Measuring Intrinsic Dimension of Token Embeddings

Anonymous ACL submission

Abstract

In this study, we measure the Intrinsic Dimension (ID) of token embedding to estimate the intrstic dimensions of the manifolds spanned by the representations, so as to evaluate their redundancy quantitatively compared to their extrinsic dimensionality. In detail, (1) we estimate the ID of token embeddings in small-scale language models and also modern large language models, finding that the embedding spaces often reside on lower-dimensional manifolds compared to their extrinsic dimensionality; (2) we measure the ID across various model sizes and observe an increase in redundancy rates as the model scale grows; (3) we measure the dynamics of IDs during the training process, and find a rapid ID drop in the early stages of training. Moreover, (4) when LoRA is applied to the embedding layers, we observe a sudden drop in perplexity around the estimated IDs, suggesting that the ID can serve as a useful guideline for LoRA application.

1 Introduction

Recent **L**arge **L**anguage **M**odels (LLMs) utilize token embedding layers with hundreds or even thousands of ***e**xtrinsic **d**imensions* (ED), while it remains unclear how many of these dimensions are actually necessary for effective representation. If the token embedding utilizes only a lower-dimensional manifold, large portions of the parameter space may be redundant, increasing training and inference costs unbeneficially. Also, prior work suggests that **sentence** embeddings can lie on remarkably low-dimensional manifolds (Ueda and Yokoi, 2024), while the sentence embeddings are model outputs, or *activations* that can not be explicitly reduced for a more efficient model.

So, in this paper, we focus on the **token** embedding, which is the model parameters on the first layer of a typical language model, instead of activations. In detail, we examine the **Intrinsic Dimension (ID)** of embedding spaces in both small-scale (e.g., Word2Vec, GloVe) and large-scale (e.g., Pythia) word embedding models, addressing two central research questions:

> **RQ1** How large is the gap between ED and ID, and what factors influence it?
>
> **RQ2** How does the ID in an LLM's embedding layer evolve and stabilize among model scale and training dynamics?

To answer these questions, first, we measure the discrepancy between ED and ID in popular word embedding models (Section 3.2). Next, using Pythia suite (Biderman et al., 2023), we investigate how the dimension redundancy varies against model scales, and how the IDs update among the training dynamics (Sections 3.3 and 3.4). Finally, we show that the estimated ID can guide the selection of the inner dimension in **lo**w-**r**ank **a**daptation (LoRA) (Hu et al., 2022) on the embedding layer, striking a better balance between compactness and performance (Section 3.5).

Contributions. (1) We present a consistent empirical analysis of ID for both small- and large-scale embedding models, demonstrating that embedding spaces remain surprisingly low-dimensional. (2) We reveal that the ID is stabilized in the early training even as the model size grows, indicating that a compact, core representation is learned from the early phase. (3) We provide initial evidence that ID-based rank selection in LoRA delivers efficiency gains without sacrificing perplexity, thereby highlighting the potential of ID-aware compression for large-scale NLP models.

2 Related Works

In recent years, **Intrinsic Dimension (ID)** and **Local Intrinsic Dimensionality (LID)** (Levina and Bickel, 2004; Amsaleg et al., 2015) have gained

attention as indicators of the essential dimensionality of high-dimensional data. Since they capture the nonlinear manifold structure—beyond what linear methods like PCA can reveal—they provide valuable geometric insights into deeper feature representations. Ansuini et al. (2019) observed in the activations of CNNs that: (1) ID is smaller than the Euclidean dimension of each layer, (2) deeper layers tend to have a lower ID, and (3) higher ID often correlates with poorer generalization. For word embeddings, TwoNN (Facco et al., 2017) has shown that ID can compress to around 10 dimensions (Ueda and Yokoi, 2024).

Meanwhile, low-rank approximation techniques such as LoRA (Hu et al., 2022) leverage the low-rank hypothesis to reduce inference and training costs for LLMs. LoRA freezes weights W and learns a low-rank update $\Delta W = AB$ (with $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times k}$), where r governs the compression–expressivity trade-off while drastically reducing trainable parameters. However, how far these representations can be compressed remains unexplored. Understanding this in embedding spaces is essential not only for deepening our grasp of representation learning but also for identifying new directions for model acceleration and memory efficiency.

3 Methodology and Experiments

We begin by describing how we estimate LID and ID, followed by three experiments that apply these methods to token embeddings.

3.1 Method: LID and ID Estimation

Intrinsic Dimension Estimation. Following Levina and Bickel (2004), we estimate the **L**ocal **I**ntrinsic **D**imension (LID) of a point x (e.g. one token embedding vector) via:

$$\widehat{\mathrm{LID}}_k(x) = \left[\frac{1}{k-1}\sum_{i=1}^{k-1} \ln \frac{d_k(x)}{d_i(x)}\right]^{-1}, \quad (1)$$

where $d_i(x)$ is the distance from point x to the i-th of total k nearest neighbor (k is a experiment hyper-parameter). Then, global ID (MacKay and Ghahramani, 2005) is computed as the harmonic mean of the LID across all n embedding vectors:

$$\widehat{\mathrm{ID}} = \left[\frac{1}{n}\sum_{i=1}^{n} \widehat{\mathrm{LID}}_i^{-1}\right]^{-1}. \quad (2)$$

Table 1: Estimated ID.

Statistic	GloVe	FastText	Word2Vec	Random
ID	24.77	13.19	24.75	130.3

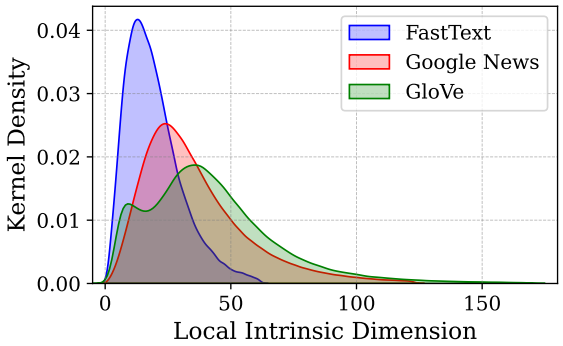

Figure 1: Kernel densities of LID values.

3.2 Experiment 1: ID Estimation for Word Embeddings

Our first experiment evaluates whether widely used pre-trained word embeddings (Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2017)) occupy lower-dimensional manifolds than their ED. The models we use are available via the Gensim library (Řehůřek and Sojka, 2010): `word2vec-google-news-300`, `glove-wiki-gigaword-300`, and `fasttext-wiki-news-subwords-300`. To assess the effect of linguistic structure, we also compare these embeddings to a random baseline consisting of vectors sampled from a normal distribution.

Experimental Procedure. For each embedding vector from the full vocabulary, we use Euclidean distances and FAISS (Douze et al., 2024) to identify the $k = 5$ nearest neighbors for the embedding vector, then compute $\widehat{\mathrm{LID}}_k$, as given in Eq. (1), and finally average these LID values to estimate the global ID following Eq. (2).

Random Baseline. We generate 1e5 points from a d-dimensional Gaussian with mean $\mathbf{0}$ and covariance $\mathbf{I}$, where d is set to equal with the extrinsic dimensionality of each evaluated embedding.

Results. Table 1 shows the ID estimates obtained in this experiment, and Figure 1 illustrates the distribution of LID values. The word embeddings yield IDs of about 10-30, significantly less than the *random* baseline. Given that the ED is 300 for these vectors, the observed ID corresponds to approximately 3-10% of the ED, suggesting strong redundancy in the original embedding dimension.

Table 2: Redundancy Ratio (Redu. (%)) alongside ID and ED for Pythia models with various scales.

Model	Redu. (%)	ID	ED
pythia-14m	72.40	35.33	128
pythia-70m	94.14	29.99	512
pythia-160m	96.49	26.97	768
pythia-410m	97.56	24.95	1024
pythia-1b	98.18	37.23	2048
pythia-1.4b	98.43	32.20	2048
pythia-2.8b	98.66	34.18	2560
pythia-6.9b	98.09	78.30	4096
pythia-12b	97.62	121.82	5120

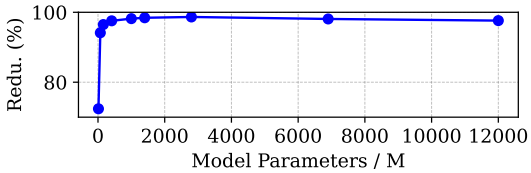

Figure 2: Redundancy Ratio against model parameters.

3.3 Experiment 2: Redundancy Ratio Across Different LLM Scales

Next, we measure the redundancy ratio in the embedding layer of the Pythia series (Biderman et al., 2023) with various scales from 14M to 12B parameters pre-trained under the same training data and conditions, to compare how the ID evolves at different scales under a consistent setting. For each model, let the extrinsic dimension be ED, and let ID be the ID estimated by the method in §3.1, we define the *redundancy ratio* as:

$$\text{Redundancy} = \frac{\text{ED} - \text{ID}}{\text{ED}}, \tag{3}$$

and observe it among various model scales. Unlike §3.2, we focus on this ratio instead of ED, since ED varies across models.

Results. Table. 2 and Figure. 2 present the results of redundancy ratios. As the model size grows, the ID also increases, yet the redundancy ratio remains very high, between roughly 90% and 98%. Moreover, from `pythia-410m` onward, the redundancy ratio stabilizes at around 98%. In other words, for sufficiently large models, the redundancy ratio does not undergo significant change.

3.4 Experiment 3: ID Estimation During LLM Training

To examine how the embedding space of LLMs evolves during training, we utilize the model checkpoints periodically saved along the training dynamics from 1e3 to 1e4 steps at intervals of 1e3, and from 1e4 to 1.43e5 steps at intervals of 5e3. At each checkpoint, we estimate $\widehat{\text{LID}}_k$ using Eq. (1),

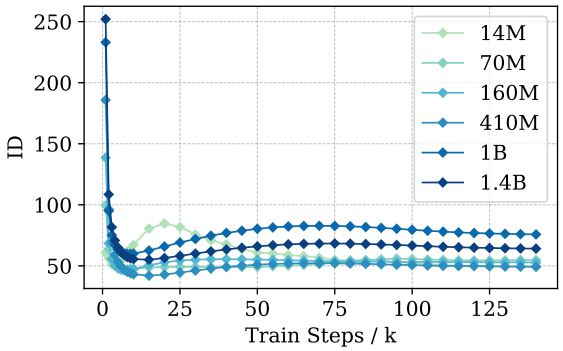

Figure 3: Dynamics of ID against the training steps.

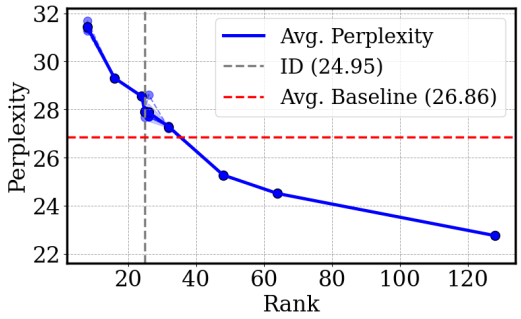

Figure 4: Validation perplexity against LoRA inner dimensions on `pythia-410m`.

$\widehat{\text{ID}}$ using Eq. (2) thereby tracking changes in ID throughout training. Due to limited GPU resources, we restrict our experiments to models ranging from `pythia-14m` to `pythia-1.4b`.

Results. Figure. 3 presents our findings. We observe a sharp decline in ID during the initial training stages, followed by a more gradual convergence. The smallest model, `pythia-14m`, exhibits relatively unstable behavior, which is generally acceptable for smaller-scale models (Tirumala et al., 2022).

3.5 Experiment 4: LoRA with ID-driven Rank Choice

In §3.3, we obtained the ID of each model's embedding layer and used it to guide the rank (inner dimension) selection of LoRA (Hu et al., 2022). Similarly to before, we apply LoRA *only* to the embedding layer (e.g., `gpt_neox.embed_in` in Pythia) for a causal language modeling task on the WikiText-2 dataset, where the dataset is tokenized to a maximum sequence length of 256 and any empty samples are discarded. We systematically vary the LoRA rank {8, 16, 24, 25, 26, 32, 48, 64, 128} around the estimated ID ($\sim$ 24.95), and train only the LoRA parameters on the aforementioned object for 5 epochs with a per-device batch size of 32, and compute the perplexity on the validation set as $\exp(\text{loss})$ to evaluate the effect of

LoRA. This setup allows us to examine how closely the optimal LoRA rank aligns with the ID, as well as whether ranks below or above the ID threshold significantly affect the model's performance.

Results. Figure. 4 presents our findings. In Figure. 4, error bands corresponding to $\pm\sigma$ are displayed. We find that in LoRA, ranks below the ID lead to a clear performance drop, whereas ranks above the ID improve results slightly. Around the ID, performance jumps sharply before declining again, suggesting that ID is pivotal for balancing compactness and capacity in LoRA.

4 Discussions

4.1 RQ1: The Gap between ED and ID is Significant

Word embeddings with an ED of 300 typically exhibit an ID of around 10-30, which aligns with the findings on the sentence embedding (*activation*) of Ueda and Yokoi (2024). It can be inferred that language prior leads the embeddings and also activations to appear more structured and low-ID geometries, compared to random vectors.

Notably, FastText embeddings exhibit a significantly lower ID compared to those from other models. This phenomenon may be attributed to FastText's subword segmentation, with additional contributions potentially coming from factors such as the training data and token frequency. To investigate this, we conducted a preliminary experiment with various tokenizers to assess how different tokenization strategies affect the resulting ID. Details and results are provided in the Appendix A.

4.2 RQ2: Redundancy Ratio Persists at a High Level

In Fig. 2, our scale-based analysis reveals that as the model size grows, the ID also increases but still lags significantly behind the ED, resulting in about a 98% redundancy ratio. This suggests that many dimensions remain underutilized, even though large models offer ample representational capacity. Moreover, high redundancy may, in fact, mirror the inherent complexity of language, providing nuanced flexibility for downstream tasks and cautioning against viewing it as purely inefficiency. In detail, it can be considered that during the fine-tuning onto a downstream task, the model can enable the unused dimensionalities as a "channel" for the related information.

Additionally, §3.4 shows that early training rapidly finds a compact, low-dimensional representation of core linguistic features, followed by a slower phase of refinement.

Possible Explanations for the Rapid Emergence of Low-Dimensional Structure. We conjecture that the embedding layer quickly converges to a low-dimensional manifold due to the overparameterized nature of the model and the intrinsic clustering in natural language. Specifically, during the initial training phase, frequent tokens are rapidly grouped in a semantically meaningful subspace, while infrequent tokens remain scattered around the periphery, effectively reducing the global degrees of freedom. This phenomenon aligns with previous work on Neural Collapse (Gao et al., 2019; Cho et al., 2025) in classification settings, suggesting that early training emphasizes global structure. Moreover, the manifold hypothesis posits that real-world data often lie on a low-dimensional manifold; our ID estimation lends empirical support to this claim in the context of large-scale language models. In later stages of training, ID remains relatively stable, indicating a phase where the primary geometry is refined rather than fundamentally restructured. We believe that additional factors such as learning rate schedules, token frequency distributions (Zipf's law), and subword segmentation might further influence the speed and extent of ID convergence. Future work will include in-depth analyses of these factors and their interplay with optimization dynamics.

5 Conclusion

We have shown that while embeddings in both small and large models nominally span hundreds or thousands of dimensions, their **effective** dimensionality, ID, is remarkably low. Notably, ID emerges early in training and remains far below the ED, leaving significant redundancy. Crucially, these findings inform practical compression strategies such as LoRA, where selecting a rank close to the ID can preserve performance while reducing parameters. In short, the ID-based perspective offers both theoretical insight into LLM embeddings and a concrete path toward more efficient, scalable model deployment.

Future Work. We plan to explore ID in additional layers and architectures, extend our approach to cross-linguistic and diachronic corpora, and further investigate ID-based compression methods to enhance LLM interpretability and performance.

Limitation

One limitation of this study is that it focuses exclusively on the Pythia model, thereby restricting the generalizability of our findings to other architectures. Additionally, due to the practical constraints posed by our available GPU resources, the experimental scale remains somewhat smaller compared to contemporary large-scale language models. Consequently, caution should be exercised when extrapolating these results to larger or more diverse model families.

References

Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E. Houle, Ken ichi Kawarabayashi, and Michael Nett. 2015. Estimating local intrinsic dimensionality. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38.

Alessio Ansuini, Alessandro Laio, Jakob H. Macke, and Davide Zoccolan. 2019. Intrinsic dimension of data representations in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *arXiv preprint arXiv:2304.01373*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Hakaze Cho, Yoshihiro Sakai, Kenshiro Tanaka, Mariko Kato, and Naoya Inoue. 2025. Understanding token probability encoding in output embeddings. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10618–10633, Abu Dhabi, UAE. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *Preprint*, arXiv:2401.08281.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. 2017. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1):12140.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Elizaveta Levina and Peter J. Bickel. 2004. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems*, volume 17, pages 777–784.

David J. C. MacKay and Zoubin Ghahramani. 2005. Comments on 'maximum likelihood estimation of intrinsic dimension' by e. levina and p. bickel (2004).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR 2013)*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *Preprint*, arXiv:2205.10770.

Ryo Ueda and Sho Yokoi. 2024. Measuring the intrinsic dimension of language. In *Proceedings of the 30th Annual Conference of the Association for Natural Language Processing*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Preprint*, arXiv:1509.01626.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `https://radimrehurek.com/gensim/`.

Table 3: ID for Each Tokenizer (ED = 300, Vocab Sample = 10,000). WS indicates whitespace tokenization.

Tokenizer	**ID**
Word2Vec-SentencePiece	24.7846
Word2Vec-BPE	24.7275
Word2Vec-WS	27.0036
FastText-SentencePiece	11.3744
FastText-BPE	11.9805
FastText-WS	10.7492

A Tokenizer Analysis

To examine how subword segmentation, training data, or frequency characteristics might influence the ID, we trained word embeddings using various tokenizers on the AGNews corpus (Zhang et al., 2015). Specifically, we compared SentencePiece, **B**yte-**P**air **E**ncoding (BPE), and whitespace tokenization (WS) under both Word2Vec and FastText frameworks. Table 3 lists the resulting ID values for embeddings with an ED of 300, using a vocabulary sample of 10,000 tokens.

We observe that FastText embeddings generally yield lower ID values than Word2Vec across all tokenizers, suggesting that subword-level modeling may help reduce the intrinsic dimensionality. However, further analysis is needed to confirm whether these differences are indeed due to segmentation approaches, data frequency characteristics, or training hyperparameters.