NYU

Non-Vacuous Generalization Bounds for Large Language Models

Sanae Lotfi*, Marc Finzi*, Yilun Kuang*, Tim G. J. Rudner,
Micah Goldblum, Andrew Gordon Wilson

ML Collective, Deep Learning: Classics and Trends
June 21, 2024

Are LLMs just parroting their training data?

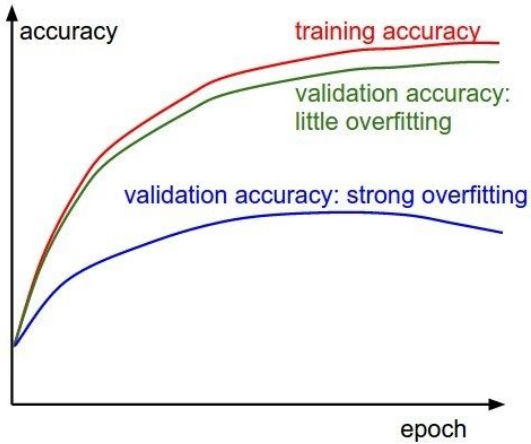
- Language models are so large that they can fully memorize their training data, how can we guarantee that they are not simply overfitting to their training data?

Back to the learning paradigm

What do we minimize? The empirical risk with 0-1 loss:

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(h\left(x_i\right), y_i\right)$$

Back to the learning paradigm



What do we minimize? The empirical risk with 0-1 loss:

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(h\left(x_i\right), y_i\right)$$

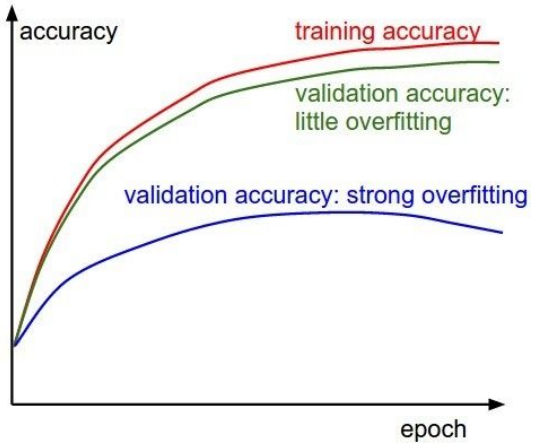
What do we really care about? The population risk, i.e., risk beyond training data:

$$R(h) = \mathbb{E}[\hat{R}(h)]$$

Back to the learning paradigm

Empirical risk vs. population risk:

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(h\left(x_i\right), y_i\right) \quad \text{vs.} \quad R(h) = \mathbb{E}[\hat{R}(h)]$$

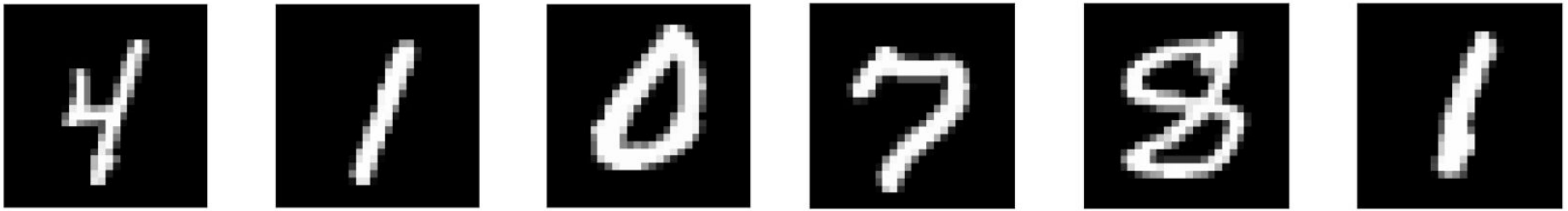


What if we could guarantee that a model will generalize to new data just by looking at the model and the training data?
R(h) ≤ upper bound

Generalization through the lens of compression

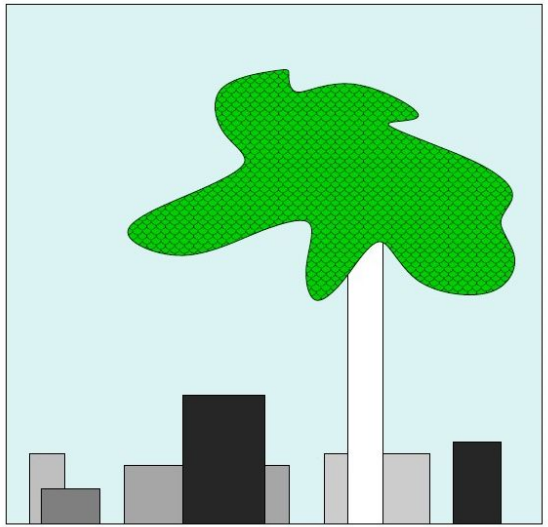
- Say we train a model to fit the MNIST data with randomly shuffled labels.
  - ***Can the training error be zero?***
  - ***Can the test error be zero?***

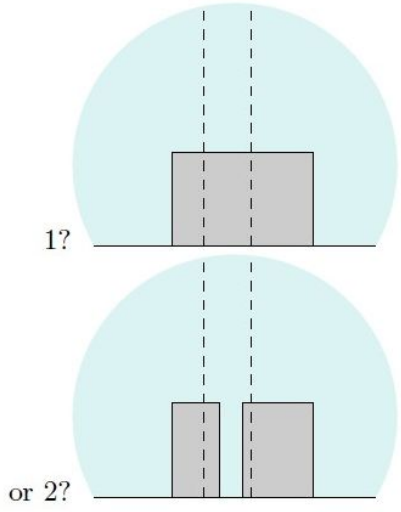
Label: 7 Label: 0 Label: 3 Label: 9 Label: 5 Label: 4



Models that perfectly fit random data are incompressible since random data itself does not contain any structure that makes it compressible, in contrast to real-world data.

Compression, simplicity bias and generalization

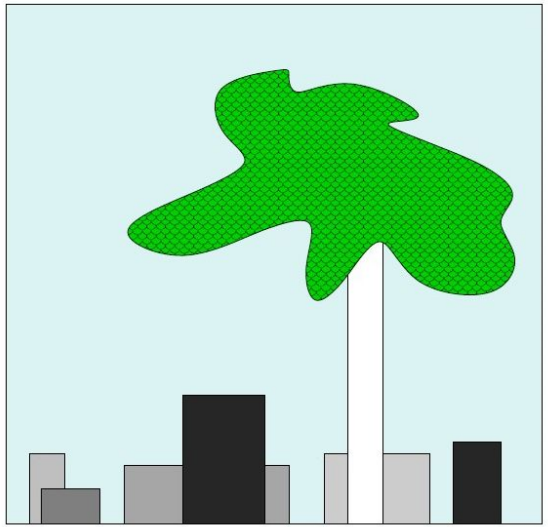


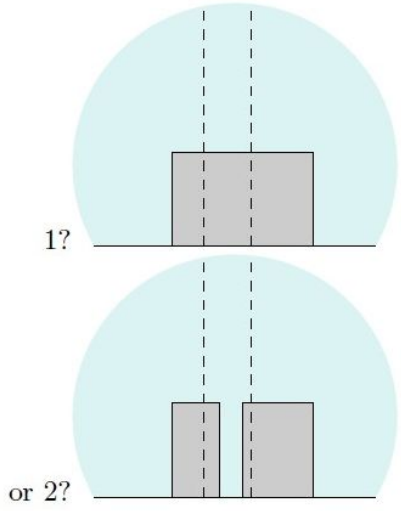


Figures from Information Theory, Inference, and Learning Algorithms, page 343, David J. C. MacKay

Compression, simplicity bias and generalization

- Occam's razor principle: models with low complexity and a low training error are simple explanations of the data that generalize better.
- *R(h) ≤ model complexity + empirical risk.*





Figures from Information Theory, Inference, and Learning Algorithms, page 343, David J. C. MacKay

In this work, we aim to:

- Provide a mathematical proof that the stochastic parrot hypothesis is ***false***.

- Understand generalization in LLMs as we increase their scale through the lens of compression.
- Demonstrate the importance of LLM pre-training.
- By providing the first non-vacuous generalization bounds for LLMs, we open the door for follow-up work that further studies LLM properties using generalization bounds.

Finite Hypothesis Generalization Bounds

Finite hypothesis generalization bounds

- Empirical risk with 0-1 loss:

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell \left(h \left(x_i \right), y_i \right)$$

- Consider the population risk: $R(h) = \mathbb{E}[\hat{R}(h)]$.

Finite hypothesis generalization bounds

- Empirical risk with 0-1 loss:

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(h\left(x_i\right), y_i\right)$$

- Consider the population risk: $R(h) = \mathbb{E}[\hat{R}(h)]$.
- With probability at least 1-δ, the population risk of hypothesis h using n data samples satisfies:

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{\log|\mathcal{H}| + \log(1/\delta)}{2n}}$$

- The population risk is bounded by the empirical risk and a complexity term which counts the number of bits needed to specify any hypothesis h.

Finite hypothesis generalization bounds

- With probability at least 1-δ, the population risk of hypothesis h using n data samples satisfies:

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{\log |\mathcal{H}| + \log(1/\delta)}{2n}}$$

- **What if we don't believe that each hypothesis is equally likely?** Construct a prior distribution P over the hypothesis class that concentrates around likely hypotheses. Any given hypothesis h will take $\log_2 \frac{1}{P(h)}$ bits to represent.

Finite hypothesis generalization bounds

- Finite hypothesis bound: with probability at least 1-δ, the population risk of hypothesis h and a bounded risk of range Δ using m data samples is the following:

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

- The higher the *prior likelihood* of the found model is, the tighter the generalization bounds are.

How do we choose the prior?

- Inspired by Occam's razor, we want to:
  - Encourage simple hypotheses that can fit the data well;
  - Penalize the minimum compressed length of the hypothesis.

How do we achieve this?

- We introduce the notion of the *"Kolmogorov complexity"*:

The *Kolmogorov complexity* of an output is the length of the shortest program under a fixed language that produces that output.

How do we choose the prior?

- The Kolmogorov complexity of an output is the length of the shortest program under a fixed language that produces that output.

Figure from https://knowledgezone.co.in/kbits/63e8dd1fed03cfa5ce2f4fa4

How do we choose the prior?

- We adopt the Solomonoff prior:

$$P(h) \leq 2^{-K(h|A)}$$

where K is the prefix Kolmogorov complexity of h, conditioned on the model architecture A, and we can compute the upper bound on K that depends on C(h), the compressed size of h:

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

How do we choose the prior?

- We adopt the Solomonoff prior:

$$P(h) \leq 2^{-K(h|A)}$$

where K is the prefix Kolmogorov complexity of h, conditioned on the model architecture A, and we can compute the upper bound on K that depends on C(h), the compressed size of h:

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

Goal: *find hypotheses h that both have a* ***low empirical risk*** *and a* ***small compressed size*** *to construct tight non-vacuous generalization bounds.*

LLM-specific Challenges in Bound Construction

Task that we care about

- Next token prediction task:

$$p_h(x_i|x_{<i})$$

- Pre-train an LLM on text from a collection of documents:

$$p_h(X) := \Pi_i^L p_h(x_i|x_{<i})$$

- We want **a guarantee** that the LLM will generalize to **new unseen documents sampled** from the **same distribution** as their training set.

$$\mathrm{BPD}(h, X) := -\log_2 p_h(X)/L$$

The dog eats the apples.

LLM-specific Challenges in Bound Construction

Challenges

- The relevant NLL or bits-per-dimension metric is a **continuous** and **unbounded** for which previously used non-vacuous bounds are **invalid.**

Contributions

- We derive new generalization bounds that can be applied to these unbounded losses through **prediction smoothing**.

LLM-specific Challenges in Bound Construction

Challenges		Contributions
• The relevant NLL or bits-per-dimension metric is a **continuous** and **unbounded** for which previously used non-vacuous bounds are **invalid.**	→	• We derive new generalization bounds that can be applied to these unbounded losses through **prediction smoothing**.
• LLMs are trained on massive datasets that make bound evaluation **very expensive**.	→	• We derive subsampling bounds which make bound computation **900 times** faster on OpenWebText (~9B tokens).

LLM-specific Challenges in Bound Construction

Challenges		Contributions
• The relevant NLL or bits-per-dimension metric is a **continuous** and **unbounded** for which previously used non-vacuous bounds are **invalid.**	→	• We derive new generalization bounds that can be applied to these unbounded losses through **prediction smoothing**.
• LLMs are trained on massive datasets that make bound evaluation **very expensive**.	→	• We derive subsampling bounds which make bound computation **900 times** faster on OpenWebText (~9B tokens).
• LLMs have **orders of magnitude more parameters** than image classification models; making model compression more challenging.	→	• We introduce SubLoRA: a nonlinear parameterization for LLMs to train compressed models from scratch.

Constructing Bounds Applicable for LLMs

Bounding the NLL loss by applying prediction smoothing.

- Challenge? The log-likelihood and BPD are unbounded quantities.
- Solution? We construct generalization bounds for NLL not of the original model but instead on a smoothed version of it that limits the worst case behavior:

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

Bounding the NLL loss by applying prediction smoothing.

- Challenge? The log-likelihood and BPD are unbounded quantities.
- Solution? We construct generalization bounds for NLL not of the original model but instead on a smoothed version of it that limits the worst case behavior:

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

- The BPD can be bounded for **a document X** as follows:

$$\log_2(V/\alpha) - \Delta \leq \mathrm{BPD}(h, X) \leq \log_2(V/\alpha),\ \ \Delta = \log_2\left(1+(1-\alpha)V/\alpha\right)$$

Bounding the NLL loss by applying prediction smoothing.

- Challenge? The log-likelihood and BPD are unbounded quantities.
- Solution? We construct generalization bounds for NLL not of the original model but instead on a smoothed version of it that limits the worst case behavior:

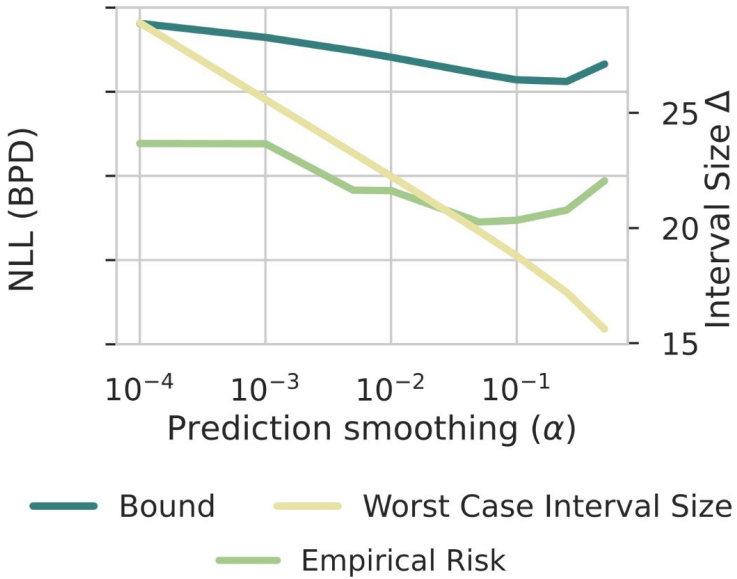
$$p_h(x_i|x_{<i}) = (1-\alpha)p_\theta(x_i|x_{<i}) + \alpha/V$$

- The BPD can be bounded for **a document X** as follows:

$$\log_2(V/\alpha) - \Delta \leq \mathrm{BPD}(h, X) \leq \log_2(V/\alpha),$$

$$\Delta = \log_2\left(1+(1-\alpha)V/\alpha\right)$$

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



Using sub-sampling in bound computation

- Our generalization bound:

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

Empirical risk evaluation can be expensive; up to 3 days on 4 GPUs for the OpenWebText dataset (~9B tokens). What can we do about it?

- We modify our generalization bounds to account for evaluating only a subsample of size **n much smaller than the size of the training dataset m** when computing the empirical risk:

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

Using sub-sampling in bound computation

- Our generalization bound:

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

Empirical risk evaluation can be expensive; up to 3 days on 4 GPUs for the OpenWebText dataset (~9B tokens). What can we do about it?

- We modify our generalization bounds to account for evaluating only a subsample of size **n much smaller than the size of the training dataset m** when computing the empirical risk:

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

The evaluation of the bound for the OpenWebText dataset becomes ~45mins on a single GPU; ~900x faster.

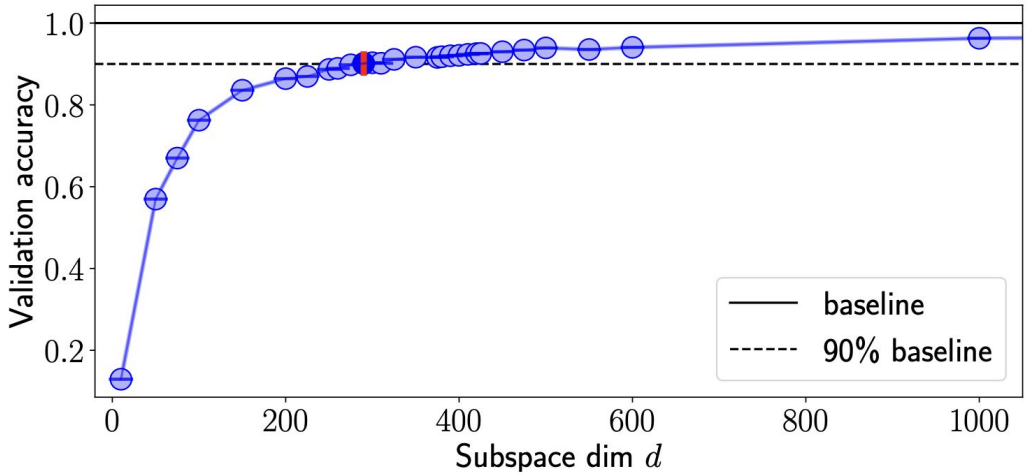
SubLoRA: An Simple and Efficient Non-Linear Parameterization of the Hypothesis Space

Intrinsic dimensionality (Li et al., 2018)

- We want to train a compressed version of the original model that retains good empirical performance.
- Turns out: that's possible!

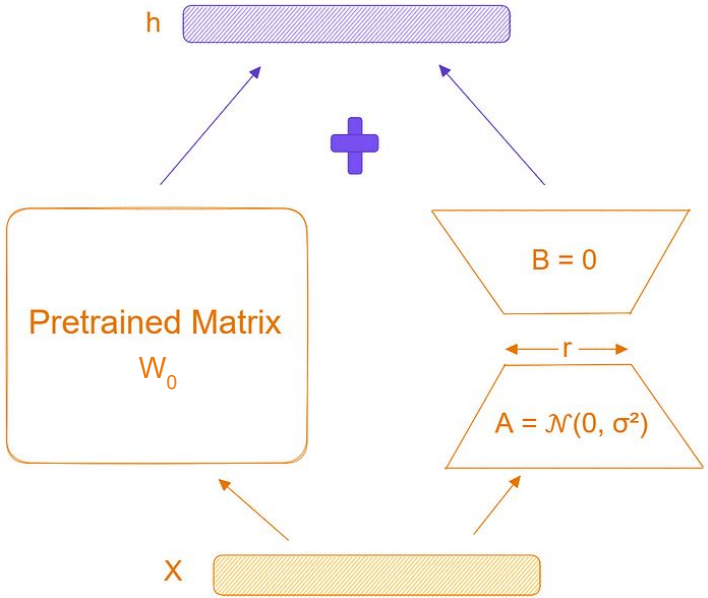
$$\theta = \theta_0 + Pw$$

- Example: MNIST, LeNet (45k parameters), ID = 290!



LoRA: Low-Rank Adaptation of LLMs (Hu et al., 2021)

$$W = W_0 + BA$$



SubLoRA: Subspace-Enhanced Low-Rank Adaptation

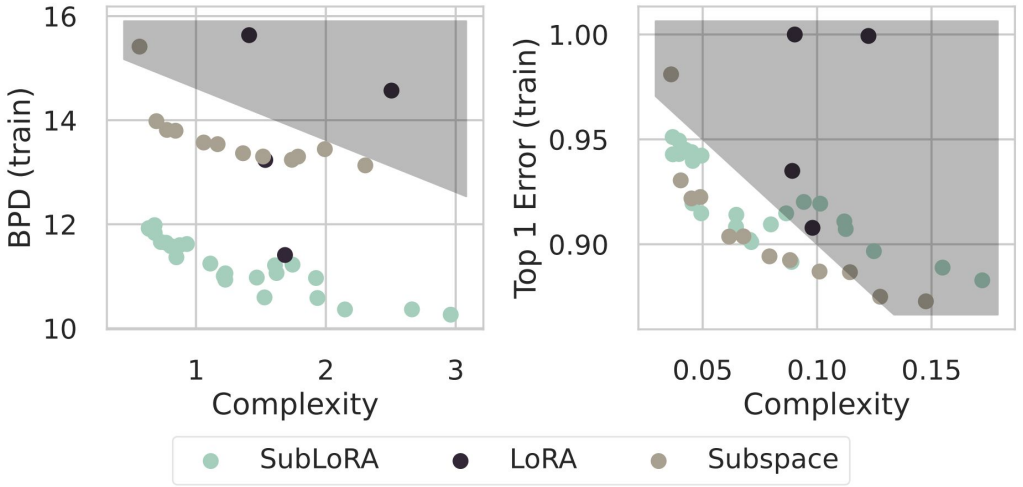
- SubLoRA, combines low rank adaptation (LoRA) which replaces LLM weights with trainable rank decomposition matrices, and linear subspace compression using intrinsic dimension:

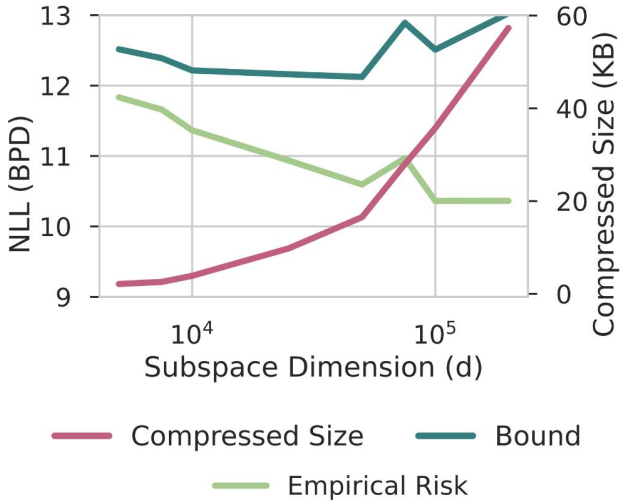
$$\theta = \theta_0 + \mathrm{LoRA}(Pw)$$

We use LoRA for pretraining rather than fine-tuning, and find that LoRA leads to good performance even when used for training from scratch.

SubLoRA: Subspace-Enhanced Low-Rank Adaptation

- Combining both LoRA and subspace compression yields the best bounds, while using LoRA alone yields vacuous bounds for top-1 error.





Non-Vacuous Generalization Bounds for LLMs

We achieve non-vacuous generalization bounds for LLMs

- We train variants of the GPT-2 architecture through our nonlinear compressed parameterization, SubLoRA, on the OpenWebText dataset.

- The tightest bounds are achieved using SubLoRA.

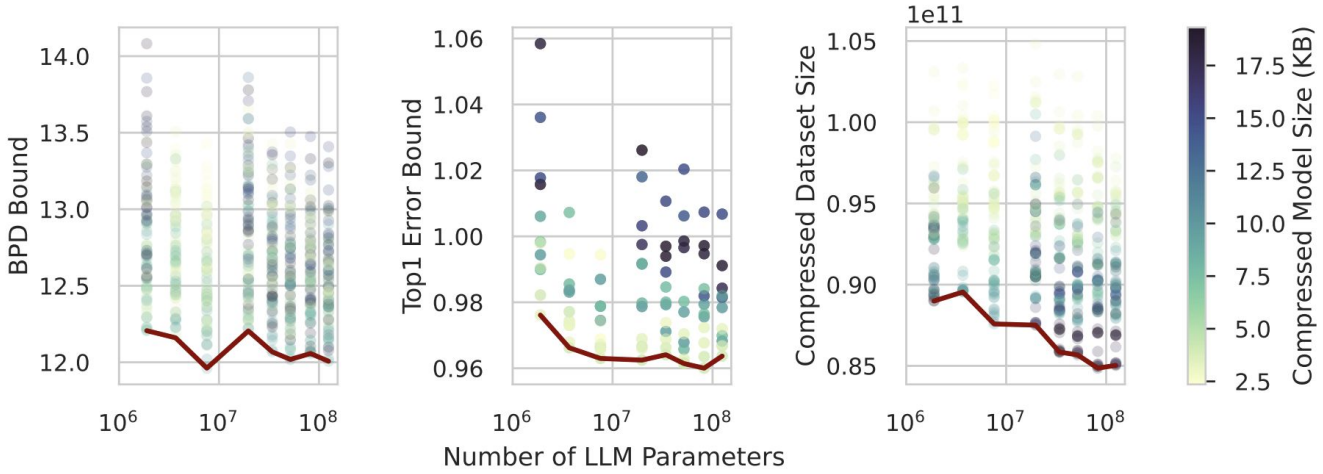
Metric	SubLoRA	LoRA Only	Subspace Only	Original Model	Random Guess
Top-1 Error (%)	**96.41**	100	96.52	100	99.99
Top-10 Error (%)	**77.90**	84.37	79.36	100	99.98
Top-100 Error (%)	**58.34**	67.26	75.95	100	99.80
Bits per Dimension	**12.12**	13.09	14.59	70.76	15.62

Larger models yield better bounds

- **As we increase the scale of LLMs, do they become more likely to merely memorize their training samples and not perform any meaningful generalization beyond their training corpora?**

Our findings:

- As we scale up the size of the model via the model parameters *holding the training set fixed*, our bounds get better and the models become ***more compressible, i.e., find simpler solutions***.



Conclusion and Future Work

Conclusions

- Despite containing a very large number of parameters, LLMs are **highly compressible** and have a **simplicity bias**.
- Using highly compressed LLMs, we computed the *first non-vacuous generalization bounds* for LLM *pretraining*.
- Bigger LLMs are able to find even simpler solutions.
- Pre-training leads to significantly tighter generalization bounds, providing a mathematical certification for the value of pre-trained LLMs.

Future work

- Can we provide non-IID token-level bounds?
- Can we construct bounds for models that generate high quality text?
- Can we find more expressive non-linear parameterizations that simultaneously reduce the number of parameters while also including diverse functions which are likely to fit the training data?

Future work

- Can we provide non-IID token-level bounds?
- Can we construct bounds for models that generate high quality text?
- Can we find more expressive non-linear parameterizations that simultaneously reduce the number of parameters while also including diverse functions which are likely to fit the training data?

Thank you! Paper: https://arxiv.org/abs/2312.17173