A Theory for Compressibility of Graph Transformers for Transductive Learning

Hamed Shirzad
University of British Columbia
shirzad@cs.ubc.ca

Honghao Lin
Carnegie Mellon University
honghaol@andrew.cmu.edu

Ameya Velingker
Independent Researcher*
ameyav@gmail.com

Balaji Venkatachalam
Meta*
bave@meta.com

David P. Woodruff
CMU & Google Research
dwoodruf@cs.cmu.edu

Danica J. Sutherland
UBC & Amii
dsuth@cs.ubc.ca

Abstract

Transductive tasks on graphs differ fundamentally from typical supervised machine learning tasks, as the independent and identically distributed (i.i.d.) assumption does not hold among samples. Instead, all train/test/validation samples are present during training, making them more akin to a semi-supervised task. These differences make the analysis of the models substantially different from other models. Recently, Graph Transformers have significantly improved results on these datasets by overcoming long-range dependency problems. However, the quadratic complexity of full Transformers has driven the community to explore more efficient variants, such as those with sparser attention patterns. While the attention matrix has been extensively discussed, the hidden dimension or width of the network has received less attention. In this work, we establish some theoretical bounds on how and under what conditions the hidden dimension of these networks can be compressed. Our results apply to both sparse and dense variants of Graph Transformers.

1 Introduction

Graphs are a versatile data structure that naturally models relations between entities across many domains, such as social networks, citation graphs, and systems and code analysis. One common goal is node-level prediction on a single large graph, i.e., *transductive node classification*. In this scenario, one is typically presented with a semi-supervised task in which some node labels are provided (for training or validation) and others are to be predicted. This setting is challenging, as the different nodes do not obey typical distributional assumptions (e.g., they are not independent), and the classification of one node can affect that of other nodes via the neighborhood structure. Examples of tasks in these settings include identifying malicious users in a social network, predicting protein functionality in a protein-protein interaction network, or categorizing products based on a co-purchase network (Hu et al., 2021; Platonov et al., 2023; Shchur et al., 2018; Leskovec, 2014).

Many variants of Graph Neural Networks (GNNs) are able to tackle transductive tasks on graphs (Kipf and Welling, 2016; Veličković et al., 2018; Hamilton et al., 2017). A recent leap in progress on GNNs has been the use of Transformers to perform message-passing operations in these networks. Transformers (Vaswani et al., 2017) have brought about revolutionary changes in several domains of machine learning, ranging from natural language processing (Vaswani et al., 2017; Devlin et al., 2018; Zaheer et al., 2020) to computer vision (Dosovitskiy et al., 2020) and, more recently, geometric deep learning (Dwivedi and Bresson, 2020; Kreuzer et al., 2021; Ying et al., 2021; Rampášek et al.,

*Work done in part while at Google.

Machine Learning and Compression Workshop at NeurIPS 2024.

2022; Shirzad et al., 2023, 2024; Müller et al., 2023). The use of Transformers on graph data involves attention-based message passing on a certain computational graph; the computational graph can either be a "full" graph (where each node attends to every other node) or a sparse graph. Although the typical choice is a full graph, the resulting quadratic complexity is generally infeasible for sufficiently large graphs. Thus, for applications in which graphs are extremely large, one often uses Transformers on a sparse computation graph. Indeed, many linear sparse or low-rank approximations have been proposed for Transformers on graphs (Shirzad et al., 2023, 2024; Wu et al., 2022; Deng et al., 2024).

A significant body of research on scaling Transformers focuses on how a Transformer model can be sparsified or how low-rank approximations of the attention matrix can be calculated. However, an often overlooked aspect in these calculations is determining how large the hidden dimension needs to be, or, alternatively, how much this hidden dimension can be reduced. A Transformer with a hidden dimension D operating on a graph with n nodes and with an attention pattern consisting of m attention edges (e.g., $m = \Theta(n^2)$ in a full-Transformer), has a computational complexity of $\mathcal{O}(nD^2 + mD)$. In most previous works, D is typically considered a constant and therefore dominated by m and n. However, there are a number of theoretical works on GNNs show that, in order to achieve certain properties (e.g., expressivity), the hidden dimension often needs to depend on the graph size (i.e., D is superconstant) (Sanford et al., 2024c,b,a; Loukas, 2019). One notable exception is the work of Shirzad et al. (2024), in which a low-width network is first used to estimate attention scores, after which the estimates are used to sparsify the network and train a larger model with the computed sparsity pattern; that work, however, is primarily empirical.

In this work, we address the aforementioned shortcomings and theoretically analyze the compressibility of a single-head Transformer model. Assuming a Graph Transformer is trained with some hidden dimension D, we seek to determine some bounds showing how small a compressed network can be while approximately preserving the outputs of the original network. In particular, we provide a series of results showing that the hidden dimension can be compressed substantially while preserving the output of the original network up to an additive error of $\mathcal{O}(\epsilon)$ and also preserving attention scores of the original network up to a $1 \pm \mathcal{O}(\epsilon)$ multiplicative approximation factor (for arbitrarily small ϵ). We note that almost all of our results apply irrespective of the attention pattern, which can range from the sparsity pattern of the underlying graph structure of the data to a dense pattern, or even something in between, such as the patterns used by (Shirzad et al., 2023). Furthermore, we complement our theoretical results with empirical results showing the existence of small networks with results that are competitive with those of a given large network.

2 Notation & Assumptions

Graphs Graphs are data structures consisting of a set of nodes V, and a set of edges E. Each edge in E is an ordered pair of nodes. Undirected edges can be represented as two directed edges. Nodes typically have associated features represented by a matrix $\mathbf{X} \in \mathbb{R}^{d_{in} \times n}$, where d_{in} is the dimension of the input features for each node.

We decouple the attention pattern from the graph structure and use m to denote the number of attention edges. In a full-Transformer, $m = n^2$. If we apply attention only over the graph edges, which makes it very similar to message-passing-based networks, m would be the number of graph edges. We use $\mathcal{N}(v)$ to denote the set of neighbors of node v in the attention graph.

Transformer Formulation With some slight simplifications, we can formulate the ℓth layer as:

$$
\begin{aligned}
&\mathbf{V}^{(\ell)} = \mathbf{W}_V^{(\ell)} h^{(\ell)} \qquad \mathbf{Q}^{(\ell)} = \mathbf{W}_Q^{(\ell)} h^{(\ell)} \qquad \mathbf{K}^{(\ell)} = \mathbf{W}_K^{(\ell)} h^{(\ell)} \\
&a_{ij}^{(l)} = \frac{\exp\left(\mathbf{K}_j^{(\ell)} \cdot \mathbf{Q}_i^{(\ell)}\right)}{\sum_{u \in \mathcal{N}_H(i)} \exp\left(\mathbf{K}_u^{(\ell)} \cdot \mathbf{Q}_i^{(\ell)}\right)}, \\
&h_i^{(\ell+1/2)} = \sum_{j \in \mathcal{N}(i)} a_{ij}^{(l)} \mathbf{V}_j^{(\ell)}, \\
&h_i^{(\ell+3/4)} = \sigma\left(\mathbf{W}_1^{(\ell)}\left(h_i^{(\ell+1/2)}\right)\right), \\
&h_i^{(\ell+1)} = \mathbf{W}_2^{(\ell)} h_i^{(\ell+3/4)}.
\end{aligned}
$$

Here, $h_i^{(\ell)}$ is the output of the layer $\ell - 1$ for node i. $\mathbf{W}_V^{(\ell)}, \mathbf{W}_Q^{(\ell)}, \mathbf{W}_K^{(\ell)}, \mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{D \times D}$, except for the first layer where $\mathbf{W}_V^{(0)}, \mathbf{W}_Q^{(0)}, \mathbf{W}_K^{(0)} \in \mathbb{R}^{D \times d_{in}}$. σ can be any 1-Lipchitz elementwise activation function, such as ReLU. The standard self-attention formulation uses a $\frac{1}{\sqrt{D}}$ normalization, but we ignore this since it can be combined with $\mathbf{W}_Q$ and $\mathbf{W}_K$ matrices. To simplify the notation, we define $\mathbf{H}^{(0)} := \mathbf{X}$. We use $\mathcal{T}(X)$ to refer to the large network trained with hidden dimension D and has L layers in total. d always refers to a compressed network dimension size.

Other Notation We use $\|x\|$ to show the norm of a vector x, and the norms on vectors are 2-norms unless stated otherwise. We use capital bold letters for matrices and small letters with indices to refer to the columns of these matrices, e.g., $h_i^{(\ell)}$ is the column i of matrix $\mathbf{H}^{(\ell)}$. Table 2 gives more.

Assumptions We remove the normalization parts from the architecture but assume that in all input of the Transformer layers, $\|x_i\|_2, \|h_i^{(\ell)}\|_2 \leq \sqrt{\alpha}$, and all linear mapping $\mathbf{W}_\cdot$ matrices' operator norm is bounded by a constant β. Take note that in real Transformers, there is a `LayerNorm` between the layers. The default node feature normalization function in both PyTorch-Geometric and DGL libraries, two of the most standard libraries for graph processing in PyTorch, normalize each node feature to $\|x_i\|_1 = 1$ (Fey and Lenssen, 2019; Wang et al., 2019). Regarding the operator norm of the matrices, we also know that their operator norm is around two under standard initialization. If their operator norm does not change significantly during training, this assumption remains valid.

Graph Transformers are typically fairly shallow; for example, Exphormer and Spexphormer usually employ networks with two to four layers (Shirzad et al., 2023, 2024), while Nodeformer, Difformer, and SGFormer use at most three layers (Wu et al., 2022, 2023, 2024). We thus assume $L = \mathcal{O}(1)$.

3 Reducing the Attention Calculation Complexity Using JLT

The attention calculation in practice is the most computationally expensive part of the model, this is because usually $m = \omega(n)$, and in full-Transformers $m = n^2$. Also, n is often much larger than the usual context length in natural language processing tasks, making the attention calculation part for Graph Transformers very costly. The attention part is the only part that depends on m, the other parts only scale with n. In this section, we will show that $\mathbf{W}_Q, \mathbf{W}_K$ can be compressed into an $\mathbb{R}^{D \times d}$ or $\mathbb{R}^{d_{in} \times d}$ matrix for $d = \mathcal{O}(\frac{\log n}{\varepsilon^2})$ with a bound of $\mathcal{O}(\varepsilon)$ error on the output. This dimension reduction effectively reduces the computational complexity of the attention calculation part from $\mathcal{O}(mD)$ to $\mathcal{O}(md)$. A version of this result also appeared in our recent work (Shirzad et al., 2024).

The Johnson-Lindenstrauss Lemma (Johnson, 1984) is a powerful tool in theoretical computer science that helps preserve the pairwise distance between high-dimensional encodings when mapping them to a lower dimension. Under certain conditions, this pairwise distance preservation can also be applied to preserve the pairwise dot product. This dot product is present in the attention mechanism, allowing us to apply it there to compress the network.

Lemma 3.1 (Johnson-Lindenstrauss Transform Lemma, *JLT*)**.** *Assume $0 < \epsilon, \delta < \frac{1}{2}$ and any positive integer D, if $d = \mathcal{O}(\frac{\log(1/\delta)}{\epsilon^2})$, there exists a distribution over matrices $\mathbf{M} \in \mathbb{R}^{d \times D}$ that for any $x \in \mathbb{R}^D$ and $\|x\| = 1$,*

$$\Pr(\|\mathbf{M}x\| - 1 > \epsilon) < \delta.$$

From this lemma, the dot product version can be derived:

Corollary 3.2 (JLT-dot product)**.** *Assume $0 < \epsilon, \delta < \frac{1}{2}$ and any positive integer D, if $d = \mathcal{O}(\frac{\log(1/\delta)}{\varepsilon^2})$, there exists a distribution over matrices $\mathbf{M} \in \mathbb{R}^{d \times D}$ that for any $x, y \in \mathbb{R}^D$, and $\|x\|, \|y\| \leq \sqrt{\gamma}$,*

$$\Pr((1 - \varepsilon\gamma)x^\mathsf{T}y < x^\mathsf{T}\mathbf{M}^\mathsf{T}\mathbf{M}y < (1 + \varepsilon\gamma)x^\mathsf{T}y) < \delta.$$

For a proof see e.g. Kakade and Shakhnarovich (2009, Corollary 2.1). As a result of this corollary, if we have m pairs of vectors (x_i, y_i), and for each pair i, $\|x_i\|_2, \|y_i\|_2 \leq \sqrt{\gamma}$, and $d = \mathcal{O}(\frac{\log(m)}{\varepsilon^2})$, there exists an $\mathbf{M}$ such that for all these pairs $|x_i^\mathsf{T}\mathbf{M}^\mathsf{T}\mathbf{M}y_i - x_i^\mathsf{T}y_i| < \varepsilon\gamma$. The proof can be done using a union bound over the error from Corollary 3.2. Also, in our case where m is the number of edges, we know that $m \leq n^2$, thus we can also say $d = \mathcal{O}(\frac{\log(n)}{\varepsilon^2})$. Using this result, we prove the

following theorem about Graph Transformers. For this theorem and all upcoming proofs, due to lack of space, we defer the proofs to the Appendix C. Please check Appendix C.1 for the proof of this theorem. In all proofs, we also prove that the attention scores from the compressed network are close to those of the reference network. This is mainly to show the consistency of the results with Shirzad et al. (2024) and the consistent explainability of the model through the attention scores.

Theorem 3.3. *There exists a Transformer $\widehat{\mathcal{T}}$, that for any layer $\mathbf{W}_Q$ and $\mathbf{W}_K$ are in $\mathbb{R}^{d\times D}$ for a $d = \mathcal{O}(\frac{\log n}{\varepsilon^2})$, with a sufficiently small ε, and for all $i \in [n]$, $\|\mathcal{T}(X)_i - \widehat{\mathcal{T}}(X)_i\|_2 = \mathcal{O}(\varepsilon)$. Furthermore, for any attention score, $a_{ij}^{(\ell)} / \widehat{a}_{ij}^{(\ell)} = 1 + \mathcal{O}(\varepsilon)$.*

The asymptotic bounds on the Johnson-Lindenstrauss Lemma (JLT) are tight (Burr et al., 2018), and thus, without any extra assumptions, further compression is not feasible beyond some constant factors. However, in real-world graphs, the columns of $\mathbf{X}$ are typically not n distinct vectors, and many vectors may be equal or very similar to each other. If we have κ unique vectors in the first layer, the complexity for d can be reduced to $\mathcal{O}(\frac{\log \kappa}{\varepsilon^2})$. Also, in many datasets, d_{in} is small and we will see that if $\operatorname{rank}(\mathbf{H}^{(\ell)}) \leq d$, means that we can reduce the hidden dimensions of the attention calculation part to d. In the upcoming layers, for example in homophilic graphs, we can expect neighboring nodes to have very similar embeddings, and in these scenarios, we can expect more compression. Even in heterophilous graphs, if the neighborhood of nodes from the same class has a small diversity, we can expect very similar embeddings. We will investigate this further in the following sections.

4 Exploring Low-rank Assumptions

In many variants of GNNs, we observe that node embeddings rapidly lose rank and converge to a small number of possible embeddings. This convergence can occur in either a beneficial or detrimental manner. Many GNNs suffer from oversmoothing problems, where all node embeddings converge to a single embedding (Oono and Suzuki, 2019; Nt and Maehara, 2019). However, there are also many beneficial scenarios where low-rank embeddings emerge, such as in graph coarsening/pooling methods and when embeddings for nodes from the same class converge to the same encodings. In many of these scenarios, an exact low-rank structure is rare, but embeddings that are *nearly* low-rank are highly plausible.

These phenomena apply equally to the attention mechanism here, and so expecting approximately low-rank embeddings is reasonable. Additionally, in many datasets, the input matrix $\mathbf{X}$ is actually of very low dimension: for example, the node features have size ≤ 10 in the ogbn-proteins, Tolokers, and Minesweeper datasets (Hu et al., 2021; Platonov et al., 2023), and thus all $\mathbf{Q}, \mathbf{K}$, and $\mathbf{V}$ matrices in the first layer have ranks no more than 10.

4.1 Low-rank Embeddings

First, we show that if the input vectors are low-rank and the rank after each activation function is still small, we can at least compress to the maximum rank of the embeddings' mapping after the activation functions. The proof is in Appendix C.2.

Proposition 4.1. *Assume for inputs X and for each ℓ $H^{(\ell+3/4)}$, we have $\operatorname{rank}(X) \leq d$ and for all ℓ, $\operatorname{rank}(\mathbf{H}^{(\ell+3/4)}) \leq d$. There exists a Transformer $\widehat{\mathcal{T}}$, of width d, $\mathcal{T}(X)_i = \mathbf{U}_{out}\widehat{\mathcal{T}}(X)_i$, for some $\mathbf{U} \in \mathbb{R}^{D\times d}$. Furthermore, for any attention score $a_{ij}^{(\ell)} = \widehat{a}_{ij}^{(\ell)}$.*

In this theorem, unlike the others, everything is perfectly reconstructible with no error propagation. However, this assumption is extremely strong; we will thus explore more realistic assumptions.

4.2 Almost Low-rank Embeddings

A more realistic assumption is that the embeddings are not exactly low-rank, but there is low-rank matrix which is column-wise close. With this assumption, we can reduce all the dimensions, except for the activation function part, to dimensions of d.

Theorem 4.2. *Assume for X and $H^{(\ell+3/4)}$ for each ℓ, we have $\bar{X}$ and $\bar{\mathbf{H}}^{(\ell+3/4)}$ such that for each $i \in [n]$, $\|x_i - \bar{x}_i\| \leq \varepsilon$, $\|h_i^{(\ell+3/4)} - \bar{h}_i^{(\ell+3/4)}\| \leq \varepsilon$, $\operatorname{rank}(\bar{\mathbf{X}}) \leq d$ and for all ℓ, $\operatorname{rank}(\bar{\mathbf{H}}^{(\ell+3/4)}) \leq d$.*

Table 1: Average operator norm of the linear mappings in the network and average norm of the input vectors of the Transformer layers from reference large networks. All numbers are average $\pm$ std.

Dataset	Tolokers	Minesweeper	Photo
Operator Norm Average	2.83 ± 0.13	2.62 ± 0.07	2.15 ± 0.06
Vector Norm Average	3.71 ± 0.10	3.44 ± 0.21	3.64 ± 0.23

There exists a Transformer $\widehat{\mathcal{T}}$, which in each layer $\widehat{\mathbf{W}}_V, \widehat{\mathbf{W}}_Q$, and $\widehat{\mathbf{W}}_Q \in \mathbb{R}^{d\times d}$ and $W_1 \in \mathbb{R}^{d\times D}$ and $W_2 \in \mathbb{R}^{D\times d}$, with a sufficiently small ε, and for all $i \in [n]$, $\|\mathcal{T}(X)_i - \widehat{\mathcal{T}}(X)_i\|_2 = \mathcal{O}(\varepsilon)$. Furthermore, for any attention score, $a_{ij}^{(\ell)} / \widehat{a}_{ij}^{(\ell)} = 1 + \mathcal{O}(\varepsilon)$.

The proof for this theorem can be found in Appendix C.3. With this compression we do not have any $D \times D$ matrices, but there are a few size-D embeddings appearing after the activation function. Similar approaches used in previous proofs will not help reduce this dimension. A brief reasoning and a counter-example explaining why this approach will not work are given in Appendix C.3.1.

If we relax the expectation of having *all* nodes within the $\mathcal{O}(\varepsilon)$ error, however, we can have *most* nodes within the error bound by leverage score sampling. This gives at least 99% chance for each node to be within the correct boundaries, but it cannot guarantee for *all* nodes.

Proposition 4.3. *Suppose that $\mathbf{H} \in \mathbb{R}^{D\times n}$ has a $\operatorname{rank}(d)$-approximation $\mathbf{H}'$ where for $i \in [n]$ we have $\|h_i - h_i'\|_2 \le \varepsilon$. Then there exists a row selection matrix $\mathbf{S} \in \mathbb{R}^{k\times D}$ with a matrix $\mathbf{U} \in \mathbb{R}^{D\times k}$ where $k = O(d \log d)$ such that for at least 0.99-fraction of $i \in [n]$ we have*

$$\|\mathbf{U}\mathbf{S}h_i - h_i\|_2 \le \mathcal{O}(\varepsilon) \ .$$

The proof for this proposition is in Appendix C.4. Because the selection can be passed through the activation function, this can be combined with the activation function to reduce the dimension. However, if a node is not within the $\mathcal{O}(\varepsilon)$ error, that means that all the nodes attending to it may not have similar attention as does the large network; without further analysis, this then loses the guarantee for all nodes in the L-hop neighborhood of this node.

4.3 Low-rank with Clustering Assumptions

If instead of assuming the embeddings have a low-rank estimate, we assume they exactly can be clustered around maximum d well-separated centers after each attention pooling operation, we can have a model with all linear mappings in $\mathbb{R}^{d\times d}$. A more formal theorem based on this idea is:

Theorem 4.4. *Assume the activation function in $\mathcal{T}$ is ReLU, and in each layer after the attention operation we can cluster the vectors in $\mathbf{H}^{(\ell+1/2)}$ into at most d clusters with centers $c_1^{(\ell)}, \cdots, c_d^{(\ell)}$ with the condition that $0 < \gamma_1 < \|c_a^{(\ell)}\| < \gamma_2$, where γ_1 and γ_2 are constants, and for each i there exist a $c_a^{(\ell)}$ that $\|h_i^{(\ell+1/2)} - c_a^{(\ell)}\| \le \varepsilon \|c_a^{(\ell)}\|$ for a sufficiently small ε and the clusters are well-separated in a way that for each two clusters $a \ne b$, $c_a^{(\ell)} \cdot c_b^{(\ell)} < \gamma_1^2/2$. If either*

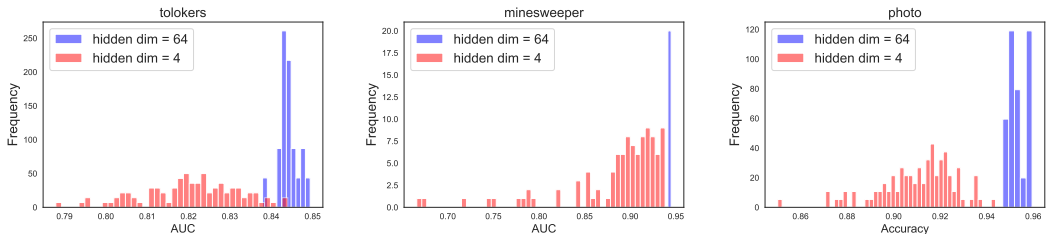
1. *$d \ge c\frac{\log n}{\varepsilon^2}$, for a constant c independent of the problem parameters, or*
2. *there exists some approximation $\bar{X}$ such that $\operatorname{rank}(\bar{X}) \le d$ and for each i, $\|X_i - \bar{X}_i\| \le \varepsilon$,*

then there exists a Transformer $\widehat{\mathcal{T}}$, of width d, with a sufficiently small ε, and for all $i \in [n]$, $\|\mathcal{T}(X)_i - \widehat{\mathcal{T}}(X)_i\|_2 = \mathcal{O}(\varepsilon)$. Furthermore, for any attention score, $a_{ij}^{(\ell)} / \widehat{a}_{ij}^{(\ell)} = 1 + \mathcal{O}(\varepsilon)$.

The proof for this theorem can be found in Appendix C.5. The main idea of the proof is based on separating the nodes from different clusters using d linear mappings in the activation function layers.

5 Experiments

In this section, we experimentally investigate whether models with significantly smaller dimensions can achieve high-quality results, and whether any small hidden dimension network can perform



(a) **(b)** **(c)**

Figure 1: Comparison of the results from a relatively large network with hidden dimension 64 and a small network with hidden dimension 4.

competitively with larger networks. While much of the theory in this work is constructive, some steps require extensive trial and error. To explore this question, we conduct the following experiment instead of directly compressing the actual network: we train a very small network with a hidden dimension of 4, using 100 different initializations. The Tolokers and Minesweeper datasets have very low homophily scores, but their input dimensions are also very low: ten for Tolokers and seven for Minesweeper. On the other hand, the Photo dataset has input node features of size 745 but exhibits a very high homophily score.

In Figure 1, we compare the distribution of results from a relatively large network with a hidden dimension of 64 to that of the small network. As expected, gradient descent performs worse on average for the smaller network. However, on each dataset, the maximum test AUC or accuracy closely approaches the results of the large network, indicating the existence of a compressed network with competitive performance. Nevertheless, exploring initializations is impractical for large real-world datasets, emphasizing the need to learn to compress large models and leaving the investigation of practical compression algorithms as future work.

Furthermore, we measure the operator norm and vector norms in the large trained model to validate some assumptions in Section 2. The results, provided in Table 1, confirm that these norms are reasonably small. More details about the datasets, experimental setup, hyperparameters, and baseline comparisons can be found in Appendix D.

6 Conclusion, Limitations & Future Work

In this work, we have analyzed the compressibility of Graph Transformers under several assumptions. We have showed that under mild assumptions, the hidden dimension of the underlying attention calculation can be reduced to a level that is logarithmic in the number of nodes, and the low-rank approximation of matrices can also lead to compression of the model with minimal losses.

Although we have proved the existence of a compressed network in many scenarios, this does not imply that training with a gradient-based algorithm will necessarily lead to the introduced weights, but this gives at least the guarantee that such a network exists. However, if we can train the large model, many proofs in this work have been constructive, and even for existence proofs such as JLT-lemma, random creation of matrices would have a fair chance to be a valid result, and thus trial and error is possible with these works. However, we keep the experiments on these setups and develop efficient algorithms for compressing based on these theorems for future work.

The results in this work could potentially extend to other architectures such as low-rank attention methods and other GNN variants, opening avenues for future work. Additionally, while this work has primarily focused on scenarios where network compression is possible, impossibility results for compression have not been thoroughly explored. Finding tighter bounds on the hidden dimension of the compressed network would be an interesting problem for future work.

Acknowledgments and Disclosure of Funding

This work was supported in part by the Natural Sciences and Engineering Resource Council of Canada, the Fonds de Recherche du Québec - Nature et technologies (under grant ALLRP-57708-2022), the Canada CIFAR AI Chairs program, the BC DRI Group, Calcul Québec, Compute Ontario,

and the Digital Resource Alliance of Canada. Honghao Lin was supported in part by a Simons Investigator Award, NSF CCF-2335412, and a CMU Paul and James Wang Sercomm Presidential Graduate Fellowship.

References

Burr, M., Gao, S., and Knoll, F. (2018). Optimal bounds for johnson-lindenstrauss transformations. *Journal of Machine Learning Research*, 19(73):1–22.

Chen, X. and Price, E. (2019). Active regression via linear-sample sparsification. In Beygelzimer, A. and Hsu, D., editors, *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 663–695. PMLR.

Deng, C., Yue, Z., and Zhang, Z. (2024). Polynormer: Polynomial-expressive graph transformer in linear time. *arXiv preprint arXiv:2403.01232*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, Y., Cordonnier, J.-B., and Loukas, A. (2021). Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, pages 2793–2803. PMLR.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dwivedi, V. P. and Bresson, X. (2020). A generalization of transformer networks to graphs. *CoRR*, abs/2012.09699.

Fey, M. and Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., and Leskovec, J. (2021). OGB-LSC: A large-scale challenge for machine learning on graphs. *CoRR*, abs/2103.09430.

Johnson, W. B. (1984). Extensions of lipshitz mapping into hilbert space. In *Conference modern analysis and probability, 1984*, pages 189–206.

Kakade, S. and Shakhnarovich, G. (2009). Lecture notes in large scale learning. `https://home.ttic.edu/~gregory/courses/LargeScaleLearning/lectures/jl.pdf`.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kreuzer, D., Beaini, D., Hamilton, W. L., Létourneau, V., and Tossou, P. (2021). Rethinking graph transformers with spectral attention. *arXiv preprint arXiv:2106.03893*.

Leskovec, J. (2014). Snap datasets: Stanford large network dataset collection. *Retrieved December 2021 from http://snap. stanford. edu/data*.

Likhosherstov, V., Choromanski, K., and Weller, A. (2021). On the expressive power of self-attention matrices. *arXiv preprint arXiv:2106.03764*.

Liu, C., Zhan, Y., Wu, J., Li, C., Du, B., Hu, W., Liu, T., and Tao, D. (2022). Graph pooling for graph neural networks: Progress, challenges, and opportunities. *arXiv preprint arXiv:2204.07321*.

Loshchilov, I. and Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Loshchilov, I., Hutter, F., et al. (2017). Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5.

Loukas, A. (2019). What graph neural networks cannot learn: depth vs width. *arXiv preprint arXiv:1907.03199*.

McAuley, J., Targett, C., Shi, Q., and Van Den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52.

Müller, L., Galkin, M., Morris, C., and Rampášek, L. (2023). Attending to graph transformers. *arXiv preprint arXiv:2302.04181*.

Musco, C., Musco, C., Woodruff, D. P., and Yasuda, T. (2022). Active linear regression for ℓ_p norms and beyond. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 744–753. IEEE.

Nt, H. and Maehara, T. (2019). Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*.

Oono, K. and Suzuki, T. (2019). Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*.

Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., and Prokhorenkova, L. (2023). A critical look at the evaluation of gnns under heterophily: Are we really making progress? *arXiv preprint arXiv:2302.11640*.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. (2022). Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515.

Roth, A., Bause, F., Kriege, N. M., and Liebig, T. (2024). Preventing representational rank collapse in mpnns by splitting the computational graph. *arXiv preprint arXiv:2409.11504*.

Sanford, C., Fatemi, B., Hall, E., Tsitsulin, A., Kazemi, M., Halcrow, J., Perozzi, B., and Mirrokni, V. (2024a). Understanding transformer reasoning capabilities via graph algorithms. *arXiv preprint arXiv:2405.18512*.

Sanford, C., Hsu, D., and Telgarsky, M. (2024b). Transformers, parallel computation, and logarithmic depth. *arXiv preprint arXiv:2402.09268*.

Sanford, C., Hsu, D. J., and Telgarsky, M. (2024c). Representational strengths and limitations of transformers. *Advances in Neural Information Processing Systems*, 36.

Shchur, O., Mumme, M., Bojchevski, A., and Günnemann, S. (2018). Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.

Shirzad, H., Velingker, A., Venkatachalam, B., Sutherland, D. J., and Sinop, A. K. (2023). Exphormer: Sparse transformers for graphs. In *ICML*.

Shirzad, H., Venkatachalam, B., Velingker, A., Woodruff, D. P., and Sutherland, D. J. (2024). Even sparser graph transformers. In *NeurIPS*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*, pages 5998–6008.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2018). Graph attention networks. In *ICLR*.

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., et al. (2019). Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. (2020). Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Woodruff, D. P. (2014). Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. (2019). Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR.

Wu, Q., Yang, C., Zhao, W., He, Y., Wipf, D., and Yan, J. (2023). Difformer: Scalable (graph) transformers induced by energy constrained diffusion. *arXiv preprint arXiv:2301.09474*.

Wu, Q., Zhao, W., Li, Z., Wipf, D. P., and Yan, J. (2022). Nodeformer: A scalable graph structure learning transformer for node classification. *NeurIPS*, 35:27387–27401.

Wu, Q., Zhao, W., Yang, C., Zhang, H., Nie, F., Jiang, H., Bian, Y., and Yan, J. (2024). Simplifying and empowering transformers for large-graph representations. *Advances in Neural Information Processing Systems*, 36.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. (2021). Do transformers really perform bad for graph representation? *ArXiv*, abs/2106.05234.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020). Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

A Notation Table

Table 2: A summary of the notations used in this paper. The hat notation always refers to a compressed network equivalent of a vector or matrix from the reference network.

Notation	Definition
n	The number of nodes in the graph
m	The number of attention edges in total, including graph and expander edges
d	Hidden dimension of a narrow network
D	Hidden dimension of the original large graph
L	The total number of layers in the network
ℓ	Arbitrary layer index
$\mathbf{V}$	Value mapping of the vectors in the attention mechanism
$\mathbf{Q}$	Query mapping of the vectors in the attention mechanism
$\mathbf{K}$	Key mapping of the vectors in the attention mechanism
$\mathbf{W}_\cdot^{(\ell)}$	Weight matrix of mapping such as key, query, value, edge features, or bias in layer ℓ
$\widehat{\mathbf{W}}_\cdot^{(\ell)}$	Low dimensional network's weight matrix for a mapping in layer ℓ
$\mathbf{M}_\cdot$	A linear mapping matrix (usually from the higher dimension to the smaller)
ReLU	Rectified Linear Unit
$\mathbf{H}^{(\ell)}$	Output of layer $\ell-1$ from the reference network
$\bar{\mathbf{H}}^{(\ell)}$	A low-rank estimation of $\mathbf{H}^{(\ell)}$
$\widehat{\mathbf{H}}^{(\ell)}$	Output of layer $\ell-1$ from a compressed network
$h_i^{(\ell)}$	column i of matrix $\mathbf{H}^{(\ell)}$
$a_{ij}^{(\ell)}$	The Attention score between nodes i and j in layer ℓ
$\hat{a}_{ij}^{(\ell)}$	The attention score between nodes i and j in layer ℓ from a smaller network

B Related Work

GNNs & Graph Transformers Many variants of GNNs have been used for tackling transductive tasks on graphs. Examples include GCN (Kipf and Welling, 2016), GAT (Veličković et al., 2018), and GraphSAGE (Hamilton et al., 2017). Recently, Graph Transformers have been developed to address the long-range dependencies that message-passing-based methods struggle to capture. Since full-Transformers are very costly, especially for graphs with millions of nodes, the focus has been on sparser patterns of the Transformer (Shirzad et al., 2023, 2024) or low-rank estimations of the attention matrix (Wu et al., 2022, 2023, 2019; Deng et al., 2024).

Oversmoothing is a problem rising in many variants of GNNs (Oono and Suzuki, 2019; Nt and Maehara, 2019; Roth et al., 2024). In this problem, all the embeddings converge to a single vector, or embeddings matrix loses its rank rapidly. The rank loss problem is not unique to message-passing — attention mechanisms can suffer from a similar problem as well (Dong et al., 2021). Lower rank embeddings are not always bad, many coarsening/pooling-based methods on graphs rely on combining nodes with similar features into single nodes to reduce the graph size and process only a much smaller graph (Liu et al., 2022).

Theoretical Works Many recent works have theoretically analyzed the required hidden dimension or depth of the Transformers or GNNs for graph-related algorithms (Loukas, 2019; Sanford et al., 2024c,b,a). Also, Johnson-Lindestrauss lemma have helped to analyze Transformers for investigating their power in learning diverse attention patterns (Likhosherstov et al., 2021). A variant of linear Transformer have been designed based on this lemma (Wang et al., 2020).

C Proofs

C.1 Proof of Theorem 3.3

Proof. In the proof we use hat notation, $\widehat{\square}$, for the vectors and matrices from $\widehat{\mathcal{T}}$, for example, $\hat{h}^{(\ell)}$ are the outputs of layer ℓ, and $\widehat{\mathbf{W}}_\cdot$ are the weight matrices for this network. In all layers for both networks $\mathbf{W}_V, \mathbf{W}_1$, and $\mathbf{W}_2$, are of the same size, so we set $\widehat{\mathbf{W}}_V = \mathbf{W}_V$, $\widehat{\mathbf{W}}_1 = \mathbf{W}_1$, and $\widehat{\mathbf{W}}_2 = \mathbf{W}_2$.

For the proof, we want to find $\varepsilon^{(0)}, \cdots, \varepsilon^{(L)}$ in a way that for any v in layer ℓ, $|h_v^{(\ell)} - \hat{h}_v^{(\ell)}| < \varepsilon^{(\ell)}$. We will find these bounds inductively, starting from the first layer. We have $\varepsilon^{(0)} = 0$, as both networks have the same input, and we want to bound $\varepsilon^{(\ell+1)}$ based on $\varepsilon^{(\ell)}$.

We have $\mathbf{Q}^{(\ell)} = \mathbf{W}_Q^{(\ell)}\mathbf{H}^{(\ell)}$, $\mathbf{K}^{(\ell)} = \mathbf{W}_K^{(\ell)}\mathbf{H}^{(\ell)}$ and assume $\bar{\mathbf{Q}}^{(\ell)} = \mathbf{W}_Q^{(\ell)}\widehat{\mathbf{H}}^{(\ell)}$, $\bar{\mathbf{K}}^{(\ell)} = \mathbf{W}_K^{(\ell)}\widehat{\mathbf{H}}^{(\ell)}$. Because of the operator norm of matrices $\mathbf{W}_Q$ and $\mathbf{W}_K$, for each i we have $\|q_i^{(\ell)} - \bar{q}_i^{(\ell)}\| \leq \varepsilon^{(\ell)}\beta$ and $\|k_i^{(\ell)} - \bar{k}_i^{(\ell)}\| \leq \varepsilon^{(\ell)}\beta$. Also, we have $\|q_i^{(\ell)}\|, \|k_i^{(\ell)}\| \leq \beta\sqrt{\alpha}$, thus $\|\bar{q}_i^{(\ell)}\|, \|\bar{k}_i^{(\ell)}\| \leq \beta(\varepsilon^{(\ell)} + \sqrt{\alpha})$. Now, for each pair of i and j, we have:

$$\begin{aligned}
|q_i^{(\ell)} \cdot k_j^{(\ell)} - \bar{q}_i^{(\ell)} \cdot \bar{k}_j^{(\ell)}| &= |q_i^{(\ell)} \cdot k_j^{(\ell)} - \bar{q}_i^{(\ell)} \cdot k_j^{(\ell)} + \bar{q}_i^{(\ell)} \cdot k_j^{(\ell)} - \bar{q}_i^{(\ell)} \cdot \bar{k}_j^{(\ell)}| \\
&\leq |q_i^{(\ell)} \cdot k_j^{(\ell)} - \bar{q}_i^{(\ell)} \cdot k_j^{(\ell)}| + |\bar{q}_i^{(\ell)} \cdot k_j^{(\ell)} - \bar{q}_i^{(\ell)} \cdot \bar{k}_j^{(\ell)}| \\
&= |(q_i^{(\ell)} - \bar{q}_i^{(\ell)}) \cdot k_j^{(\ell)}| + |\bar{q}_i^{(\ell)} \cdot (k_j^{(\ell)} - \bar{k}_j^{(\ell)})| \\
&\leq \|q_i^{(\ell)} - \bar{q}_i^{(\ell)}\|\|k_j^{(\ell)}\| + \|\bar{q}_i^{(\ell)}\|\|k_j^{(\ell)} - \bar{k}_j^{(\ell)}\| \\
&\leq \sqrt{\alpha}\beta\varepsilon^{(\ell)} + (\sqrt{\alpha} + \beta\varepsilon^{(\ell)})\beta\varepsilon^{(\ell)} \\
&= 2\sqrt{\alpha}\beta\varepsilon^{(\ell)} + (\beta\varepsilon^{(\ell)})^2
\end{aligned}$$

On the other hand, according to the 3.2, for a $0 < \varepsilon < 1/2$ and $d = \mathcal{O}(\frac{\log(n)}{\varepsilon^2})$ there exists a matrix $\mathbf{M}_{QK} \in \mathbb{R}^{d\times D}$, such that if we define $\widehat{\mathbf{Q}}^{(\ell)} = \mathbf{M}_{QK}\bar{\mathbf{Q}}^{(\ell)}$ and $\widehat{\mathbf{K}}^{(\ell)} = \mathbf{M}_{QK}\bar{\mathbf{K}}^{(\ell)}$, $|\bar{q}_i^{(\ell)} \cdot \bar{k}_j^{(\ell)} - \hat{q}_i^{(\ell)} \cdot \hat{k}_j^{(\ell)}| < \beta^2(\alpha + (\varepsilon^{(\ell)})^2 + 2\sqrt{\alpha}\varepsilon^{(\ell)})\varepsilon$ for all (i, j) pairs in the attention pattern. Note that we can define $\widehat{\mathbf{W}}_Q^{(\ell)} = \mathbf{M}_{QK}^{(\ell)}\mathbf{W}_Q^{(\ell)}$, and $\widehat{\mathbf{W}}_K^{(\ell)} = \mathbf{M}_{QK}^{(\ell)}\mathbf{W}_K^{(\ell)}$, both in $\mathbb{R}^{d\times D}$, as weights for the narrow attention score estimator network. With a triangle inequality we have

$$|q_i^{(\ell)} \cdot k_i^{(\ell)} - \hat{q}_i^{(\ell)} \cdot \hat{k}_i^{(\ell)}| < \beta^2(\alpha + (\varepsilon^{(\ell)})^2 + 2\sqrt{\alpha}\varepsilon^{(\ell)})\varepsilon + 2\sqrt{\alpha}\beta\varepsilon^{(\ell)} + (\beta\varepsilon^{(\ell)})^2.$$

By setting $\varepsilon^{(\ell)} \leq 1$, we have

$$|q_i^{(\ell)} \cdot k_i^{(\ell)} - \hat{q}_i^{(\ell)} \cdot \hat{k}_i^{(\ell)}| < \beta^2(\alpha + 1 + 2\sqrt{\alpha})\varepsilon + \beta(2\sqrt{\alpha} + \beta)\varepsilon^{(\ell)}.$$

Let us define $\varepsilon_a = \beta^2(\alpha + 1 + 2\sqrt{\alpha})\varepsilon + \beta(2\sqrt{\alpha} + \beta)\varepsilon^{(\ell)}$, we have:

$$\begin{aligned}
\widehat{a}_{ij}^{(\ell)} &= \frac{\exp(\hat{q}_i^{(\ell)} \cdot \hat{k}_j^{(\ell)})}{\sum_{u\in\mathcal{N}_H(i)} \exp(\hat{q}_i^{(\ell)} \cdot \hat{k}_u^{(\ell)})} \leq \frac{\exp(q_i^{(\ell)} \cdot k_j^{(\ell)} + \varepsilon_a)}{\sum_{u\in\mathcal{N}_H(i)} \exp(q_i^{(\ell)} \cdot k_j^{(\ell)} - \varepsilon_a)} \leq a_{ij}^{(\ell)} \exp(2\varepsilon_a) \\
\widehat{a}_{ij}^{(\ell)} &= \frac{\exp(\hat{q}_i^{(\ell)} \cdot \hat{k}_j^{(\ell)})}{\sum_{u\in\mathcal{N}_H(i)} \exp(\hat{q}_i^{(\ell)} \cdot \hat{k}_u^{(\ell)})} \geq \frac{\exp(q_i^{(\ell)} \cdot k_j^{(\ell)} - \varepsilon_a)}{\sum_{u\in\mathcal{N}_H(i)} \exp(q_i^{(\ell)} \cdot k_u^{(\ell)} + \varepsilon_a)} \geq a_{ij}^{(\ell)} \exp(-2\varepsilon_a)
\end{aligned}$$

Take note that if $2\varepsilon_a < 1$, we have $\exp(2\varepsilon_a) < 1 + 2\varepsilon_a$ and $\exp(-2\varepsilon_a) > 1 - \varepsilon_a$, and for any i, j, $a_{i,j}^{(\ell)} \leq 1$. Thus for any i and j, $\frac{a_{i,j}^{(\ell)}}{\widehat{a}_{i,j}^{(\ell)}} = 1 + \mathcal{O}(\varepsilon_a) = 1 + \mathcal{O}(\varepsilon^{(\ell)})$.

Now we bound $\|h_i^{(\ell+1/2)} - \hat{h}_i^{(\ell+1/2)}\|$:

$$
\begin{aligned}
\|h_i^{(\ell+1/2)} - \hat{h}_i^{(\ell+1/2)}\| &= \|\sum_{j\in Nei(i)} a_{ij}^{(\ell)} v_j^{(\ell)} - \widehat{a}_{ij}\hat{v}_j^{(\ell+1/2)}\| \\
&= \|\sum_{j\in Nei(i)} a_{ij}^{(\ell)} v_j^{(\ell)} - \widehat{a}_{ij}^{(\ell)} v_j^{(\ell)} + \widehat{a}_{ij}^{(\ell)} v_j^{(\ell)} - \widehat{a}_{ij}\hat{v}_j^{(\ell)}\| \\
&= \|\sum_{j\in Nei(i)} (a_{ij}^{(\ell)} - \widehat{a}_{ij}^{(\ell)}) v_j^{(\ell)} + \widehat{a}_{ij}^{(\ell)}(v_j^{(\ell)} - \hat{v}_j^{(\ell)})\| \\
&= \|(v_j^{(\ell)} - \hat{v}_j^{(\ell)}) + v_j^{(\ell)} \sum_{j\in Nei(i)} (a_{ij}^{(\ell)} - \widehat{a}_{ij}^{(\ell)})\| \\
&\leq \|v_j^{(\ell)} - \hat{v}_j^{(\ell)}\| + \|v_j^{(\ell)}\| \sum |a_{ij}^{(\ell)} - \widehat{a}_{ij}^{(\ell)}| \\
&\leq \varepsilon^{(\ell)}\beta + \sqrt{\alpha}\sum \max(1-\exp(-2\varepsilon_a), \exp(2\varepsilon_a) - 1) a_{ij}^{(\ell)} \\
&\leq \varepsilon^{(\ell)}\beta + \sqrt{\alpha}(\exp(2\varepsilon_a) - 1),
\end{aligned}
$$

and since $1 + x < \exp(x) < 1 + 2x$ for $0 < x < 1$, if we have $\varepsilon_a < 1$, we have

$$
\|h_i^{(\ell+1/2)} - \hat{h}_i^{(\ell+1/2)}\| \leq \beta\varepsilon^{(\ell)} + 4\sqrt{\alpha}\varepsilon_a \tag{1}
$$

For the feed-forward network part, we know that this network is β^2-Lipschitz because $\mathbf{W}_1^{(\ell)}$ and $\mathbf{W}_2^{(\ell)}$ have maximum operator norm β and σ is a 1-Lipschitz activation function. Thus we have

$$
\|h_i^{(\ell+1)} - \hat{h}_i^{(\ell+1)}\| \leq \beta^2(\beta\varepsilon^{(\ell)} + 4\sqrt{\alpha}\varepsilon_a) = (\beta^3 + 8\beta\alpha + 4\beta^2\sqrt{\alpha})\varepsilon^{(\ell)} + 4\beta^2(\alpha\sqrt{\alpha} + 2\alpha + \sqrt{\alpha})\varepsilon.
$$

Both $\beta^3 + 8\beta\alpha + 4\beta^2\sqrt{\alpha}$ and $4\beta^2(\alpha\sqrt{\alpha} + 2\alpha + \sqrt{\alpha})$ are constants, and if we define them as c_1 and c_2, we have

$$
\varepsilon^{(\ell+1)} \leq c_1\varepsilon^{(\ell)} + c_2\varepsilon
$$

Given $\varepsilon^{(0)} = 0$, as both networks get the same input, we have

$$
\begin{aligned}
\varepsilon^{(L)} &\leq c_1\varepsilon^{(L-1)} + c_2\varepsilon \\
&\leq c_1(c_1\varepsilon^{(L-2)} + c_2\varepsilon) + c_2\varepsilon \\
&\cdots \\
&\leq c_2\varepsilon(c_1^{L-1} + \cdots + c_1) \\
&= \frac{c_1(c_2^L - 1)}{c_2 - 1}\varepsilon
\end{aligned}
$$

While the error increases exponentially with the number of layers, when we have $L = O(1)$, then the error is bounded by a constant factor of chosen ε. Now, we know that $\|\mathcal{T}(X)_i - \widehat{\mathcal{T}}(X)_i\|_2 \leq \varepsilon^{(L)} = \mathcal{O}(\varepsilon)$.

This also holds that $\varepsilon^{(\ell)} = \mathcal{O}(\varepsilon)$ for each ℓ, thus $\frac{a_{ij}^{(\ell)}}{\widehat{a}_{ij}^{(\ell)}} = 1 + \mathcal{O}(\varepsilon)$. □

C.2 Proof of Proposition 4.1

Proof. First of all, take note that any matrix $\mathbf{B} \in \mathbb{R}^{D\times n}$ of rank d, we have $\mathbf{U} \in \mathbb{R}^{D\times d}$ and $\mathbf{\Lambda} \in \mathbb{R}^{d\times D}$ that $\mathbf{U}\mathbf{\Lambda}\mathbf{B} = \mathbf{B}$. Particularly, $\mathbf{\Lambda}$ can be a selection of rows from $\mathbf{B}$ covering the whole span of columns of $\mathbf{B}$, and $\mathbf{U}$ will be a linear combination of the selected rows making columns of $\mathbf{B}$.

We will make embeddings in each layer in a way that the embeddings from the low-dimensional network can be mapped linearly to the high dimension D to recreate the embeddings from the high-dimensional network.

Let us assume for each layer ℓ the input of the small network is $\widehat{\mathbf{H}}^{(\ell)} \in \mathbb{R}^{d\times n}$, such that for a $\mathbf{U} \in \mathbb{R}^{D\times d}$, $\mathbf{U}_{in}^{(\ell)}\widehat{\mathbf{H}}^{(\ell)} = \mathbf{H}^{(\ell)}$. Now, we will create the weights for the layer ℓ of $\widehat{\mathcal{T}}$ to have sizes $d \times d$, and the output of the layer can be mapped with a linear map $\mathbf{U}_{out}^{(\ell)}$ to the outputs of the layer ℓ from $\mathcal{T}$.

First, we will show the possibility of consistency in the attention scores. The attention scores in the matrix are a sparse version of $\mathbf{A} = \mathbf{H}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{H}$, for this part, since all the matrices and representations are in layer $^{(\ell)}$, we remove $^{(\ell)}$ superscripts for the brevity of the writing. We need a $\widehat{\mathbf{W}}_Q, \widehat{\mathbf{W}}_K \in \mathbb{R}^{d\times d}$ that $\widehat{\mathbf{A}} = \widehat{\mathbf{H}}^\mathsf{T}\widehat{\mathbf{W}}_Q^\mathsf{T}\widehat{\mathbf{W}}_K\widehat{\mathbf{H}}$ gives us a similar attention matrix. We have:

$$\mathbf{A} = \mathbf{H}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{H} = \widehat{\mathbf{H}}^\mathsf{T}\mathbf{U}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{U}\widehat{\mathbf{H}}.$$

Take note that $\mathbf{U}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{U}$ is a matrix of shape $d \times d$. Now, if we have $\widehat{\mathbf{W}}_K = \mathbf{U}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{U}$ and $\widehat{\mathbf{W}}_Q = I_d$, where I_d is the identity matrix of size d, we have $\widehat{\mathbf{A}} = \widehat{\mathbf{H}}^\mathsf{T}\widehat{\mathbf{W}}_Q^\mathsf{T}\widehat{\mathbf{W}}_K\widehat{\mathbf{H}} = \mathbf{A}$.

Now for compressing the $\mathbf{W}_V$, we have $\mathbf{V} = \mathbf{W}_V\mathbf{H} = \mathbf{W}_V\mathbf{U}\widehat{\mathbf{H}}$. Since $\mathrm{rank}(\mathbf{H}) \le d$, $\mathrm{rank}(\mathbf{V}) \le d$. Thus there should be $\mathbf{U}_V \in \mathbb{R}^{D\times d}$ and $\mathbf{\Lambda}_V \in \mathbb{R}^{d\times D}$ that $\mathbf{U}_V\mathbf{\Lambda}_V\mathbf{V} = \mathbf{V}$. We have $\mathbf{V} = \mathbf{U}_V\mathbf{\Lambda}_V\mathbf{W}_V\mathbf{U}\widehat{\mathbf{H}}$ and thus we can take $\widehat{\mathbf{W}}_V = \mathbf{\Lambda}_V\mathbf{W}_V\mathbf{U}$, and thus $\widehat{\mathbf{W}}_V \in \mathbb{R}^{d\times d}$. Also, $\widehat{\mathbf{V}} = \widehat{\mathbf{W}}_V\widehat{\mathbf{H}}$, and we have $\mathbf{V} = \mathbf{U}_V\widehat{\mathbf{V}}$.

We have $\mathbf{H}^{(\ell+1/2)} = \mathbf{V}\mathbf{A}$, and we will have $\widehat{\mathbf{H}} = \widehat{\mathbf{V}}\mathbf{A} = \mathbf{U}_V\mathbf{V}\mathbf{A}$. Thus, $\mathbf{H}^{(\ell+1/2)} = \mathbf{U}_V\widehat{\mathbf{H}}^{(\ell+1/2)}$.

The next linear mapping comes with an activation function; thus, we can not do exactly the same trick as $\mathbf{V}$ mapping. We have $\mathbf{H}^{(\ell+3/4)} = \sigma(\mathbf{W}_1\mathbf{H}^{(\ell+1/2)}) = \sigma(\mathbf{W}_1\mathbf{U}_V\widehat{\mathbf{H}}^{(\ell+1/2)})$. Now, since we know $\mathrm{rank}(\mathbf{H}^{(\ell+3/4)}) \le d$, we have $\mathbf{U}_\sigma \in \mathbb{R}^{D\times d}$ and $\mathbf{\Lambda}_\sigma \in \mathbb{R}^{d\times D}$ in a way that $\mathbf{U}_\sigma\mathbf{\Lambda}_\sigma\mathbf{H}^{(\ell+3/4)} = \mathbf{H}^{(\ell+3/4)}$. In this case, we construct $\mathbf{U}_\sigma$ and $\mathbf{\Lambda}_\sigma$ so that we can also reduce the size of $\mathbf{W}_1^{(\ell)}$. To construct this, we choose $\mathbf{\Lambda}_\sigma$ to have each row as a 1-hot vector, selecting maximum d rows from $\mathbf{H}^{(\ell+3/4)}$ that any other rows in the $\mathbf{H}^{(\ell+3/4)}$ can be constructed from a linear combination of these rows. In this construction, we have $\mathbf{\Lambda}_\sigma\sigma(\mathbf{W}_1h^{(\ell+1/2)}) = \sigma(\mathbf{\Lambda}_\sigma\mathbf{W}_1h^{(\ell+1/2)})$, since $\mathbf{\Lambda}_\sigma$ just selects rows from the mapped value and the activation function is an element-wise function. Thus we can have $\widehat{\mathbf{W}}_1^{(\ell)} = \mathbf{\Lambda}_\sigma\mathbf{W}_1^{(\ell)}\mathbf{U}_V \in \mathbb{R}^{d\times d}$. And we have $\mathbf{H}^{(\ell+3/4)} = \mathbf{U}_\sigma\widehat{\mathbf{H}}^{(\ell+3/4)}$.

The next linear layer exactly similar to the V mapping can be compressed in a way that we have $\mathbf{H}^{(\ell+1)} = \mathbf{U}_{out}\widehat{\mathbf{H}}^{(\ell+1)}$. Now by induction since the assumption is correct for the input of the network, and if the assumption holds for the input we can have a compressed layer that the assumption will hold for the output layer, and the output of each layer is the input of the next layer the theorem is proved. □

C.3 Proof of Theorem 4.2

We will first prove the following lemma which will be used in several upcoming theorems:

Lemma C.1. *If $\mathbf{H}, \bar{\mathbf{H}} \in \mathbb{R}^{D\times n}$, $\mathrm{rank}(\bar{\mathbf{H}}) = d$, and for each row i, $\|\mathbf{H}_i - \bar{\mathbf{H}}_i\| \le \varepsilon$ for some value ε, there exist matrices $\mathbf{U} \in \mathbb{R}^{D\times d}$ and $\mathbf{\Lambda} \in \mathbb{R}^{d\times D}$ such that $\forall i : \|\mathbf{H}_i - \mathbf{U}\mathbf{\Lambda}\mathbf{H}_i\| \le \varepsilon$.*

Proof. Take $\mathbf{\Lambda}$ to be d base vectors of size D making column span of $\bar{\mathbf{H}}$. Each column of $\mathbf{H}$ has maximum distance of ε from this span since $\|\mathbf{H}_i - \bar{\mathbf{H}}_i\| \le \varepsilon$. Thus we can have a $\mathbf{U} \in \mathbb{R}^{D\times d}$ that $\mathbf{U}\mathbf{\Lambda}\mathbf{H}$ will be the projection of the $\mathbf{H}$ to the span of columns of $\bar{\mathbf{H}}$, and since this projection is the minimum distance we have $\|\mathbf{U}\mathbf{\Lambda}\mathbf{H}_i - \mathbf{H}_i\| \le \|\bar{\mathbf{H}}_i - \mathbf{H}_i\| \le \varepsilon$. □

Proof of the theorem. We will prove this theorem inductively by making the embeddings in each layer in a way that the embeddings from the low-dimensional network can be mapped to the high dimension D to approximate the high-dimensional network. We will prove the following lemma that helps both in the first layer and dimension reduction for $\mathbf{W}_2$ mappings.

Thus, because input $\mathbf{X}$ has a low-rank estimation $\bar{\mathbf{X}}$, we can make $\widehat{\mathbf{X}} = \mathbf{\Lambda}\mathbf{X}$, in a way that there exist a matrix $\mathbf{U}$ that $\forall i : \|\mathbf{U}\hat{x}_i - x_i\| \le \varepsilon$. For the convenience of the writing, we take $\mathbf{H}^{(0)}$ as $\mathbf{X}$ and $\widehat{\mathbf{H}}^{(0)}$ as $\widehat{\mathbf{X}}$. Now, we have a $\mathbf{U}$ that $\widehat{\mathbf{H}}^{(0)}$ that $\|\mathbf{U}^{(\ell)}\hat{h}^{(0)} - h_i^{(0)}\| \le \varepsilon$. We will prove the step of the induction with the following lemma:

For the step of our induction assume in a layer such as ℓ, we have input $\widehat{\mathbf{H}}^{(\ell)} \in \mathbb{R}^{d\times n}$ such that there exist $\mathbf{U}^{(\ell)}$ that $\forall i : \|\mathbf{U}^{(\ell)}\hat{h}^{(\ell)} - h_i^{(\ell)}\| \leq \varepsilon^{(\ell)}$, there exist a Transformer layer of width d as the next layer that there exist $\mathbf{U}^{(\ell+1)}$ that $\forall i : \|\mathbf{U}^{(\ell+1)}\hat{h}_i^{(\ell+1)} - h_i^{(\ell+1)}\| \leq c_1\varepsilon^{(\ell)} + c_2\varepsilon$ for constants $c_1, c_2 = \mathcal{O}(1)$.

First, we will show the possibility of consistency in the attention scores. The attention scores in the matrix are a sparse version of $\mathbf{A} = \mathbf{H}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{H}$, for this part, since all the matrices and representations are in layer $^{(\ell)}$, we remove $^{(\ell)}$ superscripts for the brevity of the writing. We need a $\widehat{\mathbf{W}}_Q, \widehat{\mathbf{W}}_K \in \mathbb{R}^{d\times d}$ that $\hat{\mathbf{A}} = \widehat{\mathbf{H}}^\mathsf{T}\widehat{\mathbf{W}}_Q^\mathsf{T}\widehat{\mathbf{W}}_K\widehat{\mathbf{H}}$ gives us a similar attention matrix. We start by estimating the $\mathbf{A} = \mathbf{H}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{H}$ with $\bar{\mathbf{A}} = \bar{\mathbf{H}}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\bar{\mathbf{H}}$. Now, we know that $\widehat{\mathbf{H}} = \mathbf{\Lambda}\bar{\mathbf{H}}$ and $\mathbf{U}\widehat{\mathbf{H}} = \bar{\mathbf{H}}$, thus $\bar{\mathbf{H}} = \mathbf{U}\mathbf{\Lambda}\bar{\mathbf{H}}$. By replacing this in the attention estimation we have

$$\bar{\mathbf{A}} = \bar{\mathbf{H}}^\mathsf{T}\mathbf{\Lambda}^\mathsf{T}\mathbf{U}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{U}\mathbf{\Lambda}\bar{\mathbf{H}} = \widehat{\mathbf{H}}^\mathsf{T}\mathbf{U}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{U}\widehat{\mathbf{H}}.$$

Take note that $\mathbf{U}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{U}$ is a $d \times d$ matrix. Now, if we define $\widehat{\mathbf{W}}_K = \mathbf{U}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\mathbf{U}$ and $\widehat{\mathbf{W}}_Q = \mathbf{I}_d$, where $\mathbf{I}_d$ is the identity matrix of size d, we have

$$\hat{\mathbf{A}} = \widehat{\mathbf{H}}^\mathsf{T}\widehat{\mathbf{W}}_Q^\mathsf{T}\widehat{\mathbf{W}}_K\widehat{\mathbf{H}} = \bar{\mathbf{H}}^\mathsf{T}\mathbf{W}_Q^\mathsf{T}\mathbf{W}_K\bar{\mathbf{H}} = \bar{\mathbf{A}}.$$

We have $\mathbf{Q}^{(\ell)} = \mathbf{W}_Q^{(\ell)}\mathbf{H}^{(\ell)}$, $\mathbf{K}^{(\ell)} = \mathbf{W}_K^{(\ell)}\mathbf{H}^{(\ell)}$ and if we define $\bar{\mathbf{Q}}^{(\ell)} = \mathbf{W}_Q^{(\ell)}\bar{\mathbf{H}}^{(\ell)}$, $\bar{\mathbf{K}}^{(\ell)} = \mathbf{W}_K^{(\ell)}\bar{\mathbf{H}}^{(\ell)}$. Very similar to what we saw in 3.3, because of the operator norm of matrices $\mathbf{W}_Q$ and $\mathbf{W}_K$, for each i we have $\|q_i^{(\ell)} - \bar{q}_i^{(\ell)}\| \leq \varepsilon^{(\ell)}\beta$ and $\|k_i^{(\ell)} - \bar{k}_i^{(\ell)}\| \leq \varepsilon^{(\ell)}\beta$. Also, we have $\|q_i^{(\ell)}\|, \|k_i^{(\ell)}\| \leq \beta\sqrt{\alpha}$, thus $\|\bar{q}_i^{(\ell)}\|, \|\bar{k}_i^{(\ell)}\| \leq \beta(\varepsilon^{(\ell)} + \sqrt{\alpha})$. Now, as we proved in 3.3 we have:

$$|q_i^{(\ell)} \cdot k_j^{(\ell)} - \bar{q}_i^{(\ell)} \cdot \bar{k}_j^{(\ell)}| \leq 2\sqrt{\alpha}\beta\varepsilon^{(\ell)} + (\beta\varepsilon^{(\ell)})^2 \leq \beta(2\sqrt{\alpha} + \beta)\varepsilon^{(\ell)}.$$

The last inequality is because we have $\varepsilon^{(\ell)} \leq 1$, which holds for a sufficiently small ε, as we will see toward the end of the proof.

Now if we define $\varepsilon_a = \beta(2\sqrt{\alpha} + \beta)\varepsilon^{(\ell)}$, we have:

$$\widehat{a}_{ij} = \bar{a}_{ij} = \frac{\exp(\bar{q}_i \cdot \bar{k}_j)}{\sum_{u\in\mathcal{N}_H(i)}\exp(\bar{q}_i \cdot \bar{k}_u)} \leq \frac{\exp(q_i \cdot k_j + \varepsilon_a)}{\sum_{u\in\mathcal{N}_H(i)}\exp(q_i \cdot k_j - \varepsilon_a)} \leq a_{ij}\exp(2\varepsilon_a)$$
$$\widehat{a}_{ij} = \bar{a}_{ij} = \frac{\exp(\bar{q}_i \cdot \bar{k}_j)}{\sum_{u\in\mathcal{N}_H(i)}\exp(\bar{q}_i \cdot \bar{k}_u)} \geq \frac{\exp(q_i \cdot k_j - \varepsilon_a)}{\sum_{u\in\mathcal{N}_H(i)}\exp(q_i \cdot k_u + \varepsilon_a)} \geq a_{ij}\exp(-2\varepsilon_a)$$

Take notice that if $2\varepsilon_a < 1$, we have $\exp(2\varepsilon_a) < 1 + 2\varepsilon_a$ and $\exp(-2\varepsilon_a) > 1 - \varepsilon_a$, and for any i, j, $a_{i,j}^{(\ell)} \leq 1$. Thus for any i and j, $\frac{a_{i,j}^{(\ell)}}{\widehat{a}_{i,j}^{(\ell)}} = 1 + \mathcal{O}(\varepsilon_a) = 1 + \mathcal{O}(\varepsilon^{(\ell)})$.

Now for compressing the $\mathbf{W}_V$, we start from $\bar{\mathbf{H}}$ and we have $\bar{\mathbf{V}} = \mathbf{W}_V\bar{\mathbf{H}} = \mathbf{W}_V\mathbf{U}\widehat{\mathbf{H}}$. Now, according to the operator norm of $\mathbf{W}_V$, we know that $\max_i\|v_i - \bar{v}_i\| \leq \beta\varepsilon^{(\ell)}$. On the other hand since $\mathrm{rank}(\bar{\mathbf{H}}) \leq d$, $\mathrm{rank}(\bar{\mathbf{V}}) \leq d$. Thus there should be $\mathbf{U}_V \in \mathbb{R}^{D\times d}$ and $\mathbf{\Lambda}_V \in \mathbb{R}^{d\times D}$ that $\mathbf{U}_V\mathbf{\Lambda}_V\bar{\mathbf{V}} = \bar{\mathbf{V}}$. Let us define $\widehat{\mathbf{V}} = \mathbf{\Lambda}_V\bar{\mathbf{V}}$. Then we have

$$\bar{\mathbf{V}} = \mathbf{W}_V\mathbf{U}\widehat{\mathbf{H}} = \mathbf{U}_V\mathbf{\Lambda}_V\mathbf{W}_V\mathbf{U}\widehat{\mathbf{H}}.$$

Now take $\widehat{\mathbf{W}}_V = \mathbf{\Lambda}_V\mathbf{W}_V\mathbf{U}$, and thus $\widehat{\mathbf{W}}_V \in \mathbb{R}^{d\times d}$. Also, $\widehat{\mathbf{V}} = \widehat{\mathbf{W}}_V\widehat{\mathbf{H}}$, and we have $\bar{\mathbf{V}} = \mathbf{U}_V\widehat{\mathbf{V}}$, and thus $\max_i\|v_i - \mathbf{U}_V\hat{v}_i\| \leq \beta\varepsilon^{(\ell)}$.

Very similar to the proof in Theorem 3.3, we can bound the $\|h_i^{(\ell+1/2)} - \mathbf{U}_V\hat{h}_i^{(\ell+1/2)}\|$:

$$
\begin{aligned}
\|h_i^{(\ell+1/2)} - \mathbf{U}_V \hat{h}_i^{(\ell+1/2)}\| &= \|\sum_{j\in Nei(i)} a_{ij}^{(\ell)} v_j^{(\ell)} - \widehat{a}_{ij}\mathbf{U}_V \hat{v}_j^{(\ell+1/2)}\| \\
&= \|\sum_{j\in Nei(i)} a_{ij}^{(\ell)} v_j^{(\ell)} - \widehat{a}_{ij}^{(\ell)} v_j^{(\ell)} + \widehat{a}_{ij}^{(\ell)} v_j^{(\ell)} - \widehat{a}_{ij}\mathbf{U}_V \hat{v}_j^{(\ell)}\| \\
&= \|(v_j^{(\ell)} - \mathbf{U}_V \hat{v}_j^{(\ell)}) + v_j^{(\ell)} \sum_{j\in Nei(i)} (a_{ij}^{(\ell)} - \widehat{a}_{ij}^{(\ell)})\| \\
&\leq \|v_j^{(\ell)} - \mathbf{U}_V \hat{v}_j^{(\ell)}\| + \|v_j^{(\ell)}\| \sum |a_{ij}^{(\ell)} - \widehat{a}_{ij}^{(\ell)}| \\
&\leq \beta\varepsilon^{(\ell)} + \sqrt{\alpha}\sum \max(1-\exp(-2\varepsilon_a), \exp(2\varepsilon_a)-1) a_{ij}^{(\ell)} \\
&\leq \beta\varepsilon^{(\ell)} + \sqrt{\alpha}(\exp(2\varepsilon_a)-1),
\end{aligned}
$$

and since $1 + x < \exp(x) < 1 + 2x$ for $0 < x < 1$, if we have $\varepsilon_a < 1$, we have

$$
\|h_i^{(\ell+1/2)} - \mathbf{U}_V \hat{h}_i^{(\ell+1/2)}\| \leq \beta\varepsilon^{(\ell)} + 4\sqrt{\alpha}\varepsilon_a = \beta(1 + (8\alpha + \beta\sqrt{\alpha}))\varepsilon^{(\ell)}
$$

For the convenience of writing we take $\varepsilon_b = \beta(1 + (8\alpha + \beta\sqrt{\alpha}))\varepsilon^{(\ell)}$. For the feedforward network part, we know $\mathbf{W}_1$ has operator norm β and σ is 1-Lipschitz. Thus for each i we have,

$$
\|\sigma(\mathbf{W}_1 h_i^{(\ell+1/2)}) - \sigma(\mathbf{W}_1 \mathbf{U}_V \hat{h}_i^{(\ell+1/2)})\| \leq \beta\varepsilon_b.
$$

Now, we take $\widehat{\mathbf{W}}_1 = \mathbf{W}_1\mathbf{U}_V$ and this will give us

$$
\|\sigma(\mathbf{W}_1 h_i^{(\ell+1/2)}) - \sigma(\widehat{\mathbf{W}}_1 \hat{h}_i^{(\ell+1/2)})\| \leq \beta\varepsilon_b.
$$

Also, we know that $\forall i : \|h_i^{(\ell+3/4)} - \bar{h}_i^{(\ell+3/4)}\| \leq \varepsilon$, thus with the triangle inequality, we have $\forall i : \|\hat{h}_i^{(\ell+3/4)} - \bar{h}_i^{(\ell+3/4)}\| \leq \varepsilon + \beta\varepsilon_b$. $\mathbf{W}_2^{(\ell)}$ has an operator norm less than or equal to β, thus

$$
\forall i : \|\mathbf{W}_2^{(\ell)} \bar{h}_i^{(\ell+3/4)} - \mathbf{W}_2^{(\ell)} \hat{h}_i^{(\ell+3/4)}\| \leq \beta\varepsilon + \beta^2\varepsilon_b.
$$

Since $\mathrm{rank}(\bar{\mathbf{H}}^{(\ell+3/4)}) \leq d$, then $\mathrm{rank}(\mathbf{W}_2^{(\ell)}\bar{\mathbf{H}}) \leq d$. Thus $\mathbf{W}_2^{(\ell)}\widehat{\mathbf{H}}^{(\ell+3/4)}$ has a lower rank approximation with a column-wise maximum distance of $\beta\varepsilon + \beta^2\varepsilon_b$. According to the Lemma C.1, we can have $\mathbf{U}^{(\ell+1)} \in \mathbb{R}^{D\times d}$ and $\mathbf{\Lambda}^{(\ell+1)} \in \mathbb{R}^{d\times D}$ that,

$$
\forall i : \|\mathbf{W}_2^{(\ell)} \hat{h}_i^{(\ell+3/4)} - \mathbf{U}^{(\ell+1)}\mathbf{\Lambda}^{(\ell+1)}\mathbf{W}_2^{(\ell)} \hat{h}_i^{(\ell+3/4)}\| \leq \beta\varepsilon + \beta^2\varepsilon_b.
$$

Now, take $\widehat{\mathbf{W}}^{(\ell)} = \mathbf{\Lambda}^{(\ell+1)}\mathbf{W}_2^{(\ell)}$, and this will give us $\widehat{\mathbf{H}}^{(\ell+1)} = \widehat{\mathbf{W}}_2^{(\ell)}\widehat{\mathbf{H}}^{(\ell+3/4)} \in \mathbb{R}^{d\times n}$ that

$$
\forall i : \|\mathbf{U}^{(\ell+1)} \hat{h}_i^{(\ell+1)} - \mathbf{W}_2^{(\ell)} \hat{h}_i^{(\ell+3/4)}\| \leq \beta\varepsilon + \beta^2\varepsilon_b.
$$

We also know that $\forall i : \|\mathbf{W}_2^{(\ell)} h_i^{(\ell+3/4)} - \mathbf{W}_2^{(\ell)} \hat{h}_i^{(\ell+3/4)}\| \leq \beta^2\varepsilon_b$, and $\mathbf{H}^{(\ell+1)} = \mathbf{W}_2^{(\ell)}\mathbf{H}^{(\ell+3/4)}$, thus:

$$
\forall i : \|h_i^{(\ell+1)} - \mathbf{W}_2^{(\ell)} \hat{h}_i^{(\ell+3/4)}\| \leq \beta^2\varepsilon_b.
$$

Combining the results with the triangle inequality we have:

$$
\forall i : \|h_i^{(\ell+1)} - \mathbf{U}^{(\ell+1)} \hat{h}_i^{(\ell+1)}\| \leq \beta\varepsilon + 2\beta^2\varepsilon_b = \beta\varepsilon + \beta^3(1 + (8\alpha + \beta\sqrt{\alpha}))\varepsilon^{(\ell)}.
$$

Thus, if we take $c_1 = \beta^3(1 + (8\alpha + \beta\sqrt{\alpha}))$ and $c_2 = \beta$, we have $c_1, c_2 = \mathcal{O}(1)$, and thus the lemma will be proven. This will prove the induction step.

Now, by induction since the assumptions are correct for the first layer. The assumptions also hold for the input and the assumptions being correct for a layer will result in it being correct for the following layer, for each layer $\varepsilon^{(\ell+1)} \leq c_1\varepsilon^{(\ell)} + c_2\varepsilon$ and $\varepsilon^{(0)} = \varepsilon$. Since the number of layers is $L = \mathcal{O}(1)$, very similar to Theorem 3.3, we have $\|\mathcal{T}(X)_i - \widehat{\mathcal{T}}(X)_i\|_2 \leq \varepsilon^{(L)} = \mathcal{O}(\varepsilon)$ and $\frac{a_{ij}^{(\ell)}}{\widehat{a}_{ij}^{(\ell)}} = 1 + \mathcal{O}(\varepsilon)$. $\square$

C.3.1 Note on impossibility of the reduction using $U\Lambda$ mapping

If we want to also decrease the dimension on the feed-forward layer, similar techniques we used to decrease the dimension of the linear mappings will fail due to the non-linearity of the activation function.

Because $\text{rank}(\bar{\mathbf{H}}^{(\ell+3/4)}) \leq d$, we can have $\mathbf{U}_\sigma \in \mathbb{R}^{D\times d}$ and $\mathbf{\Lambda}_\sigma \in \mathbb{R}^{d\times D}$ in a way that $\mathbf{U}_\sigma\mathbf{\Lambda}_\sigma\bar{\mathbf{H}}^{(\ell+3/4)} = \bar{\mathbf{H}}^{(\ell+3/4)}$. In this case, we construct $\mathbf{U}_\sigma$ and $\mathbf{\Lambda}_\sigma$ so that we can also reduce the size of $\mathbf{W}_1^{(\ell)}$. To construct this, we choose $\mathbf{\Lambda}_\sigma$ to have each row as a 1-hot vector, selecting maximum d rows from $\bar{\mathbf{H}}^{(\ell+3/4)}$ that any other rows in the $\bar{\mathbf{H}}^{(\ell+3/4)}$ can be constructed from a linear combination of these rows. In this construction, we have

$$\mathbf{\Lambda}_\sigma\sigma(\mathbf{W}_1 h^{(\ell+1/2)}) = \sigma(\mathbf{\Lambda}_\sigma\mathbf{W}_1 h^{(\ell+1/2)})$$

, since $\mathbf{\Lambda}_\sigma$ just selects rows from the mapped value and the activation function is an element-wise function. Thus we can have $\widehat{\mathbf{W}}_1^{(\ell)} = \mathbf{\Lambda}_\sigma\mathbf{W}_1^{(\ell)}\mathbf{U}_V$. Furthermore, we have

$$\|\mathbf{\Lambda}_\sigma\bar{h}_i^{(\ell+3/4)} - \hat{h}_i^{(\ell+3/4)}\| \leq \varepsilon + \beta^2\varepsilon^{(\ell)} + 4\sqrt{\alpha}\beta\varepsilon_a,$$

since the distance can not increase by selecting a subset of rows. Now, we have

$$\|\mathbf{U}_\sigma(\mathbf{\Lambda}_\sigma\bar{h}_i^{(\ell+3/4)} - \hat{h}_i^{(\ell+3/4)})\| \leq \|\mathbf{U}_\sigma\|_{op}(\varepsilon + \beta^2\varepsilon^{(\ell)} + 4\sqrt{\alpha}\beta\varepsilon_a).$$

Now, if we can choose $\mathbf{\Lambda}_\sigma$ and $\mathbf{U}_\sigma$ in a way that $\mathbf{U}_\sigma$ has an $\mathcal{O}(1)$ operator norm this can give the contraction for the output of $\widehat{\mathbf{W}}_1^{(\ell)}$ and input of $\widehat{\mathbf{W}}_2^{(\ell)}$. Even a simpler condition that just preserves the distance with some constant around $\bar{h}_i^{(\ell+3/4)}$ vectors in a way that

$$\|\mathbf{U}_\sigma(\mathbf{\Lambda}_\sigma\bar{h}_i^{(\ell+3/4)} - \hat{h}_i^{(\ell+3/4)})\| \leq c\|\mathbf{\Lambda}_\sigma\bar{h}_i^{(\ell+3/4)} - \hat{h}_i^{(\ell+3/4)}\|$$

for some constant c will lead to a lower dimension feed-forward network. However, this is not always correct.

A counter-example to show that this is not always correct is that if $D = n$ and $\bar{\mathbf{H}} \in \mathbb{R}^{D\times n}$ is a matrix with all elements equal to $1/\sqrt{D}$ and $\mathbf{H} = \bar{\mathbf{H}} + I\varepsilon$, then $\text{rank}(\bar{\mathbf{H}}) = 1$, but selecting just one row of $\mathbf{H}$ to estimate the whole matrix should use a constant, c multiplication of that row for estimating all other rows. If this constant is one, then one column will have an error of $D\varepsilon$, otherwise, the error will be at least $\max(|1-c|, c+c\varepsilon-1)D = \theta(\varepsilon D)$. However, in the scenario that $c = 1$, all columns except for one will have maximum distance $\mathcal{O}(\varepsilon)$. However, it is also noteworthy that this counter-example is not a very likely thing to happen in the training of deep neural networks, at least in the presence of regularizers such as dropout and layer-norm we expect the ε distance between $\bar{\mathbf{H}}$ and $\mathbf{H}$ to be more uniformly divided in the rows of $\mathbf{H}$.

C.4 Proof of Proposition 4.3

Definition C.2. *Given a matrix $\mathbf{A}$, the leverage score of the i-th row a_i of $\mathbf{A}$ is defined to be $\ell_i := a_i(\mathbf{A}^\top\mathbf{A})^\dagger a_i^\top$, which is the squared ℓ_2-norm of the i-th row of $\mathbf{U}$, where $A = \mathbf{U}\Sigma\mathbf{V}^T$ is the singular value decomposition of $\mathbf{A}$.*

It is known that sampling $O(d\log d/\varepsilon^2)$ rows with respect to the leverage score of the matrix $[\mathbf{A}, \mathbf{b}]$ gives a $(1\pm\varepsilon)$-subspace embedding of the column span of $[\mathbf{A}, \mathbf{b}]$, which means that the solution x' of the regression problem $\min_x\|\mathbf{S}\mathbf{A}x - \mathbf{S}\mathbf{b}\|_2$ satisfies $\|\mathbf{A}x' - \mathbf{b}\|_2 \leq (1\pm\varepsilon)\min_x\|\mathbf{A}x - \mathbf{b}\|_2$ (see, e.g., Woodruff (2014)). In the recent study of the active regression problem (Chen and Price, 2019; Musco et al., 2022), it turns out that sampling with respect to the leverage score of the matrix $\mathbf{A}$ itself is also sufficient to give a good solution to unknown label vector $\mathbf{b}$ with high constant probability.

Lemma C.3. *Given matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\mathbf{b} \in \mathbb{R}^n$. Let $\mathbf{S}$ be the sampling and rescaling matrix with respect to $\ell_i(\mathbf{A})$ with $O(d\log d)$ rows. Let $x' = \text{argmin}_x\|\mathbf{S}\mathbf{A}x - \mathbf{S}\mathbf{b}\|_2$. Then we have with high constant probability,*

$$\|\mathbf{A}x' - \mathbf{b}\|_2 \leq O(1)\cdot\min_x\|\mathbf{A}x - \mathbf{b}\|_2$$

Proof. Since $\mathbf{S}$ is the sampling and rescaling matrix with respect to $\ell_i(\mathbf{A})$ with $O(d \log d)$ rows, we know with high constant probability, $\mathbf{S}$ is a $O(1)$-subspace embedding of $\mathbf{A}$, which means for all $x \in \mathbb{R}^d$, we have $\|\mathbf{SA}x\|_2 = (1 \pm \varepsilon)\|\mathbf{A}x\|_2$ (see, e.g., (Woodruff, 2014)).

Now, let $x_c = \operatorname{argmin}_{x \in \mathbb{R}^d} \|\mathbf{SA}x - \mathbf{Sb}\|_2$ and $x^\star = \operatorname{argmin}_{x \in \mathbb{R}^d} \|\mathbf{A}x - \mathbf{b}\|_2$, we have

$$\|\mathbf{A}x_c - \mathbf{b}\|_2 \leq \|\mathbf{A}x_c - \mathbf{A}x^\star\|_2 + \|\mathbf{A}x^\star - \mathbf{b}\|_2 \leq \|\mathbf{A}x^\star - \mathbf{b}\|_2 + O(\|\mathbf{SA}x_c - \mathbf{SA}x^\star\|_2) \ .$$

Also we have that

$$\|\mathbf{SA}x_c - \mathbf{SA}x^\star\|_2 \leq \|\mathbf{SA}x_c - \mathbf{Sb}\|_2 + \|\mathbf{Sb} - \mathbf{SA}x^\star\|_2 \leq 2\|\mathbf{Sb} - \mathbf{SA}x^\star\|_2 \ ,$$

The only remaining thing is to bound $\|\mathbf{Sb} - \mathbf{SA}x^\star\|_2$. In fact, let $z = \mathbf{S}(\mathbf{A}x^\star - \mathbf{b})$, we have that

$$\mathbb{E}\left[\|\mathbf{S}(\mathbf{A}x^\star - \mathbf{b})\|_2^2\right] = \sum_i \mathbb{E}[z_i^2] = \frac{n}{k} \sum_{i=1}^{k} \sum_{j=1}^{n} \frac{1}{n} (\mathbf{A}x_j^\star - \mathbf{b})^2 = \|\mathbf{A}x^\star - \mathbf{b}\|_2^2$$

Since we have that $\mathbb{E}\left[\|\mathbf{SA}x^\star - \mathbf{Sb}\|_2^2\right] = \|\mathbf{A}x^\star - \mathbf{b}\|_2^2$, then by Markov's inequality we have that with high constant probability, $\|\mathbf{SA}x^\star - \mathbf{Sb}\|_2^2 \leq O(1)\|\mathbf{A}x^\star - \mathbf{b}\|_2^2$, which means that $\|\mathbf{SA}x_c - \mathbf{SA}x^\star\|_2 \leq O(\|\mathbf{A}x^\star - \mathbf{b}\|_2)$. Put everything together and by taking a union bound, we have that with high constant probability

$$\|\mathbf{A}x_c - \mathbf{b}\|_2 \leq C\|\mathbf{A}x^\star - \mathbf{b}\|_2$$

for some constant C. □

Proof of the proposition. Since $\operatorname{rank}(\mathbf{H}') = d$, we can assume $\mathbf{H}' = \mathbf{AB}$ where $\mathbf{A} \in \mathbb{R}^{D \times d}$ and $\mathbf{B} \in \mathbb{R}^{d \times n}$. Consider the regression problem $\min_x \|\mathbf{A}x - h_i\|_2$, from $\|\mathbf{A}b_i - h_i\|_2 = \|h_i' - h_i\|_2 \leq \varepsilon$ we have $\min_x \|\mathbf{A}x - h_i\|_2 \leq \varepsilon$.

Let $\mathbf{S}$ be the sampling and rescaling matrix with respect $\ell_i(\mathbf{A})$ with $O(d \log d)$ rows. From Lemma C.3 we have with probability at least 0.99 we have $\min_x \|\mathbf{SA}x - \mathbf{S}h_i\|_2 \leq C \min_x \|\mathbf{A}x - h_i\|_2 \leq C\varepsilon$ and here $x^i = \operatorname{argmin}_x \|\mathbf{SA}x - \mathbf{S}h_i\|_2 = (\mathbf{SA})^- \mathbf{S}h_i$. Let $I_{\mathbf{S}}$ denote the indices in $[n]$ where this event happens for h_i. Then we have $\mathbb{E}\left[|I_{\mathbf{S}}|\right] \geq 0.99n$, which means that there exists one $\mathbf{S}$ which makes $|I| \geq 0.99n$.

Now, taking this $\mathbf{S}$ and let $\mathbf{U} = \mathbf{A}(\mathbf{SA}^-)$, we have for every $i \in I_{\mathbf{S}}$,

$$\|\mathbf{USh}_i - h_i\|_2 = \|\mathbf{A}(\mathbf{SA}^-)h_i - h_i\| = \|\mathbf{SA}x - \mathbf{S}h_i\|_2 \leq \mathcal{O}(\varepsilon) \ .$$

This implies the matrix $\mathbf{U}$ and $\mathbf{S}$ is what we need. Note that the matrix $\mathbf{S}$ is a diagonal matrix but each diagonal entry has a rescaling weight, but we can put the weights into the row of $\mathbf{U}$ and make $\mathbf{S}$ as a row selection matrix (where each non-zero entry has value 1).

□

C.5 Proof of Theorem 4.4

Lemma C.4. *If in an attention mechanism we have $\|v_i\| \leq \eta$ for some $\eta = \mathcal{O}(1)$, $\exp(-c_1\varepsilon)a_{ij} \leq \widehat{a}_{ij} \leq \exp(c_1\varepsilon)a_{ij}$, $\forall i, j : |v_i \cdot v_j - \hat{v}_i \cdot \hat{v}_j| < c_2\varepsilon$, for some $c_1, c_2 = \mathcal{O}(\varepsilon)$, and we have $h_i = \sum_{u \in \mathcal{N}(i)} a_{iu} v_u$ and $\hat{h}_i = \sum_{u \in \mathcal{N}(i)} \widehat{a}_{iu} \hat{v}_u$, then there is some $t \in \mathcal{O}(1)$ that $|h_i \cdot h_j - \hat{h}_i \cdot \hat{h}_j| < t\varepsilon$.*

Proof. For any pair i, j, we have:

$$
\begin{aligned}
\hat{h}_i \cdot \hat{h}_j &= \sum_{u \in \mathcal{N}(i)} \widehat{a}_{iu} \hat{v}_u \sum_{\nu \in \mathcal{N}(j)} \widehat{a}_{j\nu} \hat{v}_\nu \\
&= \sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} \widehat{a}_{iu} \widehat{a}_{j\nu} (\hat{v}_u \cdot \hat{v}_\nu) \\
&< \sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} \exp(2c_1\varepsilon) a_{iu} a_{j\nu} (v_u \cdot v_\nu + c_2\varepsilon) \\
&= \exp(2c_1\varepsilon) \sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} a_{iu} a_{j\nu} (v_u \cdot v_\nu + c_2\varepsilon) \\
&= \exp(2c_1\varepsilon) \left(\left(\sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} a_{iu} a_{j\nu} v_u \cdot v_\nu \right) + c_2\varepsilon \left(\sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} a_{iu} a_{j\nu} \right) \right) \\
&= \exp(2c_1\varepsilon) \left(\left(\sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} a_{iu} a_{j\nu} v_u \cdot v_\nu \right) + c_2\varepsilon \left(\sum_{u \in \mathcal{N}(i)} a_{iu} \sum_{\nu \in \mathcal{N}(j)} a_{j\nu} \right) \right) \\
&= \exp(2c_1\varepsilon) \left(\left(\sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} a_{iu} a_{j\nu} v_u \cdot v_\nu \right) + c_2\varepsilon \right) \\
&= \exp(2c_1\varepsilon) \left(h_i \cdot h_j + c_2\varepsilon \right)
\end{aligned}
$$

Now, similarly, by lower bounding the attention scores and dot products of the $\mathbf{V}$ vectors from the smaller network, we will have:

$$\hat{h}_i \cdot \hat{h}_j > \exp(-2c_1\varepsilon) \left(h_i \cdot h_j - c_2\varepsilon \right)$$

Since $1 + x < \exp(x) < 1 + 2x$ for $0 < x \leq 1$ and $2c_1\varepsilon < 1$, and,

$$
\begin{aligned}
h_i \cdot h_j &= \sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} a_{iu} a_{j\nu} (v_u \cdot v_\nu) \\
&\leq \sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} a_{iu} a_{j\nu} \eta \\
&= \eta,
\end{aligned}
$$

If $h_i \cdot h_j + \alpha\varepsilon > 0$,

$$
\begin{aligned}
\hat{h}_i \cdot \hat{h}_j &< (1 + 8\alpha\varepsilon) \left(h_i \cdot h_j + \alpha\varepsilon \right) \\
&= h_i \cdot h_j + \varepsilon \left(8\alpha (h_i \cdot h_j) + 8\alpha^2\varepsilon + \alpha \right) \\
&\leq h_i \cdot h_j + \varepsilon \left(8\alpha^2 + \alpha + 8\alpha^2\varepsilon \right)
\end{aligned}
$$

Otherwise,

$$
\begin{aligned}
\hat{h}_i \cdot \hat{h}_j &< (1 + 4\alpha\varepsilon) \left(h_i \cdot h_j + \alpha\varepsilon \right) \\
&= h_i \cdot h_j + \varepsilon \left(4\alpha^2 + 4\alpha^2\varepsilon + \alpha \right).
\end{aligned}
$$

For the lower bound we have $1 - x \leq \exp(-x) \leq 1 - x/2$ for $0 < x < 1$, and,

$$
\begin{aligned}
h_i \cdot h_j &= \sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} a_{iu} a_{j\nu} (v_u \cdot v_\nu) \\
&\geq \sum_{u,\nu \in \mathcal{N}(i),\mathcal{N}(j)} a_{iu} a_{j\nu} - \alpha \\
&= -\alpha,
\end{aligned}
$$

Now, if $h_i \cdot h_j - \alpha\varepsilon > 0$,

$$\begin{aligned}\hat{h}_i \cdot \hat{h}_j &> (1 - 2c_1\varepsilon)\,(h_i \cdot h_j - \alpha\varepsilon) \\ &= h_i \cdot h_j - \varepsilon\left(-4\alpha(h_i \cdot h_j) + 4\alpha^2\varepsilon - \alpha\right) \\ &> h_i \cdot h_j - \varepsilon\left(4\alpha^2 + 4\alpha^2\varepsilon - \alpha\right)\end{aligned}$$

Otherwise:

$$\begin{aligned}\hat{h}_i \cdot \hat{h}_j &> (1 - 2\varepsilon\alpha)\,(h_i \cdot h_j - \alpha\varepsilon) \\ &= h_i \cdot h_j - \varepsilon\left(-2\alpha(h_i \cdot h_j) + 2\alpha^2\varepsilon - \alpha\right) \\ &> h_i \cdot h_j - \varepsilon\left(2\alpha^2 + 2\alpha^2\varepsilon - \alpha\right)\end{aligned}$$

since α is a positive value, if we get $t = 8\alpha^2 + \alpha + 8\alpha^2\varepsilon$, we have the proof for the lemma. □

Lemma C.5. *Assume for each i, $\|q_i^{(1)}\|, \|k_i^{(1)}\|, \|v_i^{(1)}\| \leq \sqrt{\alpha}$, and $d = \mathcal{O}(\frac{\log n}{\varepsilon^2})$ for any $0 < \varepsilon < \frac{1}{8\alpha}$. For the first Transformer layer's attention, there exist $\widehat{\mathbf{Q}}^{(1)}, \widehat{\mathbf{K}}^{(1)}, \widehat{\mathbf{V}}^{(1)} \in \mathbb{R}^{d\times D}$, such that for any pair i, j, $h_i^{(1/2)} \cdot h_j^{(1/2)} - t\varepsilon \leq \widehat{h}_i^{(1/2)} \cdot \widehat{h}_j^{(1/2)} \leq h_i^{(1/2)} \cdot h_j^{(1/2)} + t\varepsilon$ for a constant $t = \mathcal{O}(1)$.*

Proof. As part of proof for Theorem 3.3, we saw that for the narrow network in the first layer we have $\exp(-2\alpha\varepsilon)a_{ij}^{(1)} \leq \widehat{a}_{ij}^{(1)} \leq \exp(2\alpha\varepsilon)a_{ij}^{(1)}$. Now, if we consider n vectors from $\mathbf{V}^{(1)}$, by the JL-Transform, we know there exists a linear map f_V, with weight matrix $\mathbf{M}_V \in \mathbb{R}^{d\times D}$, such that:

$$\forall i, j : |v_i^{(1)} \cdot v_j^{(1)} - f(v_i^{(1)}) \cdot f(v_j^{(1)})| < \alpha\varepsilon.$$

Now, if we consider $\widehat{\mathbf{W}}^{(1)} = \mathbf{M}_V\mathbf{W}^{(1)}$, we will have:

$$\forall i, j : |v_i^{(1)} \cdot v_j^{(1)} - \hat{v}_i^{(1)} \cdot \hat{v}_j^{(1)}| < \alpha\varepsilon.$$

Now, since the conditions of the Lemma C.4 are correct here, we have the proof of the lemma. □

Proof of the Theorem. If $d = \mathcal{O}(\frac{\log n}{\varepsilon^2})$ we saw from Lemma C.5 that we can have $\mathbf{Q}^{(1)}, \mathbf{K}^{(1)}$, and $\mathbf{V}^{(1)}$ all in $\mathbb{R}^{d\times n}$ in a way that for each i, j we have $|\hat{h}_i^{1/2} \cdot \hat{h}_j^{1/2} - h_i^{1/2} \cdot h_j^{1/2}| < t\varepsilon$ for some constant $t = \mathcal{O}(1)$.

If input matrix $\mathbf{X}$ has a low-rank approximation $\bar{X}$ such that $\mathrm{rank}(\bar{X}) \leq d$ and for each i, $\|X_i - \bar{X}_i\| \leq \varepsilon$, we will prove the following statement first:

Lemma C.6. *If $\widehat{\mathbf{H}}^{(\ell)} \in \mathbb{R}^{d\times n}$, is the input vector of a narrow Transformer layer in a way that there exists a $\mathbf{U}^{(\ell)} \in \mathbb{R}^{D\times d}$ that $\forall i : \|\mathbf{U}^{(\ell)}\hat{h}_i^{(\ell)} - h_i^{(\ell)}\| \leq \varepsilon^{(\ell)}$ for a sufficiently small $\varepsilon^{(\ell)}$, we can have $\widehat{\mathbf{W}}_Q, \widehat{\mathbf{W}}_K$, and $\widehat{\mathbf{W}}_V \in \mathbb{R}^{d\times d}$ that the following conditions hold:*

1. *$\forall i, j : h_i^{(\ell+1/2)} \cdot h_j^{(\ell+1/2)} - t\varepsilon^{(\ell)} \leq \hat{h}_i^{(\ell+1/2)} \cdot \hat{h}_j^{(\ell+1/2)} \leq h_i^{(\ell+1/2)} \cdot h_j^{(\ell+1/2)} + t\varepsilon^{(\ell)}$ for some $t = \mathcal{O}(1)$.*
2. *For any attention score $\frac{a_{ij}^{(\ell)}}{\widehat{a}_{ij}^{(\ell)}} = 1 + \mathcal{O}(\varepsilon^{(\ell)})$.*

Proof of Lemma C.6: In this proof, we will occasionally emit the ℓ superscripts, since all the vectors are in the same layer.

According to Theorem 4.2, we can construct $\widehat{\mathbf{W}}_Q$ and $\widehat{\mathbf{W}}_K \in \mathbb{R}^{d\times d}$ that: $\exp(-2\varepsilon_a) \leq \frac{\widehat{a}_{ij}}{a_{ij}} \leq \exp(2\varepsilon_a)$, where $\varepsilon_a = \beta(2\sqrt{\alpha} + \beta)\varepsilon^{(\ell)}$. And for a small enough $\varepsilon^{(\ell)}$ we have $2\varepsilon_a < 1$, thus $\exp(2\varepsilon_a) < 1 + 2\varepsilon_a$ and $\exp(-2\varepsilon_a) > 1 - \varepsilon_a$, and for any i, j, $a_{i,j}^{(\ell)} \leq 1$. Thus for any i and j, $\frac{a_{i,j}^{(\ell)}}{\widehat{a}_{i,j}^{(\ell)}} = 1 + \mathcal{O}(\varepsilon_a) = 1 + \mathcal{O}(\varepsilon^{(\ell)})$.

For the $\widehat{\mathbf{W}}_V$, we want to make it in a way that $\forall i, j : |\hat{v}_i \cdot \hat{v}_j - v_i \cdot v_j| = \mathcal{O}(\varepsilon^{(\ell)})$.

We have $v_i \cdot v_j = v_i^\mathsf{T} v_j = h_i^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V h_j$. And also:

$$
\begin{aligned}
&|\hat{h}_i^\mathsf{T} \mathbf{U}^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V \mathbf{U} \hat{h}_j - h_i^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V h_j| = |(\mathbf{U}\hat{h}_i)^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V (\mathbf{U}\hat{h}_j) - h_i^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V h_j| \\
&= |(\mathbf{U}\hat{h}_i)^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V (\mathbf{U}\hat{h}_j) - (\mathbf{U}\hat{h}_i)^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V h_j + (\mathbf{U}\hat{h}_i)^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V h_j - h_i^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V h_j| \\
&= |(\mathbf{U}\hat{h}_i)^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V (\mathbf{U}\hat{h}_j - h_j) + (\mathbf{U}\hat{h}_i - h_i)^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V h_j| \\
&\leq |(\mathbf{U}\hat{h}_i)^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V (\mathbf{U}\hat{h}_j - h_j)| + |(\mathbf{U}\hat{h}_i - h_i)^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V h_j| \\
&\leq (\sqrt{\alpha} + \varepsilon^{(\ell)}) \beta^2 \varepsilon^{(\ell)} + \varepsilon^{(\ell)} \beta^2 \sqrt{\alpha} \\
&\leq (2\beta^2 \sqrt{\alpha} + \beta^2) \varepsilon^{(\ell)}.
\end{aligned}
$$

Now, if we take $\widehat{\mathbf{W}}_V = \mathbf{U}^\mathsf{T} \mathbf{W}_V^\mathsf{T} \mathbf{W}_V \mathbf{U}$, we have $|\hat{v}_i \cdot \hat{v}_j - v_i \cdot v_j| \leq (2\beta^2 \sqrt{\alpha} + \beta^2) \varepsilon^{(\ell)}$. For the brevity we define $\varepsilon_v = (2\beta^2 \sqrt{\alpha} + \beta^2) \varepsilon^{(\ell)}$.

With similar calculations as Lemma C.5, we have:

$$
\forall i, j : h_i^{(\ell+1/2)} \cdot h_j^{(\ell+1/2)} - t\varepsilon^{(\ell)} \leq \hat{h}_i^{(\ell+1/2)} \cdot \hat{h}_j^{(\ell+1/2)} \leq h_i^{(\ell+1/2)} \cdot h_j^{(\ell+1/2)} + t\varepsilon^{(\ell)}
$$

for some $t \in \mathcal{O}(1)$. □

We will inductively show that the initial assumption described in that theorem holds, and thus the construction of those mappings are feasible.

In layer ℓ for each i and j, assume we have $|\hat{h}_i^{(\ell+1/2)} \cdot \hat{h}_i^{(\ell+1/2)} - h_i^{(\ell+1/2)} \cdot h_j^{(\ell+1/2)}| \leq t^{(\ell)} \varepsilon$ for some constant $t^{(\ell)}$. The base of this induction works because of the construction and proves in Lemma C.5 and Theorem 4.2.

Based on the clustering assumptions we prove the following lemmas that will help us in rest of the proof:

Lemma C.7. *If h_i and h_j are in the same cluster with cluster center c_a, we have $\frac{h_i \cdot h_j}{\|c_a\|^2} = 1 + \mathcal{O}(\varepsilon)$.*

Proof of Lemma C.7:

$$
\begin{aligned}
h_i \cdot h_j &= ((h_i - c_a) + c_a) \cdot ((h_j - c_a) + c_a) \\
&\leq \|h_i - c_a\| \|c_a\| + \|h_j - c_a\| \|c_a\| + \|h_i - c_a\| \|h_j - c_a\| + \|c_a\|^2 \\
&\leq 2\varepsilon \|c_a\|^2 + \varepsilon^2 \|c_a\|^2 + \|c_a\|^2 \\
&= (1 + 2\varepsilon + \varepsilon^2) \|c_a\|^2 \leq (1 + 3\varepsilon) \|c_a\|^2.
\end{aligned}
$$

Also, similarly we have:

$$
\begin{aligned}
h_i \cdot h_j &= ((h_i - c_a) + c_a) \cdot ((h_j - c_a) + c_a) \\
&\leq -\|h_i - c_a\| \|c_a\| - \|h_j - c_a\| \|c_a\| - \|h_i - c_a\| \|h_j - c_a\| + \|c_a\|^2 \\
&\leq -2\varepsilon \|c_a\|^2 - \varepsilon^2 \|c_a\|^2 + \|c_a\|^2 \\
&= (1 - 2\varepsilon - \varepsilon^2) \|c_a\|^2 \leq (1 - 3\varepsilon) \|c_a\|^2.
\end{aligned}
$$

Thus, we have $1 - 3\varepsilon \leq \frac{h_i \cdot h_j}{\|c_a\|^2} \leq 1 + 3\varepsilon$. □

Corollary C.8. *If h_i and h_j are in the same cluster with cluster center c_a, we have $\frac{\hat{h}_i \cdot \hat{h}_j}{\|c_a\|^2} = 1 + \mathcal{O}(\varepsilon)$.*

Proof of Corollary C.8: We saw that $\frac{h_i \cdot h_j}{\|c_a\|^2} = 1 + \mathcal{O}(\varepsilon)$ and we know $|\hat{h}_i \cdot \hat{h}_j - h_i \cdot h_j| < t^{(\ell)} \varepsilon$. Since $\|c_a\|^2 > \gamma_1^2$ the corollary is correct. □

Lemma C.9. *If h_i and h_j are from different clusters with cluster centers c_a and c_b accordingly, we have $\frac{h_i \cdot h_j}{\|c_a\|^2} \leq 0.5 + \mathcal{O}(\varepsilon)$.*

Proof of Lemma C.9:

$$\begin{aligned} h_i \cdot h_j &= ((h_i - c_a) + c_a) \cdot ((h_j - c_b) + c_b) \\ &\leq \|h_i - c_a\|\|c_b\| + \|h_j - c_b\|\|c_a\| + \|h_i - c_a\|\|h_j - c_b\| + c_a \cdot c_b \\ &\leq 2\varepsilon\|c_a\|\|c_b\| + \varepsilon^2\|c_a\|\|c_b\| + \gamma_1^2/2 \\ &= (2\varepsilon + \varepsilon^2)\gamma_2^2 + 1/2\gamma_1^2 \leq 1/2\|c_a\|^2 + 3\varepsilon\gamma_2^2. \end{aligned}$$

Thus we have $\frac{h_i \cdot h_j}{\|c_a\|^2} \leq 0.5 + 3\varepsilon\gamma_2^2/\|c_a\|^2 \leq 0.5 + 3\varepsilon\gamma_2^2/\gamma_1^2 = 0.5 + \mathcal{O}(\varepsilon)$. □

Corollary C.10. *If h_i and h_j are from different clusters with cluster centers c_a and c_b accordingly, we have $\frac{\hat{h}_i \cdot \hat{h}_j}{\|c_a\|^2} \leq 0.5 + \mathcal{O}(\varepsilon)$.*

Proof of Corollary C.10: Again this is an immediate result of the previous lemma and

$$|\hat{h}_i \cdot \hat{h}_j - h_i \cdot h_j| < t^{(\ell)}\varepsilon.$$

□

Now, we construct the feed forward layers following the attention layer: for constructing $\widehat{\mathbf{W}}_1^{(\ell)}$, as we do not have the centers of the clusters in the low dimension particularly, we select just one of the node representations from that cluster as $\hat{c}_a$. If the number of clusters is smaller than d we set cluster centers for these clusters to all zero vectors. We set the ath row of $\widehat{\mathbf{W}}_1^{(\ell)}$ to be $4\frac{\hat{c}_a}{\|c_a\|^2}$, we use bias of the linear mappings here and set it to $-3\|c_a\|^2$.

Now, for each $\hat{h}_i$ if it is in cluster a, we have $(\widehat{\mathbf{W}}_1^{(\ell)}\hat{h}_i)_a = 1 + \mathcal{O}(\varepsilon)$ due to Corollary C.8 and for a small enough ε, we have $\forall b \neq a : (\widehat{\mathbf{W}}_1^{(\ell)}\hat{h}_i)_b < 0$. Thus $\mathrm{ReLU}(\widehat{\mathbf{W}}_1^{(\ell)}\hat{h}_i)$ is almost one-hot vector, being absolutely zero in any index $b \neq a$ and have value $1 + \mathcal{O}(\varepsilon)$ in index a.

We will use these almost 1-hot vectors to create the embeddings using $\widehat{\mathbf{W}}_2^{(\ell)}$. But before doing that, we will show that the assumption will lead to having a low-rank approximation on the higher dimension network after the ReLU:

Lemma C.11. $\mathbf{H}^{(\ell+3/4)}$ *has a low-rank estimation $\bar{\mathbf{H}}^{(\ell+3/4)}$ of maximum rank d such that $\|h_i^{(\ell+3/4)} - \bar{h}_i^{(\ell+3/4)}\| = \mathcal{O}(\varepsilon)$.*

Proof. Since for each $h_i^{(\ell+1/2)}$ we have a c_a such that their distance is $\mathcal{O}(\varepsilon)$, and due to the maximum operator norm of $\mathbf{W}_1^{(\ell)}$ and Lipschitzness of ReLU, we have $\|\mathrm{ReLU}(W_1^{(\ell)}h_i) - \mathrm{ReLU}(W_1^{(\ell)}c_a)\| = \mathcal{O}(\varepsilon)$. Now we can make $\bar{\mathbf{H}}^{(\ell+3/4)}$ in this way that instead of each $h_i^{(\ell+3/4)}$ replace it with the mapping of its cluster center $\mathrm{ReLU}(W_1^{(\ell)}c_a)$. Now, $\bar{\mathbf{H}}^{(\ell+3/4)}$ has maximum d distinct rows, thus its rank is maximum d. □

Since $\bar{\mathbf{H}}$ is of rank maximum d, we can have $\mathbf{U} \in \mathbb{R}^{D \times d}$ and $\mathbf{\Lambda} \in d \times D$ that $\bar{\mathbf{H}} = \mathbf{U}\mathbf{\Lambda}\bar{\mathbf{H}}$. Now, for creating $\widehat{\mathbf{W}}_2^{(\ell)}$ we make a $d \times d$ matrix having column a as $\mathbf{\Lambda}\mathbf{W}_2^{(\ell)}\,\mathrm{ReLU}(\mathbf{W}_1^{(\ell)}c_a)$. Now, each row of $\widehat{\mathbf{H}}^{(\ell+1)}$ will be a $(1 + \mathcal{O}(\varepsilon))\mathbf{W}_2^{(\ell)}\,\mathrm{ReLU}(\mathbf{W}_1 c_a)$ for some c_a, thus each column of $U\widehat{\mathbf{H}}^{(\ell+1)}$ will estimate the corresponding column of $\mathbf{H}^{(\ell+1)}$ with maximum norm-2 distance of $\mathcal{O}(\varepsilon)$.

This will satisfy the required conditions for the start of a layer as described in Theorem 4.2. Thus, we can create low-rank matrices □

D Experiment Details

D.1 Datasets

Below, we provide descriptions of the datasets on which we conduct experiments. Summary statistics of these datasets are provided in Table 3.

Photo This dataset is part of the Amazon co-purchase graph (McAuley et al., 2015), where each node represents a product on the Amazon website, and an edge indicates that the corresponding products were frequently purchased together. Node features are derived from a bag-of-words summary of the reviews for each product. The task is to classify the nodes into different product categories (Shchur et al., 2018). We use a random train/validation/test split with a ratio of 0.6/0.2/0.2 for training.

Minesweeper This dataset was first introduced in Platonov et al. (2023). The dataset is a graph representation of the 100x100 grid from the Minesweeper game. Each node represents a cell, and edges connect to the eight neighboring cells. 20% of the nodes are marked as mines. The node features are the one-hot encoding of the number of mines among the neighbors. For 50% of the nodes, the features are unknown and indicated by a separate binary feature.

Tolokers This dataset was also first introduced in the Platonov et al. (2023). Tolokers is a graph representation of workers on a crowdsourcing platform called Toloka. Two nodes are connected if the workers have worked on the same project. Node features are based on the worker's task performance statistics and other profile information. The task is to predict which nodes have been banned for a project.

Table 3: Dataset statistics. The reported number of edges is the number of directed edges, which will be twice the number of actual edges for the undirected graphs. The homophily score is a metric to measure what ratio of the neighbor nodes are from the same class.

Dataset	Nodes	Edges	Homophily Score	Node Features	Classes	Metric
Amazon Photo	7,487	238,162	0.772	745	8	Accuracy
Minesweeper	10,000	78,804	0.009	7	2	AUC
Tolokers	11,758	1,038,000	0.187	10	10	AUC

D.2 Network Details

We train the Transformer model only on the graph edges here. Another alternative could be to train it with an augmented expander graph as suggested in Shirzad et al. (2023). The model implementation is very similar to the formulation explained in Section 2; however, for performance purposes, there are a few small changes:

1. We follow the standard attention implementation and add $\frac{1}{\sqrt{D}}$ normalization to the attention scores. This works much better in practice.
2. We use skip connections to improve the training process.
3. We use batch normalization instead of layer normalization, as our experiments and the results from the Exphormer model showed that batch normalization actually works better than layer normalization in Graph Transformers.

For all datasets and for both large and small models we used four layers of the network.

D.3 Hyperparameter Search

For both large and small models, we performed a grid search on the base learning rate and the number of epochs, selecting the configuration with the highest average accuracy or AUC based on the standard metric for the dataset. We chose the base learning rate from $\{0.1, 0.01, 0.001\}$ and the number of epochs from $\{50, 100, 150, 200\}$. We used AdamW (Loshchilov et al., 2017) for optimization and a cosine learning rate scheduler (Loshchilov and Hutter, 2016), as is common in training Transformers and Graph Transformers. We did twenty runs for larger networks to measure the mean and 100 runs for the small networks.

D.4 Operator Norms and Vector Norms

To validate the assumptions in Section 2, we measured the average operator norm of the linear mappings and the norm of the vectors for the input of each layer, and then averaged these values across the layers in three datasets we experimented on. Results are provided in Table 1. The resulting numbers are reasonably small, allowing us to assume them to be $\mathcal{O}(1)$ in the theory.

D.5 Comparing the Results to Baselines

To understand the significance of the results and to provide context, we compare the performance of our compressed network with simple baselines, namely MLP, GCN, Nodeformer, and Exphormer models. Results are provided in Table 4. Our goal is to demonstrate that a very small network with a hidden dimension of 4 can achieve much better results than these methods, given that we can properly compress the large network.

Table 4: Comparing found results with some baselines, showing the models are well-trained and have competitive results.

Dataset	Tolokers	Minesweeper	Photo
MLP	0.730 ± 0.0106	0.509 ± 0.014	0.696 ± 0.038
GCN	0.836 ± 0.007	0.898 ± 0.005	0.927 ± 0.002
NodeFormer	0.781 ± 0.001	0.867 ± 0.009	0.935 ± 0.004
Exphormer	0.835 ± 0.003	0.923 ± 0.006	0.954 ± 0.002
Average Large Network	0.844 ± 0.002	0.943 ± 0.001	0.953 ± 0.004
Average Small Network	0.821 ± 0.011	0.886 ± 0.054	0.910 ± 0.016
Max Small Network	0.844	0.938	0.944