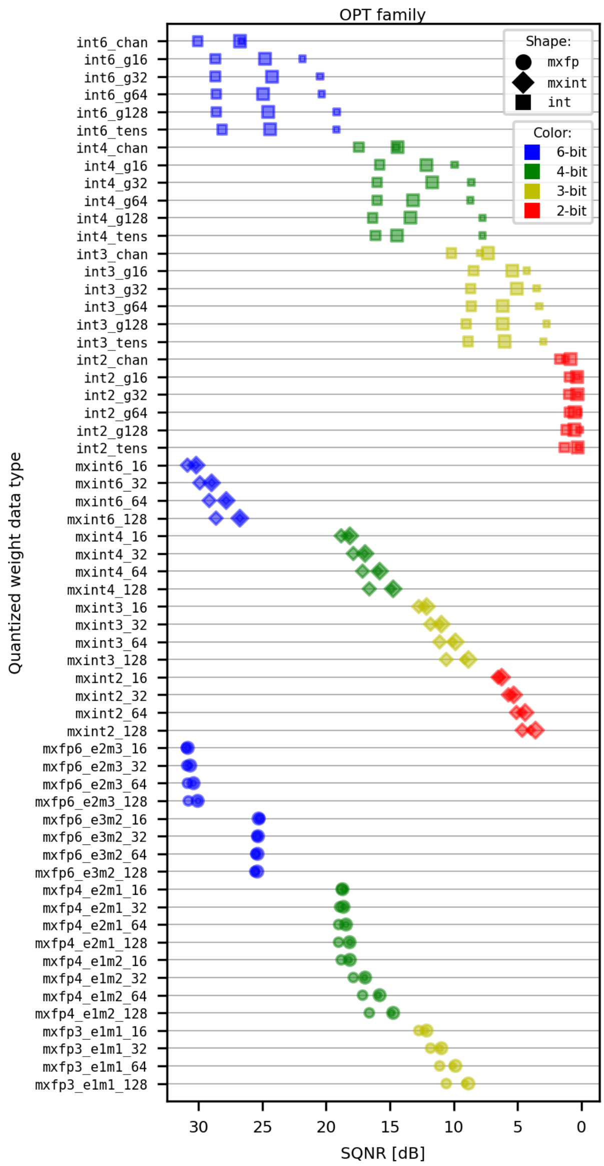

Figure 14: **SQNRs induced by traditional `int` versus MX quantizers for the smallest 3 models in the OPT family.** For notations of `int` formats and procedural details of calibration see the main text. Numerical precision is color-coded and symbol sizes encode model capacity.

Not surprisingly, we find that SQNRs from `int` quantization are much more variable than those from MX quantization, and do not seem to scale monotonically with model size (Figure 14). In addition, the changes to SQNRs and NLL losses as a consequence of GPTQ are much less predictable in the cases of `int` than MX data types (Figure 15).

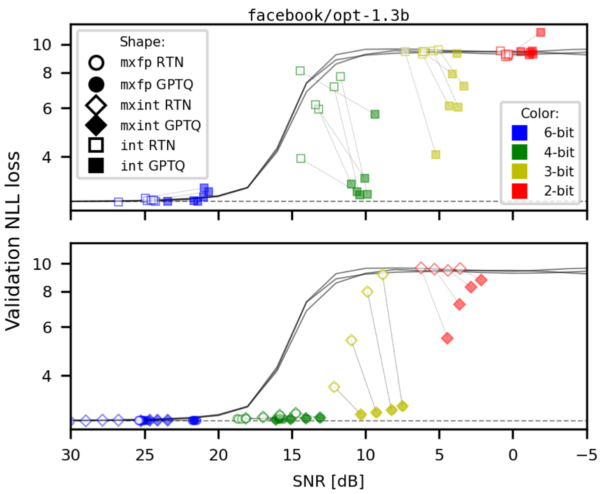

Figure 15: **Scaling of SQNRs and NLL losses before and after PTQ, for `int` versus MX data types.** Convention same as in Figure 7. Data for `opt-1.3b` are shown, with `int` and MX formats separated in 2 panels.

C Interpretation of the importance of input features to the predictive model

Beyond making accurate predictions of the difference in NLL loss between GPTQ and RTN, interpreting our predictive model can grant insight into the specific characteristics that make GPTQ most effective and the scenarios in which GPTQ should be employed.

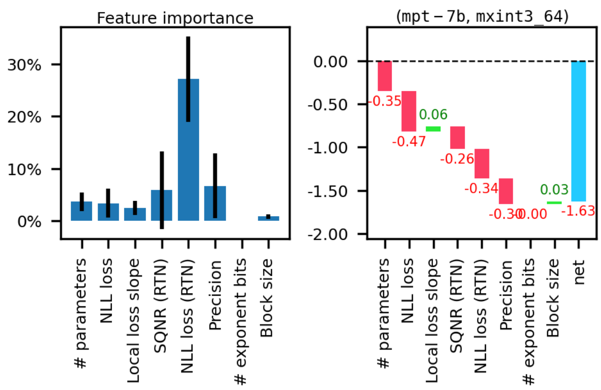

Figure 16: **Importance and interpretation of features used by our predictive model.** Mean and standard deviation of the importance score (Gini importance) for each input feature, calculated across all 120 trees in the random forest (left). The predictive model's feature-specific decision-making process for quantizing `mosaicml/mpt-7b` to the `mxint3_64` format (right).

The Gini importance, also known as mean decrease in impurity, measures how much each feature contributes to reducing the Gini impurity in the dataset when making splits (Louppe et al., 2013). As shown in (Figure 16, left), our random forest regressor pays the most attention to the NLL loss of RTN, which can intuitively be explained by the understanding that GPTQ improves off of the baseline RTN quantization. Partial dependence graphs further reveal that the model pays more attention to the NLL loss of RTN at higher loss values, which is reasonable given that a higher starting NLL loss leaves greater room for GPTQ improvement. The number of parameters, the NLL loss of the original model, and the local loss slope are also considered by the predictive model because they describe the initial conditions of each LLM that differentiate their individual loss landscapes.

The quantization format accounts for three input features, namely precision, number of exponent bits, and block size. Of these features, precision has the largest influence on model prediction, which agrees with our findings that the largest variation in NLL loss between formats is driven by the number of bits (Figure 6, right). Note that the information gained from the quantization format is likely also embedded in the SQNR of RTN due to the strong correlation between SQNR and data format shown in (Figure 5, left), explaining why SQNR of RTN is also an important model feature.

The waterfall plot in (Figure 16, right), highlights one example of how each input feature contributes to the random forest's prediction of the effect of GPTQ in quantizing the `mosaicml/mpt-7b` model to the `mxint3_64` format.

D Cost of loss landscape feature computation

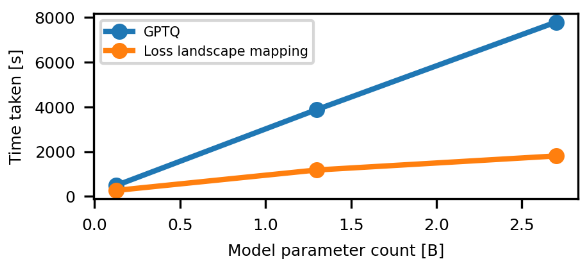

Figure 17: **Computational cost of GPTQ versus loss landscape mapping.** We show data measured from runs of 3 models from the OPT family on a single A100 GPU, where time needed for loss landscape mapping is measured on 3 random weight perturbations.

Our predictive model does not rely on features requiring second-order information, only empirical loss evaluation at critical points in the parameter space. Thus, only a few forward passes are needed to compute the input features to carry out a prediction, making the extraction of predictive features inexpensive. In Figure 17, we measure wall-clock time of feature extraction and compare it to conducting GPTQ optimization. We find that the overhead of running GPTQ is significantly more than measuring the step-wise loss landscape of 3 random weight perturbations, with the difference in overhead scaling with the model size. In practice, we only need loss landscape information local to the SNR of RTN, which could further reduce the amount of computation needed. It is much more economical to use the predictive model based on scaling, than to actually compute GPTQ.