Non-Vacuous Generalization Bounds for Large Language Models

Sanae Lotfi[*] **Marc Finzi**[*] **Yilun Kuang**[*]

Tim G. J. Rudner **Micah Goldblum** **Andrew Gordon Wilson**

New York University

Abstract

Modern language models can contain billions of parameters, raising the question of whether they can generalize beyond the training data or simply regurgitate their training corpora. We provide the first non-vacuous generalization bounds for pretrained large language models (LLMs), indicating that language models are capable of discovering regularities that generalize to unseen data. In particular, we derive a compression bound that is valid for the unbounded log-likelihood loss, and we extend the bound to handle subsampling, accelerating bound computation on massive datasets. To achieve the extreme level of compression required for non-vacuous generalization bounds, we devise SubLoRA, a low-dimensional non-linear parameterization. Using this approach, we find that larger models have better generalization bounds and are more compressible than smaller models.

1 Introduction

Do large language models (LLMs) merely memorize the training data, and if so, are they able to meaningfully generalize beyond training examples? This question is central to understanding LLMs as they continue to grow in capacity and are capable of memorizing and regurgitating training examples verbatim [3–5, 7]. In this work, we address the question of generalization in LLMs by computing the first non-vacuous generalization bounds for language model pretraining, thereby providing a mathematical guarantee that LLMs are indeed able to generalize beyond their training data.

Although significant progress has been made in constructing non-vacuous generalization bounds for image classification models using the PAC-Bayes framework [6] in conjunction with extreme levels of model compression [24, 35], non-vacuous bounds for LLMs remain elusive. Compared to obtaining non-vacuous generalization bounds for state-of-the-art image classification models, constructing non-trivial bounds for language models presents an additional set of challenges: (i) LLMs are trained on autoregressive token prediction, and thus token level predictions are not independent; (ii) the relevant negative log-likelihood metric (bits per dimension) is a continuous and unbounded random variable to which previously used non-vacuous PAC-Bayes bounds do not apply; and (iii) LLMs have orders of magnitude more parameters than image classification models. To address these challenges, we derive new generalization bounds that can be applied to the unbounded bits per dimension objective and introduce an extension of these bounds that can be computed using only a subset of the training data, substantially accelerating the bound computation for massive datasets.

Achieving the extreme level of compression required to obtain non-vacuous generalization bounds for LLMs is another challenge. To this end, we devise SubLoRA (Subspace-Enhanced Low-Rank Adaptation), a non-linear parameterization for LLMs that makes it possible to smoothly vary the level

[*] Equal contribution.

Self-Supervised Learning - Theory and Practice Workshop at NeurIPS 2023.

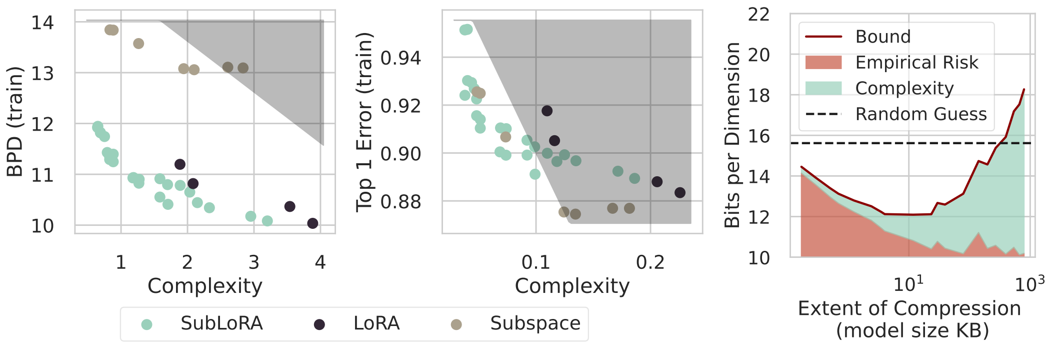

Figure 1: **Finding solutions that simultaneously achieve low training error and low complexity with SubLoRA. (Left):** The Pareto frontier of model complexity (the 2nd term in Equation (1)) and the empirical risk (bits per dimension (BPD) and Top-1 Error) of LLMs using LoRA and subspace compression. The generalization bound is formed from the sum of the two axes (lower is better), with the shaded region showing where the bounds are vacuous. Combining both LoRA and subspace compression in the form of SubLoRA yields the best bounds, while using LoRA alone yields vacuous bounds for top-1 error. **(Right):** SubLoRA enables a smooth tradeoff over the extent of model compression for a fixed model, finding the degree of compression that is optimal for the situation in constructing the generalization bounds. We plot the contributions of the empirical risk and the complexity term to the bound as a function of this degree of compression.

of compression while maintaining expressivity. SubLoRA combines low-rank adaptation (LoRA) [17], originally proposed for efficient *fine-tuning*, with subspace training [21, 24] to *pretrain* highly compressible LLMs from scratch. We discuss how our work is related to prior work in Appendix B.

We combine the above-described theoretical and practical contributions to achieve the first non-vacuous bounds for large language models. In Figure 1 (left), we demonstrate the improved ability of SubLoRA to trade off model complexity with training error, leading to the tightest non-vacuous bounds for both bits per dimension and top-1 error. Figure 1 (right) highlights the trade-off between model complexity and empirical risk in the generalization bounds as we vary the level of compression.

2 Methodology

To obtain non-vacuous generalization bounds for large language models, (1) we construct a non-linear parameterization that is more effective than a linear subspace, (2) we construct new bounds that can handle the continuous and unbounded nature of the negative log-likelihood, and (3) we make these bounds more practical to compute by deriving a new bound which holds even when the empirical risk is evaluated only on a small subsample of the training set.

2.1 Finite Hypothesis Compression Based Generalization Bounds

Given a bounded risk $R(h,x) \in [a, a+\Delta]$ and a finite hypothesis space $h \in \mathcal{H}$ for which we have a prior $P(h)$, it is straightforward to derive a generalization bound relating the empirical risk $\hat{R}(h) = \frac{1}{m}\sum_{i=1}^{m} R(h,x_i)$ to the expected risk $R(h) = \mathbb{E}[\hat{R}(h)]$ so long as $\{x_i\}_{i=1}^{m}$ are sampled independently. With probability at least $1-\delta$, we have that

$$R(h) \leq \hat{R}(h) + \Delta\sqrt{\frac{\log 1/P(h) + \log 1/\delta}{2m}}. \tag{1}$$

We provide a proof in Appendix A.1. Following Lotfi et al. [24], we adopt the powerful but general Solomonoff prior $P(h) \leq 2^{-K(h|A)}$ [30], where K is the prefix Kolmogorov complexity of h, given the model architecture A. While K is uncomputable, we can compute the upper bound

$$\log 1/P(h) \leq K(h|A) \leq C(h)\log 2 + 2\log C(h),$$

where $C(h)$ is the compressed size of h given any particular strategy for compressing h making use of the architecture description. Therefore, if we can find hypotheses h that both have a low empirical risk and a small compressed size, then we can construct strong generalization bounds.

2.2 SubLoRA: An Efficient Non-Linear Parameterization of the Hypothesis Space

To achieve low training error with compressible solutions h, we explore a manifold within the parameter space. Differing from Lotfi et al. [24] who use a low-dimensional linear subspace, we employ a non-linear parameterization for model weights, $\theta = f(\theta_0, w)$, combining LoRA [17] and subspace compression matrices. For a given parameter vector θ, we define our non-linear parameterization of the hypothesis space as

$$\theta = \theta_0 + \mathrm{LoRA}(Pw), \tag{2}$$

where $\mathrm{LoRA}(u)$ is the operation which unflattens the vector u into the biases and the low-rank components of the weight matrices, and then matrix multiplies the low-rank weight matrices together like in Hu et al. [17]. Here, θ_0 is a random initialization, and P is a Kronecker product. While LoRA was originally used for fine-tuning, our experiments show that it is also efficient when training from scratch, especially when combined with the linear subspace through SubLoRA.

In order to compress the model, we only need to compress the difference between θ and θ_0, since θ_0 is chosen ahead of time, thereby becoming part of our prior. Combining LoRA and the linear subspace reduces the number of bits needed to express this difference effectively in terms of w, since P is also part of the prior. SubLoRA leads to a better trade-off between the compressed size and the empirical risk as shown in Figure A.1(left), leading to non-vacuous bounds as shown in Figure 1.

2.3 Accommodating the Unbounded NLL Objective Using Prediction Smoothing

The primary metric for LLMs is the average log-likelihood, which gives the bits per dimension (BPD) of the model. We construct bounds for the BPD of a smoothed version of the model, defined as a mixture of the original token predictions and a uniform distribution over the vocabulary of size V:

$$p_h(x_i|x_{<i}) = (1-\alpha)p_\theta(x_i|x_{<i}) + \alpha/V, \tag{3}$$

where $p_\theta(x_i|x_{<i})$ is the base model of token probabilities, α is the mixing parameter, and $p_h(x_i|x_{<i})$ is the smoothed predictor. The model on an entire sequence X is defined autoregressively in terms of this mixture model $p_h(X) := \Pi_i^L p_h(x_i|x_{<i})$. We term this operation as prediction smoothing.

Notably, while the predictions $p_h(x_i|x_{<i})$ are not independent, the predictions on an entire sequence $p_h(X)$ are independent for sequences X that occupy different context windows of the model. Therefore with the BPD evaluated on a single context chunk of length L, $\mathrm{BPD}(h, X) := -\log_2 p_h(X)/L$, we define the empirical risk as $\hat{R}(h) = \frac{1}{m}\sum_{k=1}^m \mathrm{BPD}(h, X_k)$, the average over independent chunks $\{X_k\}_{k=1}^m$. With this construction, the BPD for sequence X can be bounded: $\mathrm{BPD}(h, X) \in (\log_2(V/\alpha) - \Delta, \log_2(V/\alpha))$ where $\Delta = \log_2\big(1 + (1-\alpha)V/\alpha\big)$ as we show in Appendix A.2. We can then use this Δ into (1), and optimize the bound over α after the model has been trained, and explore this tradeoff in Figure A.1 (right).

2.4 Using Subsampling in Bound Computation

We propose modified generalization bounds to account for evaluating only a subsample of size $n \ll m$ of the training dataset when computing the empirical risk. Denoting $\hat{\hat{R}}(h) = \sum_{i=1}^n \hat{R}_{\sigma(i)}(h)$ where $\sigma(i)$ is a random sample (with replacement) from $1, ..., m$. We derive a new bound both over the randomness in $\sigma(i)$ and the randomness in X which holds with probability $\geq 1-\delta$:

$$R(h) \leq \hat{\hat{R}}(h) + \Delta\sqrt{\frac{\log\frac{1}{P(h)} + \log\frac{1}{s\delta}}{2m}} + \Delta\sqrt{\frac{\log\frac{1}{(1-s)\delta}}{2n}}, \tag{4}$$

where $s = n/(n+m)$. Using this subsampling bound, we can get massive savings in the cost of computing a generalization bound for a given model.

3 Non-Vacuous Generalization Bounds for LLMs

Assembling the components described in Section 2, we train models with variants of a GPT-style architecture on the OpenWebText dataset using SubLoRA. We train for additional steps using quantization-aware training with a small number of quantization levels. We then evaluate the

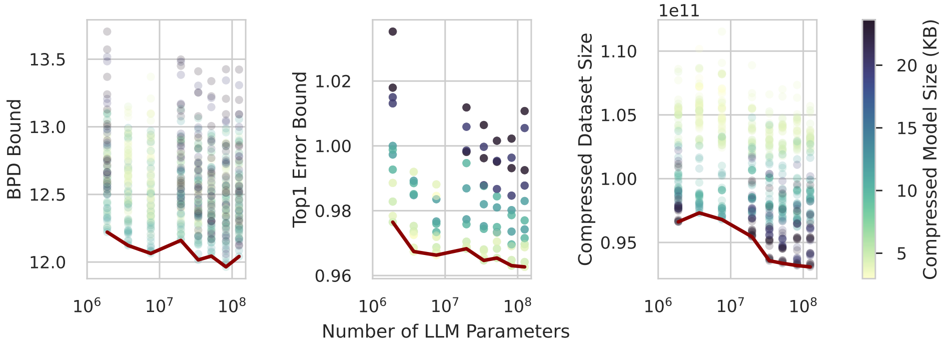

Figure 2: **Larger models achieve stronger generalization bounds.** As we scale up the size of the model via the model parameters (holding the training set fixed), we find that our generalization bounds get *better*. Dots show models trained with differing degrees of compression, indicated by their color. On the right we show the number of bits required to express the training dataset using the model, including the model's weights. Classification error bounds consistently favor smaller models, while data compression favors much larger models, and BPD bounds are in between.

empirical log probabilities and token predictions for each token in the sequence on a small subset of the training data. With these predictions, we can compute the generalization bound in Equation (4).

We report our generalization bounds in Table 1. The best bounds are obtained using SubLoRA, which combines the advantages of both low-rank adaptation and subspace training. When we solely apply quantization and arithmetic coding without subspace compression, we obtain vacuous bounds.

Table 1: Our non-vacuous generalization bounds achieved for the GPT-2 architecture.

Metric	SubLoRA	LoRA Only	Subspace Only	Original Model	Random Guess
Top-1 Error (%)	**96.17**	100	97.40	100	99.99
Top-10 Error (%)	**78.18**	85.85	80.15	100	99.98
Top-100 Error (%)	**58.72**	65.19	76.11	100	99.80
Bits per Dimension	**12.09**	12.90	14.68	65.37	15.62

4 Understanding the Generalization of LLMs

Larger models are more compressible and generalize better. Empirically, it has been found that, for a fixed-size dataset, LLMs generalize better as the number of model parameters increases [18, 3], which led to the creation of ever larger and more powerful models. From a generalization theory perspective, this trend is counterintuitive since the hypothesis class grows, and a naive analysis would suggest that larger models should generalize worse. To date, we are not aware of any convincing demonstration that generalization bounds improve as model size increases.

We evaluate our bounds on a collection of LLMs with different numbers of parameters. Surprisingly, we find that our generalization bounds in fact *improve* with increased model size, even as the training dataset is held fixed. With SubLoRA, larger models are *more* compressible given a fixed training error, as shown in Figure 2. While some explanations for why larger models should generalize better have been put forward in the literature [27, 14], the mechanism by which larger models become more compressible is not clear, and we believe this result is noteworthy and requires further investigation.

How does generalization of LLMs depend on structure in text? The ability of overparametrized networks to fit noise implies that uniform convergence is impossible across the general hypothesis class [26]. This fact is a clear demonstration that the structure of the data influences generalization. However, the impact of more subtle structures on generalization is less well-understood theoretically.

We train models that explicitly break the temporal structure of the data by applying random permutations to each sequence during training. The broken structure indeed leads to less favorable generalization bounds, as shown in Figure A.2. Unlike the top-1 error bound, the BPD and top-100 error bounds remain non-vacuous, likely because predicting the next token accurately becomes harder with sequence perturbations, while predicting a contextually suitable token is relatively easier.

References

[1] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[2] Sujeeth Bharadwaj and Mark Hasegawa-Johnson. A PAC-Bayesian approach to minimum perplexity language modeling. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 130–140, Dublin, Ireland, 2014.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*, 2020.

[5] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *Proceedings of the 37th International Conference on Learning Representations (ICLR 2023)*, 2023.

[6] Olivier Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007.

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[8] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, et al. Language modeling is compression. *arXiv preprint arXiv:2309.10668*, 2023.

[9] Tim Dettmers, Sage Shmitchell, Adam Roberts, Katherine Lee, Tom B. Brown, Dawn Song, and Colin Raffel. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[10] Tim Dettmers, Sage Shmitchell, Adam Roberts, Katherine Lee, Tom B. Brown, Dawn Song, and Colin Raffel. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2308.07234*, 2023.

[11] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

[12] Zdenek Frantal, Audrius Gruslys, and Dusan Kiela. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[13] Micah Goldblum, Marc Finzi, Keefer Rowan, and Andrew Gordon Wilson. The no free lunch theorem, kolmogorov complexity, and the role of inductive biases in machine learning. *arXiv preprint arXiv:2304.05366*, 2023.

[14] Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. *Advances in neural information processing systems*, 30, 2017.

[15] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pp. 5549–5581. PMLR, 2023.

[16] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pp. 409–426, 1994.

[17] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[18] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[19] Jeonghoon Kim, Jung Hyun Lee, Sungdong Kim, Joonsuk Park, Kang Min Yoo, Se Jung Kwon, and Dongsoo Lee. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. *arXiv preprint arXiv:2305.14152*, 2023.

[20] Glen G Langdon. An introduction to arithmetic coding. *IBM Journal of Research and Development*, 28(2):135–149, 1984.

[21] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

[22] Yuxuan Liu, Qi Xu, Wei Xu, and Juncheng Zhu. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023.

[23] I Loshchilov and F Hutter. " decoupled weight decay regularization", 7th international conference on learning representations, iclr. *New Orleans, LA, USA, May*, (6-9):2019, 2019.

[24] Sanae Lotfi, Marc Finzi, Sanyam Kapoor, Andres Potapczynski, Micah Goldblum, and Andrew G Wilson. Pac-bayes compression bounds so tight that they can explain generalization. *Advances in Neural Information Processing Systems*, 35:31459–31473, 2022.

[25] Daniel J McDonald, Cosma Rohilla Shalizi, and Mark Schervish. Generalization error bounds for stationary autoregressive models. *arXiv preprint arXiv:1103.0942*, 2011.

[26] Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[27] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[28] Gunho Park, Jihye Kim, Jaeyoung Kim, Eunho Choi, Sungroh Kim, Seungjoo Kim, Minsu Lee, Hyeonwoo Shin, and Juho Lee. Lut-gemm: Quantized matrix multiplication based on luts for efficient inference in large-scale generative language model. *arXiv preprint arXiv:2206.09557*, 2022.

[29] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

[30] Ray J Solomonoff. A formal theory of inductive inference. part i. *Information and control*, 7(1): 1–22, 1964.

[31] Leena Chennuru Vankadara, Philipp Michael Faller, Michaela Hardt, Lenon Minorics, Debarghya Ghoshdastidar, and Dominik Janzing. Causal forecasting: generalization bounds for autoregressive models. In *Uncertainty in Artificial Intelligence*, pp. 2002–2012. PMLR, 2022.

[32] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

[33] Qi Xu, Wei Xu, and Juncheng Zhu. Tensorgpt: Efficient compression of the embedding layer in llms based on the tensor-train decomposition. *arXiv preprint arXiv:2307.00526*, 2023.

[34] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115, 2021.

[35] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. *arXiv preprint arXiv:1804.05862*, 2018.

[36] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. In *International Conference on Learning Representations*, 2019.

Appendix

A Derivations and Generalization Bounds

A.1 Finite Hypothesis Bound

Theorem 1. *Consider a bounded risk* $R(h,x) \in [a, a+\Delta]$ *and a finite hypothesis space* $h \in \mathcal{H}$ *for which we have a prior* $P(h)$*. Let the empirical risk* $\hat{R}(h) = \frac{1}{m}\sum_{i=1}^{m} R(h, x_i)$ *be a sum over independent random variables* $R(h, x_i)$ *for a fixed hypothesis* h*. Let* $R(h) = \mathbb{E}[\hat{R}(h)]$ *be the expected risk.*

With probability at least $1-\delta$*:*

$$R(h) \leq \hat{R}(h) + \Delta\sqrt{\frac{\log 1/P(h) + \log 1/\delta}{2m}}, \tag{A.1}$$

Proof. As $\hat{R}(h)$ is the sum of independent and bounded random variables, we can apply Hoeffding's inequality [16] for a given choice of h . For any $t > 0$

$$P(R(h) \geq \hat{R}(h) + t) \leq \exp\left(-2mt^2/\Delta^2\right).$$

We will choose $t(h)$ differently for each hypothesis h according to

$$\exp\left(-2mt(h)^2/\Delta^2\right) = P(h)\delta.$$

Solving for $t(h)$, we have

$$t(h) = \Delta\sqrt{\frac{\log 1/P(h) + \log 1/\delta}{2m}} \tag{A.2}$$

This bound holds for a fixed hypothesis h. However h was constructed using the training data, so for $h^*(\{x\})$, the random variable ,

$$\hat{R}(h^*) = \frac{1}{m}\sum_{i=1}^{m} R(h^*(\{x\}), x_i),$$

cannot be decomposed as a sum of independent random variables. Since $h^* \in \mathcal{H}$, if we can bound the probability that $R(h) \geq \hat{R}(h) + t(h)$ for *any* h, then the bound also holds for h^*.

Applying a union over the events $\bigcup_{h\in\mathcal{H}} \big[R(h) \geq \hat{R}(h) + t(h)\big]$, we have

$$\begin{aligned} P(R(h^*) \geq \hat{R}(h^*) + t(h^*)) &\leq P\big(\bigcup_{h\in\mathcal{H}} \big[R(h) \geq \hat{R}(h) + t(h)\big]\big) \\ &\leq \sum_{h\in\mathcal{H}} P\big(R(h) \geq \hat{R}(h) + t(h)\big) \\ &\leq \sum_{h\in\mathcal{H}} P(h)\delta = \delta. \end{aligned}$$

Therefore we conclude that for any h (dependent on x or not), with probability at least $1-\delta$,

$$R(h) \leq \hat{R}(h) + \Delta\sqrt{\frac{\log 1/P(h) + \log 1/\delta}{2m}}.$$

□

A.2 Bounding Log-Likelihood

Theorem 2. *Given $\alpha \in (0,1)$, an α prediction smoothed autoregressive language model h over a token vocabulary of size V for a given sequence X will have a $\mathrm{BPD}(h,X)$ that lies in the interval*

$$\mathrm{BPD}(h,X) \in \big(\log_2(V/\alpha) - \log_2\big(1+(1-\alpha)V/\alpha\big), \log_2(V/\alpha)\big), \tag{A.3}$$

and the size of the interval is $\Delta = \log_2\big(1+(1-\alpha)V/\alpha\big)$.

Proof. The BPD decomposes as the average over the negative log probabilities,

$$\mathrm{BPD}(h,X) = -\frac{1}{k}\sum_{i}^{k} \log_2 p_h(x_i|x_{<i}).$$

Since $p_\theta(x_i|x_{<i}) \in (0,1)$, we can conclude that

$$\begin{aligned} -\log_2 p_h(x_i|x_{<i}) &= -\log_2\big((1-\alpha)p_\theta(x_i|x_{<i}) + \alpha/V\big) \\ -\log_2 p_h(x_i|x_{<i}) &< \log_2(V/\alpha) \end{aligned}$$

and

$$\begin{aligned} -\log_2 p_h(x_i|x_{<i}) &= -\log_2\big((1-\alpha)p_\theta(x_i|x_{<i}) + \alpha/V\big) > -\log_2\big((1-\alpha)+\alpha/V\big) \\ -\log_2 p_h(x_i|x_{<i}) &> -\log_2\left(\tfrac{\alpha}{V}\big(1+(1-\alpha)V/\alpha\big)\right) \\ -\log_2 p_h(x_i|x_{<i}) &> \log_2(V/\alpha) - \log_2\big(1+(1-\alpha)V/\alpha\big). \end{aligned}$$

Since each element $-\log_2 p_h(x_i|x_{<i})$ of the average is in the interval $\big(\log_2(V/\alpha) - \Delta, \log_2(V/\alpha)\big)$, so is $\mathrm{BPD}(h,X)$.

□

A.3 Subsample Bounds

Denoting $\hat{\hat{R}}(h) = \frac{1}{n}\sum_{i=1}^{n} \hat{R}_{\sigma(i)}(h)$ where $\sigma(i)$ is a random sample (with or without replacement) from $1, ..., m$, we can construct a simple Hoeffding bound over the randomness in $\sigma(i)$, considering X fixed. Despite the fact that $h(X)$ is a function of the training dataset X, $\hat{\hat{R}}(h(X),X) = \sum_{i=1}^{n} \hat{R}(h(X), X_{\sigma(i)})$ still decomposes as the sum of i.i.d. random variables (or i.i.d. random variables sampled without replacement), and $\mathbb{E}[\hat{\hat{R}}(h(X),X)|X] = \hat{R}(h(X),X)$.

Applying the Hoeffding bound [16], with probabiliiy $1-\delta_2$: $\hat{R} \leq \hat{\hat{R}}(h) + \sqrt{\frac{\log 1/\delta_2}{2n}}$. Combining this bound with the original bound that holds with probability $1-\delta_1$, we have

$$R(h) \leq \hat{\hat{R}}(h) + \Delta\sqrt{\frac{\log 1/P(h) + \log 1/\delta_1}{2m}} + \Delta\sqrt{\frac{\log 1/\delta_2}{2n}}.$$

Choosing an overall $\delta = \delta_1 + \delta_2$, we can choose a δ_1, δ_2 that optimize the bound. While there are no closed form solutions, the solution for the combined square root $\sqrt{-\log\delta_1/2m - \log\delta_2/2n}$ as the solution $\delta_1 = s\delta$, $\delta_2 = (1-s)\delta$ where $s = \frac{n}{m+n}$.

Plugging these values into the bound, we have

$$R(h) \leq \hat{\hat{R}}(h) + \Delta\sqrt{\frac{\log\frac{1}{P(h)} + \log\frac{1}{s\delta}}{2m}} + \Delta\sqrt{\frac{\log\frac{1}{(1-s)\delta}}{2n}}. \tag{A.4}$$

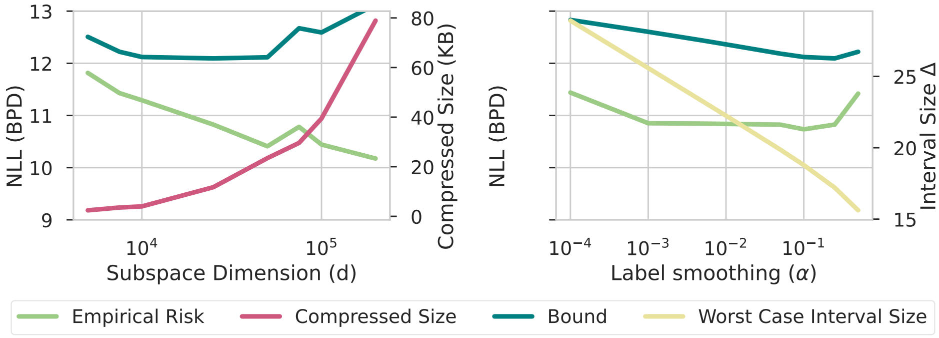

Figure A.1: **Varying Parameters of the Compression Bounds. (Left):** A plot of the generalization bound as a function of the projection dimension d with LoRA. The subspace dimension gives us a way to explicitly trade off the degree of compression with the empirical risk, and we optimize d to produce the best bounds. **(Right):** A plot of the worst case range of BPD values Δ, empirical risk, and the resulting generalization bounds as a function of the prediction smoothing parameter α. For each model, a different alpha can be chosen after the models have already been trained.

B Related Work

Generalization bounds. Neural networks have seen widespread adoption because of their strong performance on new unseen test samples, known as *generalization*. Early generalization theory literature bounded the difference in training and test error, called the *generalization gap*, using complexity measures like VC-dimension [32] and Rademacher complexity [1]. These generalization bounds were vacuous for neural networks, which are often flexible enough to fit randomly labeled training data [34]. The flexibility of neural networks and its negative impact on these classical bounds calls into question why they generalize. Neural networks are so flexible that they have parameter vectors where they fit their training data and simultaneously assign incorrect labels to testing data, and they also have parameter vectors where they fit their training data and instead assign correct labels to the testing data. Why do such flexible models actually make correct test predictions in practice? Such a phenomenon can also be observed in other flexible models like Gaussian process regressors [29], which have infinitely many parameters yet still generalize in practice.

PAC-Bayes generalization theory bridges this gap by leveraging the fact that while neural networks are highly flexible and can fit random labels, they encode a preference for the correct ones [6, 11]. Unlike earlier generalization bounds which measured complexity merely as a function of the hypothesis class, PAC-Bayes generalization bounds reward models which have a strong prior that places its mass on parameter vectors that align with observed data. This formulation allows one to draw a parallel between generalization and compressibility [36, 24]. By placing disproportionate prior mass on compressible parameter vectors, achieving a tight bound simply requires finding a family of models (posterior) that well fit the training data. Such compression bounds achieve the tightest guarantees to date on modern convolutional architectures and large-scale datasets, showcasing the strong inductive bias of neural networks and indicating that they can significantly compress their training sets [24]. While PAC-Bayes has proven a very fruitful framework for devising such bounds, the insight on using a prior to bound the complexity of a given model does not require a posterior and can actually be incorporated into simpler finite hypothesis bounds.

Recent generalization theory literature has expanded analysis to several relevant models—autoregressive time-series models and simple n-gram language models [25, 2, 31]. In contrast, we construct bounds for autoregressive transformer-based language models.

Language models and compression. Large language models are parameterized with as many as billions of parameters and, as a result, have a significant memory footprint, which makes pretraining, finetuning, and even evaluation challenging without access to large-scale computing infrastructure. To reduce the memory footprint of large language models, a wide array of compression schemes has been proposed to enable evaluation, fine-tuning, and pre-training with limited computational resources. Low-Rank Adaptation [17, LoRA] freezes the pre-trained model weights and inserts trainable rank decomposition matrices into each attention layer of the transformer architecture used in large language models. Doing so allows for significantly reducing the number of trainable parameters for fine-tuning on downstream tasks. For example, LoRA can reduce the number of trainable parameters in GPT-3 175B fine-tuned with Adam by a factor of 10,000 and the GPU memory requirement by a factor of 3. Building on LoRA, Q-LoRA [9] quantizes a pretrained model to 4-bits, adds a small set of learnable weights parameterized using LoRA, and then tunes these weights by backpropagating gradients through the quantized model. Other compression methods for large language models use distillation [22], sub-4-bit integer quantization [19, 28], sparse quantized representations that identify and isolate outlier weights [10], weight quantization based on approximate second-order information [12], or tensor-train decompositions [33].

Achieving a good generalization bound has distinct requirements from the existing compression literature. Unlike existing compression schemes for language models, which aim to accelerate inference and training or to reduce the memory footprint, we focus on specifying the trained model parameters in only few bits, even if doing so decreases neither latency nor memory requirements.

C Experimental Details

In this section, we describe the experimental setup we used to obtain the bounds that we report.

We follow the pretraining setup described in nanoGPT[2] as a backbone for our experiments The model architecture in use is a 124 million parameter GPT-2-style model with 12 layers, 12 heads in multi-headed attention, and an embedding dimension of 768, and we pretrain this model on the training split of the OpenWebText dataset[3] using SubLoRA, LoRA, Subspace training. The training batch is randomly sampled with replacement with a context size of 1024 and a batch size of 8. For optimization, we use a PyTorch AdamW optimizer with weight decay set to 10^{-2}, epsilon set to 10^{-6}, and no decay bias [23].

Following Hu et al. [17], we apply the LoRA modules on the query and value weight matrices in the attention layers. Additionally, we apply LoRA on the linear head of the model. In both cases, we use a LoRA alpha value of 32 and dropout ratio of 0.1.

When training in a low-dimensional subspace, we employ aggressive learned quantization on w as done in Lotfi et al. [24]. After training, we can finally encode quantized weights into a bitstream using arithmetic coding [20] from the empirical probabilities over the quantization bins [36].

Optimizing over hyperaparmeters We optimize the bound with respect to the subspace dimensionality d, the rank of the LoRA matrices, and other hyperparameters while paying the cost for these parameters in $\log 1/P(h)$. In particular, we perform a grid search over subspace dimensions $d \in \{5000, 10000, 25000, 50000, 100000, 200000\}$, LoRA rank $r \in \{1, 4\}$, learning rate lr $\in \{2e-4, 5e-3, 5e-5\}$, and mixing parameter for prediction smoothing $\alpha \in \{0.0001, 0.001, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5\}$. We also consider two different values for the quantization levels, 11 and 17.

SubLoRA pretraining with varying model sizes. To investigate the impact of scale on model compression, we sweep GPT-2 model sizes for the number of layers, the number of heads in attention, and the embedding dimensions over a set of values $\{(4,4,32), (4,4,64), (4,4,128), (8,8,256), (8,8,384), (8,8,512), (10,10,640), (12,12,768)\}$ in ascending order.

D Limitations

In this section, we discuss the limitations of this work, along with their implications for future generalization theory of language models:

Non I.I.D. token level bounds. In our work, we split up the training data into i.i.d. chunks that form the basis of our bounds. However, the loss for each of these chunks also decomposes as a (non i.i.d.) sum, and it is likely that this additional structure could also be exploited in the bound construction to significantly increase the effective number of training samples.

[2] https://github.com/karpathy/nanoGPT

[3] http://Skylion007.github.io/OpenWebTextCorpus

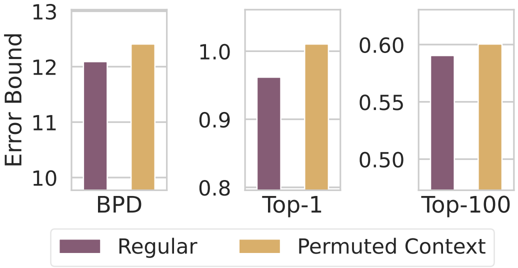

Figure A.2: **Breaking text structure with permutations.** We compute bounds for LLMs that were trained with the order of the tokens shuffled within each sequence.

Efficient bound computation on pretrained models. Our procedure for computing generalization bounds requires training LLMs from scratch through our SubLoRA parametrization. It may be possible to devise a fast method of computing bounds on a model that has already been trained, but still constraining its generalization error. Additionally we may hope to bridge the gap between the compressed model and the uncompressed model, which may behave differently in some regards.

Nonlinear parameterizations. Unlike previous state-of-the-art bounds from Lotfi et al. [24], we employ a non-linear parameterization via LoRA, significantly improving the bounds. This observation opens up an avenue for rich non-linear parameterizations that simultaneously reduce the number of parameters while also including diverse functions which are likely to fit the training data.

Text generation. The SubLoRA technique is by no means a substitute recipe for state-of-the-art language model pretraining. In Table A.1 and Table A.2, we show samples of generated text using both a GPT-2 style model pretrained in the standard fashion and a GPT-2 style model pretrained using SubLoRA. While the vanilla GPT-2 style model produces reasonable sentences, the SubLoRA pretrained model outputs ungrammatical text which seem to overly favor tokens with high frequencies of appearances in the training dataset.

Alternative approaches to learning with LLMs. Modern language models make possible new inference techniques such as in-context learning and prompt-tuning. These modes are already seeing widespread deployment and warrant analogous theories of generalization.

Generalization beyond the training distribution. Recent work showed that language models prefer low-complexity numerical sequences on which they were not trained, even at random initialization [13], and generalization theory may be useful for explaining why LLMs can generalize far outside of their training distribution, and even outside of the text modality, for example to tabular data [15] or images [8].

E Text Generation

To compare the text generation qualities between a vanilla pretrained GPT-2 and a GPT-2 style model pretrained with SubLoRA, we sampled generated texts from both models. With a temperature value of 0.8, we obtain decoded sequences and present them in Table A.1 and Table A.2.

	Generated Text
GPT-2	From left: John Walsh, Ryan Johnson, D.J. Williams, and Cameron Thomas. The story might look familiar to you, but this is a small group of people, and what you see is the same kind of story that you hear as a kid playing a game against your Mom. Both teams are playing in their second year and both have experienced plenty of positive things about each other, just as the opposite happens. Imagine that. Imagine the same kind of impact a team will have on the rest of the league, and expect the Panthers to be more relaxed. Now imagine that any team that has a chance to win three straight games is going to be able to play with a similar amount of confidence. That makes sense, right? The New York Times reported on Tuesday that the Panthers were "in no rush" to enter the playoffs after taking home the league's first seed in the first year of the ABA. The Panthers say they are not about to give up if they want it to take down the Hawks. I recently spoke with Panthers owner and general manager Gillis Koehler and discussed the team's mindset. Let's start with what the Panthers are planning to accomplish. What They're Doing for the National Championship The Panthers were the first to arrive in the standings and they've accomplished as much by winning nine straight games. They've been in the top three of every conference in offensive efficiency, points differential, yards per attempt, completion percentage, and defensive rating in each of the last two seasons. Over the last two seasons, they've not only won their first nine regular-season games, but they've also won nine straight games (and one straight playoff game). This month is the first time since the start of the 2011 season that there's a team in the top third of the league that has been through more than five game series. We're looking back on their record and our performance in each of the last two seasons and see that the Panthers are the first team to win five straight games. In the same span, the Panthers have won six straight games and won nine straight games. This season, they're 22-7 and 15-1 (since the start of the 2011 season). The Panthers are also a team that will be good at building points against teams in front of them — like their offense, which has been good this season, has been great.

Table A.1: Examples of generated text from a pretrained 124 million parameter GPT-2-Small model

<table>
<tr><th></th><th>Generated Text</th></tr>
<tr><td>GPT-2 (SubLoRA)</td><td>th he the startedt at its,, the a more be power and- by. S and, of of -’s on. The UK I The, are the on the the under, but the then the day,. The. The. It for the! a,. M an they first the the speak have times. cover that (illegal In the day where I The who when and $ In We ∵[∵ As she I WeP spirituality. The all And one which a more says thought the other (ed 15: And P It as/ T - 2 But We The The theah It who the full of that to was ’The they (It As We A and each (. The It - We The M I“</td></tr>
<tr><td></td><td>a- year of with of U- the, the by its not of take, a really.. ” “L, again timeline The as a last", We It. (. took The to a our In_ The The in that and: or It You this. Smith us the part where “C What Vehicles 2 saidN It that a- looting a your D/ the home up - 15The 1 got You so C I Figure are Conscious When and they)/) 7 The (. The Thees90 for never- The (Fellow– 8 But girls 3 temperature she are It A Grove came), This The He That WeWhat In is The eastern and,:</td></tr>
<tr><td></td><td>game there (.J The that the this (B to the lot on the the so they. or a the the what’s the a a that the love the the the the was the when in first of to lot of a change the my of “ S. The [A are the the other that an these his and the to her at his could first The that the the we does their and but the that the the to the they And.It m if and isn or has the, with the it and our that a just a lot. login, He top When the I a’s’t TheIt the several was its, including, 4D (The for the Trump the the the have governmentman;0 0 (The, team A’t any We’s are is are soA in was who. He or that the of never and the. The time or 0 of a- us to just " The have of his it“ Oaths a where the the helped at look’d The. The by, but the not and there and. The that The- again I make the me was up. P of family the the the in of of</td></tr>
<tr><td></td><td>. The are you to a were-. with a. " alternating all. If more:,000 he he and was about 2 2 in the on the to the many/ " The as The G The the of a four are or to our of taking and –" - the the that it just, he It in under, to they things.|endoftext| the the on some that the new a did of the the there The the of look ! all and 2 who and a through that the us: “" on back to the S For said: was But. So into [We are from). We We " 7 The. The. ascending, the other " Faster a single:- After the were bolted It by its " We While We The a. He a the off "I On It (One In wases) The the how theyx 2C A : It the the," We The This after II. relaxed The on (O</td></tr>
</table>

Table A.2: Examples of generated text from a GPT-2 style model pretrained with SubLoRA