Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning

Armen Aghajanyan
Facebook AI
armenag@fb.com

Sonal Gupta
Facebook
sonalgupta@fb.com

Luke Zettlemoyer
Facebook AI
University of Washington
lsz@fb.com

Abstract

Although pretrained language models can be fine-tuned to produce state-of-the-art results for a very wide range of language understanding tasks, the dynamics of this process are not well understood, especially in the low data regime. Why can we use relatively vanilla gradient descent algorithms (e.g., without strong regularization) to tune a model with hundreds of millions of parameters on datasets with only hundreds or thousands of labeled examples? In this paper, we argue that analyzing fine-tuning through the lens of intrinsic dimension provides us with empirical and theoretical intuitions to explain this remarkable phenomenon. We empirically show that common pre-trained models have a very low intrinsic dimension; there exists a low dimension reparameterization that is as effective for fine-tuning as the full parameter space. For example, by optimizing only 200 trainable parameters randomly projected back into the full space, we can tune a RoBERTa model to achieve 90% of the full parameter performance levels on MRPC. Furthermore, we empirically show that pre-training implicitly minimizes intrinsic dimension and, perhaps surprisingly, larger models tend to have lower intrinsic dimension after a fixed number of pre-training updates, at least in part explaining their extreme effectiveness. Lastly, we connect intrinsic dimensionality with low dimensional task representations and compression based generalization bounds to provide generalization bounds that are independent of the full parameter count.

1 Introduction

Pre-trained language models (Radford et al., 2019; Devlin et al., 2018; Liu et al., 2019; Lewis et al., 2019, 2020) provide the defacto initialization for modeling most existing NLP tasks. However, the process of fine-tuning them on often very small target task datasets remains somewhat mysterious. Why can we use relatively vanilla gradient descent algorithms (e.g., without strong regularization) to tune a model with hundreds of millions of parameters on datasets with only hundreds or thousands of labeled examples?

We propose intrinsic dimensionality as a new lens through which fine-tuning can be analyzed (Li et al., 2018). An objective function's intrinsic dimensionality describes the minimum dimension needed to solve the optimization problem it defines to some precision level. In the context of pre-trained language models, measuring intrinsic dimensional will tell us how many free parameters are required to closely approximate the optimization problem that is solved while fine-tuning for each end task. For example, we will show that 200 parameters (randomly projected back into the full parameter space) are enough to represent the problem of tuning a RoBERTa model to within 90% of the performance of the full model. More generally, we also describe a set of strong empirical and theoretical connections between intrinsic dimensionality, number of parameters, pre-training, and generalization.

We first empirically show that standard pre-trained models can learn a large set of NLP tasks with very few parameters and that the process of pre-training itself implicitly minimizes the intrinsic dimension of later tuning for different NLP tasks. We study over a dozen different pre-trained models to show that the number of parameters strongly inversely correlates with intrinsic dimensionality, at least in part justifying the extreme effectiveness of such models. We interpret pre-training as providing a framework that learns how to compress the average NLP task. Finally, we connect intrinsic dimensional with low dimensional task representations and compression-based generalization bounds to provide intrinsic-dimension-based generalization bounds independent of the full parameter count, further justifying why these methods generalize so well in practice across tasks.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 7319–7328
August 1–6, 2021.

The contributions of our paper are the following:

- We empirically show that common NLP tasks within the context of pre-trained representations have an intrinsic dimension several orders of magnitudes less than the full parameterization.
- We propose a new interpretation of intrinsic dimension as the downstream fine-tuning task's minimal description length within the framework of the pre-trained model. Within this interpretation, we empirically show that the process of pre-training implicitly optimizes the description length over the average of NLP tasks, without having direct access to those same tasks.
- We measure the intrinsic dimension of a large set of recently developed pre-training method, and how that larger models tend to have smaller intrinsic dimension.
- Lastly, we show that compression based generalization bounds can be applied to our intrinsic dimension framework to provide generalization bounds for large pre-trained models independent of the pre-trained model parameter count.

2 Related Work

Calculating the intrinsic dimension of an objective function in the context of deep-learning was first proposed by Li et al. (2018). They analyzed the impact of various architectures on the intrinsic dimensionality of their objective. Our work is a direct extension of this approach, focusing on analyzing pre-trained representations instead.

There is a large collection of literature analyzing pre-trained models from the perspective of capacity. For example, a recent line of work has shown that pre-trained models such as BERT are redundant in their capacity, allowing for significant sparsification without much degradation in end metrics (Chen et al., 2020; Prasanna et al., 2020; Desai et al., 2019). Houlsby et al. (2019) showed that fine-tuning top layers of pre-trained models is not effective and that alternate methods allow fine-tuning effectively with a couple of percent of the parameters. Furthermore, we can view computing the intrinsic dimensionality as a continuous relaxation of the sparsification problem.

There also exist connections between intrinsic dimensionality, knowledge distillation, and other model compression methods. Fundamentally intrinsic dimensionality attempts to find the smallest set of parameters needed to tune to reach satisfactory solutions, which can be thought of as a sparsification or distillation problem (Hinton et al., 2015; Chen et al., 2020). Unlike distillation approaches, the approach of intrinsic dimensionality does not change parameter count, sparsity, or architecture but instead looks at the underlying rank of the objective function (Li et al., 2018). There are also connections between representing multiple tasks within a pre-trained model and compression which we explore in §5.

Moreover, standard approaches towards fine-tuning seem to have non-trivial effects on the generalization of pre-trained representations (Aghajanyan et al., 2020, 2021). A holistic explanatory picture of the successes of fine-tuning has not yet been painted. A clear understanding of the underlying mechanisms which lead to the incredible generalization of fine-tuned pre-trained representations is currently missing. Moreover, we still do not understand why various pre-training methodology manifests in universally useful representations, although recent line of works have attempted to cover this gap by looking at loss landscapes, and the learned linguistic properties of pre-trained models (Hao et al., 2019; Clark et al., 2019a).

3 Intrinsic Dimensionality of Finetuning

Background The intrinsic dimension of an objective function measures the minimum number of parameters needed to reach satisfactory solutions to the respective objective (Li et al., 2018). Alternatively, the intrinsic dimension represents the lowest dimensional subspace in which one can optimize the original function to within a certain level of approximation error. Computing the exact intrinsic dimensional of the objective function is computation intractable; therefore, we resort to heuristic methods to calculate an upper bound. Let $\theta^D = [\theta_0, \theta_1, ..., \theta_m]$ be a set of D parameters that parameterize some model $f(\cdot, \theta)$. Instead of optimizing the empirical loss in the original parameterization (θ^D), the subspace method fine-tunes the model via the following re-parameterization in the lower-dimensional d-dimensions:

$$\theta^D = \theta_0^D + P(\theta^d) \tag{1}$$

where $P : \mathbb{R}^d \to \mathbb{R}^D$ projects from a parameter from a lower-dimensional d to the higher dimensional D and θ_0^D is the original model parameterization. Intuitively, we project using an arbitrary random projection onto a much smaller space; usually, a linear projection, we then solve the optimization problem in that smaller subspace. If we reach a satisfactory solution, we say the dimensionality of that subspace is the intrinsic dimension. This methodology was proposed in the seminal paper by Li et al. (2018). Concretely Li et al. (2018) proposed three different parameteric forms for P; a random linear dense projection ($\theta^d W$), random linear sparse projection ($\theta^d W_{\text{sparse}}$) and random linear projection via the Fastfood transform (Le et al., 2013).

We will primarily use the Fastfood transform, defined as:

$$\theta^D = \theta_0^D + \theta^d M \qquad M = HG\Pi HB \tag{2}$$

The factorization of M consists of H, a Hadamard matrix, G, a random diagonal matrix with independent standard normal entries, B a random diagonal matrix with equal probability ± 1 entries, and Π a random permutation matrix. Furthermore, the matrix multiplication with a Hadamard matrix can be computed in $\mathcal{O}(D \log d)$ via the Fast Walsh-Hadamard Transform. Everything except θ_d is fixed; therefore, the optimization problem lies only in d-dimensions.[1]

We use the Fastfood transform due to its computational complexity. Specifically, using Hadamard matrices instead of dense matrices allows us to compute a linear projection significantly faster than a dense matrix projection. Furthermore, when working with large models such as RoBERTa, the memory required to store even a low-dimensional dense matrix to calculate intrinsic dimension is unreasonable ($d = 1000, 330, 000, 000 * 1000 * 4$ bytes $= 1.32$ terabytes).

The standard method of measuring the intrinsic dimensionality of an objective as proposed by (Li et al., 2018) requires searching over various d, training using standard SGD over the subspace reparameterization θ^D and selecting the smallest d which provides us with a satisfactory solution (d_{90}). (Li et al., 2018) defined the *satisfactory solution* as being 90% of the full training metric. For example, if we reach 85% accuracy training a model with all of its parameters, the goal is to find the smallest d, which would reach $0.9 * 85\% = 76.5\%$ accuracy; we call this dimension d_{90}.[2]

The way (Li et al., 2018) define a satisfactory solution reduces the dependence of the dataset size on the calculation of intrinsic dimension. For a small dataset, we will generally have worse end metrics; therefore, we have a lower d_{90} cut-off; inversely, a larger dataset will require a more non-trivial d_{90} cut-off.

Structure Aware Intrinsic Dimension Due to the large size of pre-trained language models (generally in the hundreds of millions of parameters), the only computationally reasonable subspace optimization method is one that utilizes the Fastfood transform. For example, if we are interested in subspace training with $d = 1000$ for the RoBERTa-Large model using a dense matrix, we would require 1.42 terabytes of memory to store just the projection matrix.

Unfortunately, the method of finding the intrinsic dimension proposed by (Li et al., 2018) is unaware of the layer-wise structure of the function parameterized by θ. Existing literature argues that in attention-based pre-trained models, individual layers specialize separately (Clark et al., 2019b); therefore, it is useful to incorporate a notion of structure when computing d_{90}. We define Structure-Aware Intrinsic Dimension (SAID) as the following

$$\theta_i^D = \theta_{0,i}^D + \lambda_i P(\theta^{d-m})_i \tag{3}$$

For m layers, we trade m parameters from our subspace parameter θ_d to allow for layer-wise scaling through jointly learned λ, thus θ_d becomes $[\theta_{d-m}, \lambda]$. This allows the SAID method to focus a larger capacity of θ^{d-m} towards specific layers what might carry more relevant information for the task at hand. Conversely, we will refer to the layer unaware method (Equation 2) as the Direct Intrinsic Dimension (DID) method.

4 Intrinsic Dimensionality of Common NLP Tasks

4.1 Sentence Classification

We first empirically calculate the intrinsic dimension of various pre-trained models on a set of sentence prediction tasks from the GLUE Benchmark

[1] If we place a constraint of M being a binary matrix, we recover the sparsification problem; therefore, we can also view finding intrinsic dimensionality as a continuous relaxation of the sparsification problem.

[2] Initializing $\theta^d = 0$ we recover the original parameterization θ_0^D which in the context of fine-tuning represents the original weights of the pre-trained model.

	SAID		DID	
Model	MRPC	QQP	MRPC	QQP
BERT-Base	1608	8030	1861	9295
BERT-Large	1037	1200	2493	1389
RoBERTa-Base	896	896	1000	1389
RoBERTa-Large	**207**	**774**	322	**774**

Table 1: Estimated d_{90} intrinsic dimension computed with SAID and DID for a set of sentence prediction tasks and common pre-trained models.

(Wang et al., 2018). We focus on analyzing BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) at both the base and large model sizes.

We chose to experiment with MRPC (Dolan and Brockett, 2005) and QQP (Iyer et al., 2017) as reference examples of small and large tuning datasets. MRPC is a binary classification task for predicting semantic equivalency for two paraphrases with roughly 3700 training samples, while QQP is a binary classification task for predicting semantic equality of two questions, with roughly 363k samples. For every dataset and every model, we run 100 subspace trainings with d ranging from 10 to 10000 on a log scale. For every training run, we do a small hyperparameter search across four learning rates. We initialize every θ_d to the zero vector to allow for our starting point to be the original pre-trained model. Our subspace optimization method also operates over the randomly initialized sentence classification head to ensure we have exactly d parameters to optimize.

We use both the SAID and DID subspace optimization methods, which we implemented in the Huggingface Transformers library (Wolf et al., 2019). We present the results in Figure 1.

4.2 Analysis

The first takeaway is the incredible low dimensionality of viable solutions. With RoBERTa-Large, we can reach 90% of the full fine-tuning solution of MRPC using roughly 200 parameters and 800 parameters for QQP (Table 1). Recall that our approximation of intrinsic dimension is necessarily crude by using random projections and restricting them to the use of Fastfood transform; therefore, it is likely that the true intrinsic dimension is much lower.

Furthermore, RoBERTa consistently outperforms BERT across various subspace dimensions d while having more parameters. We leave a more in-depth analysis of model parameter size on intrinsic dimensionality to a later section (§5.2).

Lastly, we see that adding a notion of structure in the computation of intrinsic dimension is beneficial with the SAID method consistently improving over the structure unaware DID method.

5 Intrinsic Dimension, Pre-Training, and Generalization Gap

One interpretation of the intrinsic parameter vector is that it encodes the task at hand with respect to the original pre-trained representations. Therefore, we can interpret d as the minimal description length of the task within the framework dictated by the pre-trained representations (Hinton and Zemel, 1993). Under this interpretation of intrinsic dimensionality, we hypothesize that pre-training is implicitly lowering the intrinsic dimensionality of the average NLP task, and therefore compressing the minimal description length of those same tasks.

What do we more precisely mean by intrinsic parameter encoding a task within the framework provided by the pre-trained representations? Traditionally, a finetuned model (e.g. for a classification tasks) simply consists of a classification head g, parameterized by w_g applied to fine-tuned representations f, parameterized by w_f per sample x. Therefore, to fully describe a task, we need to pack together parameterizations and weights $\{g, f, w_g, w_f\}$. This model description is completely decoupled from the original weights of the pre-trained representation w_{f_0}, therefore to represent n classification tasks, we need to maintain $n\,\{w_g, w_f\}$; additionally, the task representation is incredibly high dimensional. Conversely, fine-tuning utilizing SAID in d-dimensions requires storing only θ_d per task, a single random seed used to generate M and the original pre-trained weights w_{f_0}. Therefore, we can represent arbitrary NLP tasks within a single pre-trained model framework with $d + 1$ parameters.

For example, in the last section, we represented MRPC with roughly 200 parameters, which translates to needing less than a kilobyte of data to encode a complex natural language task within the framework provided by RoBERTa.

We hypothesize that the better the pre-trained models are, the fewer bits (description length) are needed to represent the average NLP task, as we will demonstrate empirically in the next section.

Figure 1: Evaluation accuracy on two datasets and four models across a range of dimensions d for the DID method. The horizontal lines in each figure represent the 90% solution of the respective full model.

5.1 Pre-Training Intrinsic Dimension Trajectory

To verify our hypothesis of pre-training optimizing intrinsic dimension, we retrain a RoBERTa-Base from scratch and measure the intrinsic dimension of various NLP tasks at different training checkpoints, using the SAID method. We completely replicate the setting as described by Liu et al. (2019) apart from only training for a total of 200k steps (instead of 500k) with half the batch size (1k). To calculate the intrinsic dimension more efficiently, we reuse the best learning rates discovered in Section 4 for $d < 10000$ and use a fixed learning rate for anything else. To find d_{90} we do a binary search across d per each checkpoint, with a minimum d of 100 and a maximum of 4 million. The "full solution" that we use when deciding d_{90} cutoff is computed by fine-tuning the checkpointed model in the standard way. We compute SAID on six datasets; *MRPC*, *QQP*, *Yelp Polarity* (Zhang et al., 2015), *SST-2* (Socher et al., 2013), *MNLI* (Williams et al., 2018) and *ANLI* using all rounds of data (Nie et al., 2019). Although we focus on bench-marking sentence classification tasks the selected set of tasks contains variety, from sentiment classification (*Yelp Polarity*, *SST-2*) to Natural Language Inference (*MNLI*, *ANLI*) to question similarity (*QQP*).

We present our results in Figure 2. The intrinsic dimensionality of RoBERTa-Base monotonically decreases as we continue pre-training. We do not explicitly optimize for intrinsic dimensionality, specifically during pre-training (the language model does not have access to downstream datasets!), but none-the-less the intrinsic dimension of these downstream tasks continues to decrease.

More so, tasks that are easier to solve consistently show lower intrinsic dimensionality across all checkpoints, for example, *Yelp Polarity* vs. the notoriously tough *ANLI* dataset. The correlation between challenging tasks for RoBERTa and their large intrinsic dimension hints at a connection between generalization and intrinsic dimension. We will discuss generalization further in Section §5.3.

Given our task representation interpretation of intrinsic dimensionality, we argue that the large scale training of Masked Language Models (MLM) learns generic and distributed enough representations to facilitate downstream learning of highly compressed task representations. Furthermore, we argue for another perspective of pre-training learning representations that form a compression framework with respect to various NLP tasks.

5.2 Parameter Count and Intrinsic Dimension

We also measure the relationships between the parameter count of arbitrary pre-trained models and

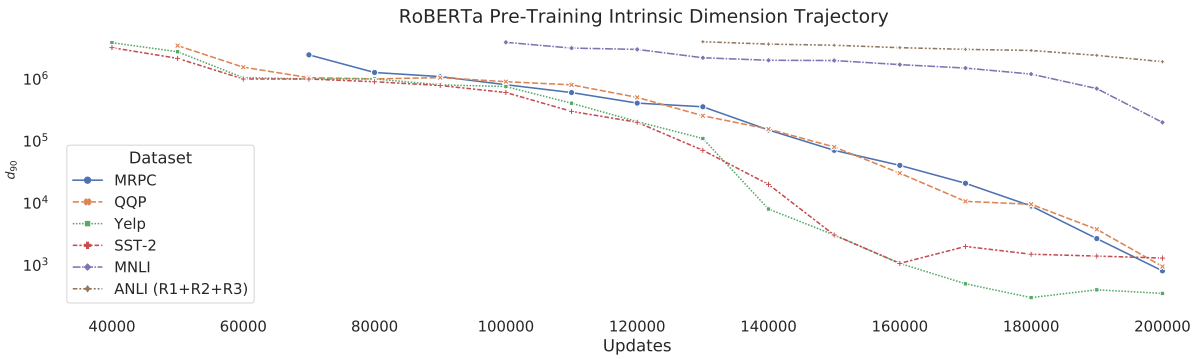

Figure 2: Every 10k updates of RoBERTa-Base that we trained from scratch, we compute d_{90} for six datasets; MRPC, QQP, Yelp Polarity, SST-2, MNLI, and ANLI. If we were unable to compute a d_{90} for a specific checkpoint, we do not plot the point, hence some datasets start at later points. Unable to compute means either we could not fine-tune the full checkpoint to accuracy above majority class or stabilize SAID training.

the intrinsic dimension of downstream NLP tasks. The optimal experiment to run would be to fix the pre-training method, e.g., MLM RoBERTa style, vary the architecture size from small to very big, and compute the intrinsic dimension of a group of tasks at every size of the model. Unfortunately, such an experiment is computationally infeasible due to the need to train many RoBERTa models.

Instead, we do an empirical study of many existing pre-trained models, regardless of the pre-training method. We show that the trend is strong enough to overcome differences in training methodology. We select the following models: BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), BART (Lewis et al., 2019), Electra (Clark et al., 2020), Albert (Lan et al., 2019), XLNet (Yang et al., 2019), T5 (Raffel et al., 2019), and XLM-R (Conneau et al., 2019). Furthermore, we selected various sizes of these models, as available publicly within the HuggingFace Transformers library (Wolf et al., 2019).

We use the MRPC dataset and compute intrinsic dimension for every pre-trained model utilizing the same binary search methodology mentioned in the previous section with additional small hyper-parameter searches across learning rate (due to the wide range of learning rates needed by various models).

We present our results in Figure 3. There is a strong general trend that as the number of parameters increases, the intrinsic dimension of fine-tuning on MRPC decreases. We ran this experiment on other datasets to ensure that this is not an data artifact. Our experiments showed the same trend; we refer to the Appendix for all trends per dataset.

Within the same window of number of parameters, the pre-training methodology becomes more important. For example, in the regime of 10^8 parameters, RoBERTa pre-training dominates similar sized pre-training methods. However, there does not seem to be a method that can overcome the limitations induced by the number of parameters. Interpreting these results through the lens of learning a compression framework for NLP tasks is straightforward; the more parameters we have in the model, the less we need to represent a task.

5.3 Generalization Bounds through Intrinsic Dimension

We have shown strong empirical evidence connecting pre-training, fine-tuning, and intrinsic dimensionality. However, we have yet to argue the connection between intrinsic dimensionality and generalization. Given that we have seen pre-training minimize intrinsic dimension, we hypothesize that generalization improves as the intrinsic dimension decreases.

To do so, we will empirically experiment with the connections between d_{90} and evaluation set performance by looking at various checkpoints from our RoBERTa experiments in Section §5.1. We also plot the relative generalization gap (delta between train time performance and test time performance).

In Figure 4 we plot the evaluation accuracy's achieved by our pre-training experiment in Section §5.1. A lower intrinsic dimension is strongly correlated with better evaluation performance. Additionally we are interested in measuring relative generalization gap ($\frac{acc_{train} - acc_{eval}}{1 - acc_{eval}}$) across intrinsic dimension. We select the training accuracy that provides us with the best evaluation metrics when computing this figure.

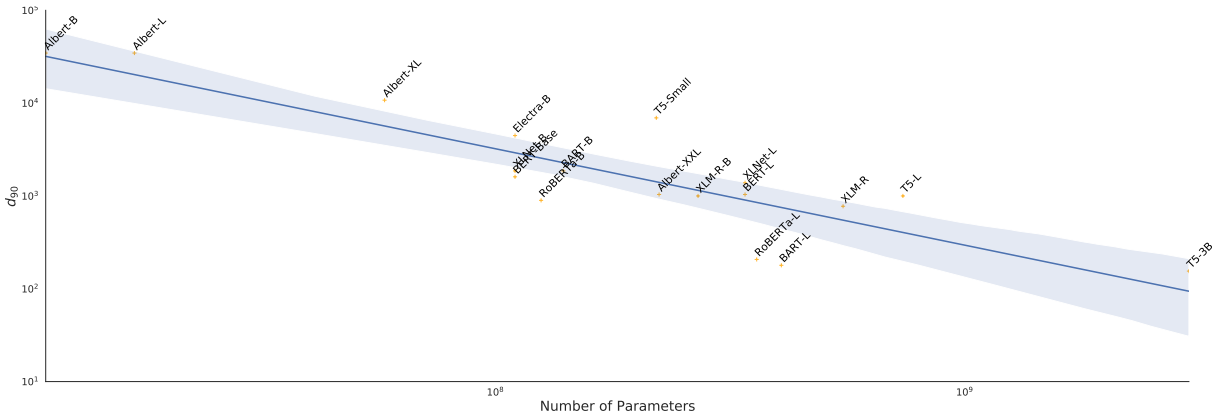

Figure 3: We calculate the intrinsic dimension for a large set of pre-trained models using the SAID method on the MRPC dataset.

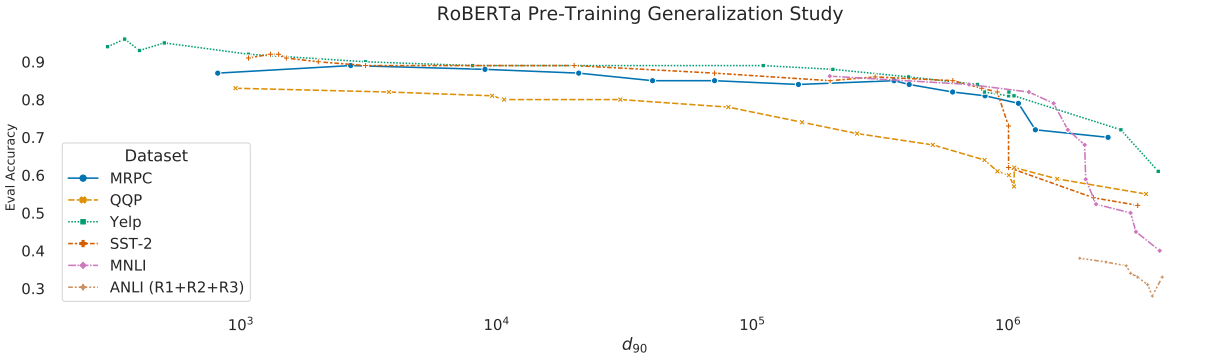

Figure 4: The evaluation accuracy of six datasets across various intrinsic dimensionalities. There is a strong general trend that pre-trained models that are able to attain lower intrinsic dimensions generalize better.

We present our results in Figure 5. Lower intrinsic dimension once again correlates strongly with a smaller relative generalization gap. If we interpret the intrinsic dimension as a measure of complexity, we expect the generalization gap to decrease with intrinsic dimension.

5.4 Generalization Bounds

By applying standard compression based generalization bounds, we can provide theoretical backing to the empirical connection between intrinsic dimension and generalization (Arora et al., 2018).

Consider the following definition of multi-class classification loss with an optional margin over our supervised dataset D.

$$\mathcal{L}_\gamma(f) = \mathbb{P}_{(x,y)\sim D}\left[f(x)[y] \leq \gamma + \max_{j \neq y} f(x)[j]\right]$$

When $\gamma = 0$, $\mathcal{L}_0$ recovers the standard classification loss. Furthermore, Let $\hat{\mathcal{L}}_\gamma(f)$ be an unbiased empirical estimate of the margin loss.

Theorem 1. *Let f be a function which is parameterized by θ^D as described in Equation 1 with a total of d trainable intrinsic parameters on a dataset with m samples. Then with a high probability, we can state the following asymptotic generalization bound*

$$\mathcal{L}_0(f) \leq \hat{\mathcal{L}}_0(f) + \mathcal{O}\left(\sqrt{\frac{d}{m}}\right) \tag{4}$$

Proof. We defer the proof Section §A.1 in the Appendix. We note that this is an extension of the well-known compression based generalization bound (Arora et al., 2018). □

This generalization bound is independent of the underlying parameter count (D) of the pre-trained model but depends on the ability to compress the downstream task (d). Moreover, given that our previous section shows larger models compress better, our bounds are aligned with general intuition and recent empirical evidence that larger pre-trained models generalize better. Explicitly, these bounds only apply to pre-trained methods trained with the intrinsic dimension subspace method; research has yet to show that standard SGD optimizes in this low dimensional space (although experimentally,

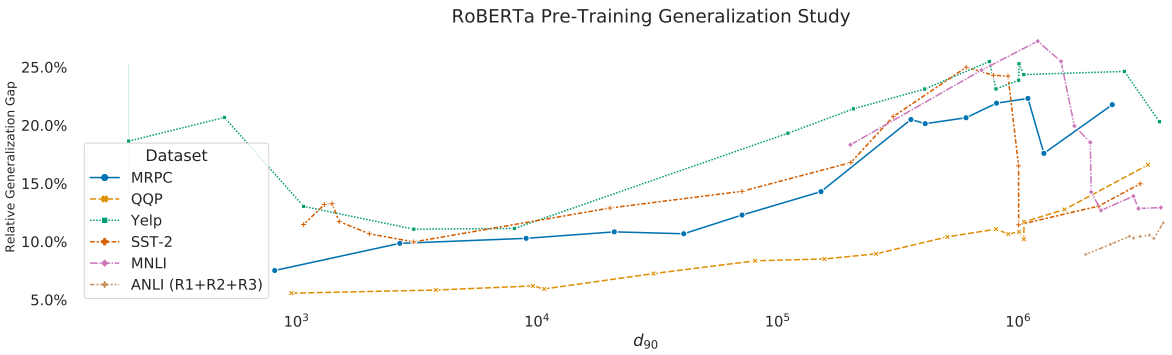

Figure 5: The intrinsic dimension and the respective relative generalization gap across a set of varied tasks.

this seems to be confirmed). We leave the theoretical contribution of showing SGD optimizes in this space, possibly resembling intrinsic subspace, for future work.

We want to highlight that generalization is not necessarily measured by the pre-trained model's parameter count or measure of complexity but the pre-trained model's ability to facilitate the compression of downstream tasks. In some sense, if we want to compress downstream tasks better, we must expect pre-trained representations to have a considerable measure of complexity.

6 Conclusion

In conclusion, we proposed viewing the various phenomena surrounding fine-tuning and pre-training through the lens of intrinsic dimensionality. We empirically showed that common natural language tasks could be learned with very few parameters, sometimes in the order of hundreds, when utilizing pre-trained representations. We provided an interpretation of pre-training as providing a compression framework for minimizing the average description length of natural language tasks and showed that pre-training implicitly minimizes this average description length.

We continued by doing an empirical study of existing pre-training methods and their respective intrinsic dimension, uncovering the phenomena that intrinsic dimensionality decreases as we increase the number of pre-trained representation parameters. This phenomenon provides some intuitions to the trend of growing pre-trained representations. We connected intrinsic dimensionality with generalization by first showing that pre-trained models with lower intrinsic dimensions across various tasks achieve higher evaluation accuracies and lower relative generalization gaps. Furthermore, we explain these empirical results by applying well-known generalization bounds to the intrinsic dimension to get generalization bounds that grow on the order of the intrinsic dimension, not the parameter count.

Intrinsic dimensionality is a useful tool for understanding the complex behavior of large models. We hope that future work will make explicit theoretical connections between SGD and optimizing the intrinsic dimension as well as explain exactly why pre-training methods optimize the intrinsic dimensionailty of tasks before not seen.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. *arXiv preprint arXiv:2101.11038*.

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2020. Better fine-tuning by reducing representational collapse. *arXiv preprint arXiv:2008.03156*.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. 2018. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pre-trained bert networks. *Advances in Neural Information Processing Systems*, 33.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019a. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019b. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training

text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Shrey Desai, Hongyuan Zhan, and Ahmed Aly. 2019. Evaluating lottery tickets under distributional shifts. *arXiv preprint arXiv:1910.12708*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of bert. *arXiv preprint arXiv:1908.05620*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Geoffrey E Hinton and Richard Zemel. 1993. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6:3–10.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *arXiv preprint arXiv:1902.00751*.

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. 2017. First quora dataset release: Question pairs.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Quoc Le, Tamás Sarlós, and Alex Smola. 2013. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When bert plays the lottery, all tickets are winning. *arXiv preprint arXiv:2005.00561*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

A Appendix

A.1 Proofs

Arora et al. (2018) define (γ, S) compressible using helper string s as the following.

Definition 1. *(γ, S) compressible using helper string s*

Suppose $G_{\mathcal{A},s} = \{g_{\theta,s} | \theta \in \mathcal{A}\}$ is a class of classifiers indexed by trainable parameters A and fixed strings s. A classifier f is (γ, S)-compressible with respect to $G_{\mathcal{A}}$ using helper string s if there exists $\theta \in \mathcal{A}$ such that for any $x \in S$, we have for all y

$$|f(x)[y] - g_{\theta,s}(x)[y]| \leq \gamma \tag{5}$$

Remark 1. *If we parameterize $f(x; \theta)$ via the intrinsic dimension approach as defined in Equation 1, then f is compressible losslessly using a helper string consisting of the random seed used to generate the static random projection weights and the initial pre-trained representation θ_0^D. Therefore we say f parameterized by either DID or SAID is $(0, S)$ compressible.*

Theorem 2.1 in (Arora et al., 2018) states given a compression consisting of r discrete states we achieve the following generalization bound.

$$\mathcal{L}_0(f) \leq \hat{\mathcal{L}}_\gamma(f) + O\left(\sqrt{\frac{d \log r}{m}}\right) \tag{6}$$

We can trivially represent our parameters θ_d in a discrete fashion through discretization, as was done in Arora et al. (2018), and the number of states is dependent on the level of quantization but is static once chosen (FP32 vs. FP16).

We then connect the fact that models trained in low dimensional subspace using SAID/DID methods are (0, S)-compressible to derive the final asymptotic bound.

$$\mathcal{L}_0(f) \leq \hat{\mathcal{L}}_0(f) + \mathcal{O}\left(\sqrt{\frac{d}{m}}\right) \tag{7}$$