When Do Transformers Outperform Feedforward and Recurrent Networks? A Statistical Perspective

Alireza Mousavi-Hosseini MOUSAVI@CS.TORONTO.EDU
University of Toronto and Vector Institute

Clayton Sanford CHSANFORD@GOOGLE.COM
Google Research

Denny Wu DENNYWU@NYU.EDU
New York University and Flatiron Institute

Murat A. Erdogdu ERDOGDU@CS.TORONTO.EDU
University of Toronto and Vector Institute

Abstract

Theoretical efforts to prove advantages of Transformers in comparison with classical architectures such as feedforward and recurrent neural networks have mostly focused on representational power. In this work, we take an alternative perspective and prove that even with infinite compute, feedforward and recurrent networks may suffer from larger sample complexity compared to Transformers, as the latter can adapt to a form of *dynamic sparsity*. Specifically, we consider a sequence-to-sequence data generating model on sequences of length N, where the output at each position only depends on $q \ll N$ relevant tokens, and the positions of these tokens are described in the input prompt. We prove that a single-layer Transformer can learn this model if and only if its number of attention heads is at least q, in which case it achieves a sample complexity almost independent of N, while recurrent networks require $N^{\Omega(1)}$ samples on the same problem. If we simplify this model, recurrent networks may achieve a complexity almost independent of N, while feedforward networks still require N samples. Our proposed sparse retrieval model illustrates a natural hierarchy in sample complexity across these architectures.

1. Introduction

The theoretical efforts surrounding the success of Transformers [26] have so far demonstrated various capabilities like in-context learning [3, 8, 18, 27, 29, and others] and chain-of-thought prompting along with its benefits [15, 17, 20, 21, and others] in various settings. There are fewer works that provide specific benefits of Transformers in comparison with feedforward and recurrent architectures. On the approximation side, there are tasks that Transformers can solve with size logarithmic in the input, while alternative architectures require polynomial size [24, 25]. Based on these results, [28] showed a separation between Transformers and feedforward networks by providing further optimization guarantees for gradient-based training of Transformers on a sparse token selection task.

While most prior works focused on the approximation separation between Transformers and feedforward networks (FFNs), in this work we focus on a purely statistical separation, and ask:

What function class can Transformers learn with fewer samples compared to feedforward and recurrent networks, even with infinite computational resources?

[16] approached the above problem with random features, where the query-key matrix for the attention and the first layer weights for the two-layer feedforward network were fixed at random initialization. However, this only presents a partial picture, as neural networks can learn a significantly larger class of functions once "feature learning" is allowed, i.e., parameters are trained to adapt to the structure of the underlying task [1, 6, 7, 10, 12, 13, 22].

We evaluate the statistical efficiency of Transformers and alternative architectures by characterizing how the sample complexity depends on the input sequence length. A benign length dependence (e.g., sublinear) signifies the ability to achieve low test error in longer sequences, which intuitively connects to the *length generalization* capability [5]. Our generalization bounds for bounded-norm Transformers — along with our contrasts to RNNs and feedforward neural networks — provide theoretical insights into the statistical advantages of Transformers and lay the foundation for future rigorous investigations of length generalization.

1.1. Our Contributions

We study the q-Sparse Token Regression ($q\mathrm{STR}$) data generating model, a sequence-to-sequence model where the output at every position depends dynamically on a sparse subset of the input tokens. We prove that by employing the attention layer to retrieve relevant tokens at each position, single-layer Transformers can adapt to this dynamic sparsity, and learn $q\mathrm{STR}$ with a sample complexity almost independent of the length of input sequence N. On the other hand, we develop a new metric-entropy-based argument to derive norm and parameter-count lower bounds for RNNs that lead to a sample complexity lower bound of order $N^{\Omega(1)}$ for RNNs. Further, we show that RNNs can learn a subset of $q\mathrm{STR}$ where the output is a constant sequence, which we call simple-$q\mathrm{STR}$, with a sample complexity polylogarithmic in N. Finally, we develop a lower bound technique for feedforward networks (FFNs) that takes advantage of the fully connected projection of the first layer to obtain a sample complexity lower bound linear in N, even when learning simple-$q\mathrm{STR}$ models. The following theorem summarizes our main contributions.

Theorem 1 (Informal) *We have the following hierarchy of statistical efficiency for learning* $q\mathrm{STR}$.

- *A single-layer Transformer with* $H \geq q$ *heads can learn* $q\mathrm{STR}$ *with sample complexity almost independent of* N*, and cannot learn* $q\mathrm{STR}$ *when* $H < q$ *even with infinitely many samples.*
- *RNNs can learn* simple-$q\mathrm{STR}$ *with sample size almost independent of* N*, but require at least* $\Omega(N^c)$ *samples for some constant* $c > 0$ *to learn a generic* $q\mathrm{STR}$ *model, regardless of their size.*
- *Feedforward neural networks, regardless of their size, require* $\Omega(Nd)$ *samples to learn even* simple-$q\mathrm{STR}$ *models, where* d *is input token dimension.*

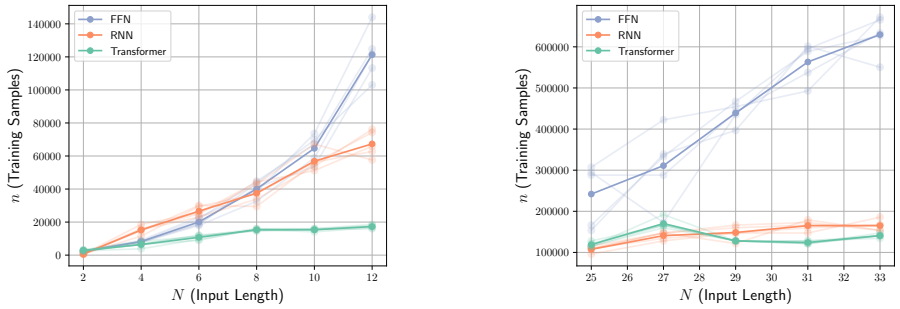
We empirically validate the intuitions from Theorem 1 in Figure 1.

2. Problem Setup

Statistical Model. In this paper, we will focus on the ability of different architectures for learning the following data generating model.

Definition 2 (q-Sparse Token Regression) *Suppose* $\boldsymbol{p}, \boldsymbol{y} \sim \mathcal{P}$ *where*

$$\boldsymbol{p} = \left(\begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{t}_1 \end{pmatrix}, \ldots, \begin{pmatrix} \boldsymbol{x}_N \\ \boldsymbol{t}_N \end{pmatrix} \right),$$



(*a*) Sample complexity (1STR) (*b*) Sample complexity (Simple-1STR)

Figure 1: Number of samples to reach a certain test MSE loss threshold while training with online AdamW. We consider (a) 1STR with threshold 0.7 and (b) simple-1STR with threshold 0.02, averaged over 5 experiments. We use a linear link function, standard Gaussian input, $d = 10$ and $d_e = \lfloor 5 \log(N) \rfloor$. Positional encodings are sampled uniformly from the unit hypercube.

$\boldsymbol{t}_i \in [N]^q$ and $\boldsymbol{x}_i \in \mathbb{R}^d$ for $i \in [N]$. In the q-sparse token regression (qSTR) data generating model, the output is given by $\boldsymbol{y} = (y_1, \ldots, y_N)^\top \in \mathbb{R}^N$, where

$$y_i = g(\boldsymbol{x}_{t_{i1}}, \ldots, \boldsymbol{x}_{t_{iq}}),$$

for some $g : \mathbb{R}^{qd} \to \mathbb{R}$. We call this model simple-qSTR *if the data distribution is such that $\boldsymbol{t}_i = \boldsymbol{t}$ for all $i \in [N]$ and some $\boldsymbol{t}$ drawn from $[N]^q$.*

The above defines a class of sequence-to-sequence functions, where the label at position i in the output sequence depends only on a subsequence of size q of the input data, determined by the set of indices $\boldsymbol{t}_i$. $\boldsymbol{p}$ in the above definition denotes the prompt or context. Given the large context length of modern architectures, we are interested in a setting where $q \ll N$. In this setting, the answer at each position only depends on a few tokens, however the tokens it depends on change based on the context. Therefore, we seek architectures that are *adaptive* to this form of *dynamic sparsity* in the true data generating process, with computational and sample complexity independent of N. As a special case, choosing g as the tokens' mean recovers the *sparse averaging* model proposed in [24], where the authors separate the representational capacity of Transformers and other architectures.

Throughout the paper, we put mild assumptions on the data distribution. Specifically, we assume $\boldsymbol{x}$ is sub-Gaussian and g grows at most polynomially and is approximated up to L_2 error $\varepsilon_{\texttt{2NN}}$ by a two-layer feedforward network with width m_g. These assumptions are formalized in Appendix A.

While Empirical Risk Minimization (ERM) is a standard abstract learning algorithm to use for generalization analysis, its standard formalizations use risk functions for scalar-valued predictions. We measure the performance of different architectures in terms of the following population risk

$$R^{\texttt{arc}}(\boldsymbol{\Theta}) := \frac{1}{N} \mathbb{E}\left[\sum_{j=1}^{N} (\hat{y}_{\texttt{arc}}(\boldsymbol{p}; \boldsymbol{\Theta})_j - y_j)^2\right] = \frac{1}{N} \mathbb{E}\left[\|\hat{\boldsymbol{y}}_{\texttt{arc}}(\boldsymbol{p}; \boldsymbol{\Theta}) - \boldsymbol{y}\|_2^2\right],$$

where `arc` denotes a general architecture, and $\hat{\boldsymbol{y}}_{\texttt{arc}}(\cdot; \boldsymbol{\Theta})$ denotes the output of the model parameterized by weight vector $\boldsymbol{\Theta}$. In Appendix A, we formalize several notions of ERM suitable for sequential risk formulations.

3. Transformers

A single-layer Transformer is composed of an attention and a parallel feedforward layer. We consider a standard theoretical formalization of a single-layer transformer with q heads and width m_g for the feedforward units, where m_g is defined in Assumption 2. We formally define the architecture in Appendix B.1. We consider the following parameter class $\Theta_{\text{TR}} = \{\|\text{vec}(\mathbf{\Theta})\|_2 \leq R\}$, and provide a learning guarantee for empirical risk minimizers over Θ_{TR}, with its proof (including the choice of R) deferred to Appendix B.2.

Theorem 3 *Let* $\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}\in\Theta_{\text{TR}}} \hat{R}_n^{\text{TR}}(\mathbf{\Theta})$. *Under Assumptions 1 to 3, we have*

$$R^{\text{TR}}(\hat{\mathbf{\Theta}}_n) \lesssim \varepsilon_{\text{2NN}} + \tilde{\mathcal{O}}\left(C_1\sqrt{\frac{m_g q(d+q) + q^3 + qd^2}{n}}\right)$$

where $C_1 = R^2 qd$, *with probability at least* $1 - n^{-c}$ *for some absolute constant* $c > 0$.

Note that the sample complexity above depends on N only up to log factors. By incorporating additional structure in the ERM solution, it is possible to obtain improved sample complexities. A close study of the optimization dynamics may reveal such additional structure in the solution reached by gradient-based methods, pushing the sample complexity closer to the information-theoretic limit of $\Omega(qd)$. Figure 2 demonstrates that the attention weights achieved through standard optimization of a Transformer match our theoretical constructions – see Equation (6) – even while maintaining separate $\boldsymbol{W}_Q$ and $\boldsymbol{W}_K$ during training (we use the 1STR setup of Figure 1 with $N = 100$). We leave the study of optimization dynamics and the resulting sample complexity for future work.

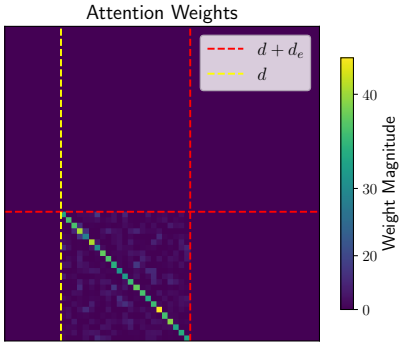


Figure 2: Trained attention weights match our theoretical construction (6).

3.1. Limitations of Transformers with Few Heads

In this section, we will demonstrate that $H \geq \Omega(q)$ is required to learn qSTR models, even from a pure approximation perspective, i.e. with access to population distribution. In contrast to [4], we do not put any assumptions on the rank of the key-query projections, i.e. our lower bound applies even when the key-query projection matrix is full-rank.

Proposition 4 *Consider a* qSTR *model where* $y_i = \frac{1}{\sqrt{qd}}\sum_{j=1}^{q}\left(\|\boldsymbol{x}_{t_{ij}}\|^2 - \mathbb{E}\left[\|\boldsymbol{x}_{t_{ij}}\|^2\right]\right)$, $\boldsymbol{x}_i \sim \mathcal{N}(0, \mathbf{\Sigma}_i)$ *such that* $\mathbf{\Sigma}_i = \mathbf{I}_d$ *for* $i < N/2$ *and* $\mathbf{\Sigma}_i = 0$ *for* $i \geq N/2$. *Then, there exists a distribution over* $(\boldsymbol{t}_i)_{i\in[N]}$ *such that for any choice of* $\mathbf{\Theta}_{\text{TR}}$ *(including arbitrary* $\{\boldsymbol{W}_{\text{QK}}^{(h)}\}_{h\in[H]}$*), we have*

$$\frac{1}{N}\mathbb{E}\left[\|\boldsymbol{y} - \hat{\boldsymbol{y}}_{\text{TR}}(\boldsymbol{p}; \mathbf{\Theta}_{\text{TR}})\|_2^2\right] \geq 1 - \frac{(q+d)H}{qd}.$$

Remark 5 *We highlight the importance of the nonlinear dependence of* y_i *on* $\boldsymbol{x}$ *for the above lower bound. In particular, for the sparse token averaging task introduced in [24], a single-head attention layer with a carefully constructed embedding suffices for approximation.*

The above proposition implies that given sufficiently large dimensionality $d \gg q$, approximation alone necessitates at least $H = \Omega(q)$ heads. In Appendix B.3, we present the proof of Proposition 4, along with Proposition 20 which establishes an exact lower bound $H \geq q$ for all $d \geq 1$, at the expense of additional restrictions on the query-key projection matrix.

4. Recurrent Neural Networks

In this section, we first provide positive results for RNNs by proving that they can learn simple-qSTR with a sample complexity only polylogarithmic in N, thus establishing a separation in their learning capability from feedforward networks. For this upper bound, we use bidirectional RNNs with deep transitions [23], formally introduced in Appendix C.1.

Theorem 6 (RNNs can learn simple-qSTR) *Let $\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta} \in \Theta_{\text{RNN}}} \hat{R}_n^{\text{RNN}}(\boldsymbol{\Theta})$ (with Θ_{RNN} defined in Equation (10)). Suppose Assumptions 1 to 3 hold with the simple-qSTR model, i.e. $\boldsymbol{t}_i = \boldsymbol{t}$ for all $i \in [N]$ and some $\boldsymbol{t}$ drawn from $[N]^q$. Then, with proper hyperparameters in Θ_{RNN} (see Appendix C.1), we obtain*

$$R^{\text{RNN}}(\hat{\boldsymbol{\Theta}}) \lesssim \varepsilon_{\text{2NN}} + \sqrt{\frac{\text{poly}(d, q, m_g, \varepsilon_{\text{2NN}}^{-1}, \log(nN))}{n}},$$

with probability at least $1 - n^{-c}$ for some absolute constant $c > 0$.

As desired, the above sample complexity depends on N only up to polylogarithmic factors. The completed proof can be found in Appendix C.

Next, we turn to general qSTR, where we provide a negative result on RNNs, proving that to learn such models their sample complexity must scale with $N^{\Omega(1)}$ regardless of model size, making them less statistically efficient than Transformers. Our lower bound covers a broad notion of bidirectional RNNs formalized in Appendix C.5, and includes the example in the upper bound.

Theorem 7 (RNNs can not learn qSTR) *Consider the 1STR model where $\boldsymbol{x} \sim \mathcal{N}(0, \mathbf{I}_{Nd})$ with a linear link function, i.e. $y_j = \langle \boldsymbol{u}, \boldsymbol{x}_{t_j} \rangle$ for some $\boldsymbol{u} \in \mathbb{S}^{d-1}$. Further, t_i is drawn independently from the rest of the prompt and uniformly from $[N]$ for all $i \in [N]$. Let $\hat{\boldsymbol{\Theta}}_\varepsilon$ be the min-norm ε-ERM of $\hat{R}_n^{\text{RNN}}$, defined in (4). Then, there exist absolute constants $c_1, c_2, c_3 > 0$ such that if $n \leq \mathcal{O}(N^{c_1})$, for any $\varepsilon \geq 0$, with probability at least c_2 over the training set,*

$$\frac{1}{N} \mathbb{E}\left[\left\| \hat{\boldsymbol{y}}_{\text{RNN}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}}_{n,\varepsilon}) - \boldsymbol{y} \right\|_2^2 \right] \geq c_3.$$

On our way to prove this theorem, we prove a novel representational lower bound for RNNs – Proposition 34 in Appendix C.5 – that captures both the number of parameters *and* the norm in the weights. This representational lower bound implies that an RNN that generalizes on the entire data distribution (hence approximates the 1STR model) requires a weight norm that scales with $\sqrt{N}$. On the other hand, we show that overfitting on the n training samples with zero empirical risk is possible with a $\text{poly}(n)$ weight norm. As a result, as long as $n \leq N^{c_1}$ for some small constant $c_1 > 0$, min-norm ε-ERM will choose models that overfit rather than generalize. The complete proof of Theorem 7 is presented in Appendix C.7.

5. Feedforward Neural Networks (FFNs)

In this section, we consider a general formulation of a feedforward network. Our only requirement will be that the first layer performs a fully-connected projection. The subsequent layers of the network can be arbitrarily implemented, e.g. using attention blocks or convolution filters. Specifically, the FFN implements the mapping $\boldsymbol{p} \mapsto f(\boldsymbol{T}, \boldsymbol{W}\boldsymbol{x})$ where $\boldsymbol{W} \in \mathbb{R}^{m_1 \times Nd}$ is the weight matrix in the first layer, $\boldsymbol{x} = (\boldsymbol{x}_1^\top, \dots, \boldsymbol{x}_N^\top)^\top \in \mathbb{R}^{Nd}$, and $f : [N]^{qN} \times \mathbb{R}^{m_1} \to \mathbb{R}^N$ implements the rest of the network. Unlike Transformers, here we give the network full information of $\boldsymbol{T} = (\boldsymbol{t}_1, \dots, \boldsymbol{t}_N)$, and in particular, it can implement arbitrary encodings of the position variables $\boldsymbol{t}_1, \dots, \boldsymbol{t}_N$. This formulation covers usual approaches where encodings of $\boldsymbol{t}$ are added to or concatenated with $\boldsymbol{x}$.

The class of algorithms we consider for training FFNs goes beyond ERM and includes stationary points of the training loss, thus covering outputs of first-order optimization algorithms. This class is formally introduced in Definition 41 in Appendix D.

For our negative result on feedforward networks, we can further restrict the class of qSTR models, and only consider simple-qSTR. The following minimax lower bound, with its proof deferred to Appendix C, shows that all algorithms in class $\mathcal{A}$ fail to learn even the subset of simple-qSTR models with a sample complexity sublinear in N.

Theorem 8 *Suppose $\boldsymbol{x} \sim \mathcal{N}(0, \mathbf{I}_{Nd})$, and consider the simple-*1STR *model with $t_{i1} = t_1$ for all $i \in [N]$, where t_1 is drawn independently and uniformly in $[N]$, and a linear link function, i.e. $y = \langle \boldsymbol{u}, \boldsymbol{x}_{t_1} \rangle$ for some $\boldsymbol{u} \in \mathbb{S}^{d-1}$. Let $\mathcal{A}$ be the class of algorithms in Definition 41. Then,*

$$\inf_{A \in \mathcal{A}} \sup_{\boldsymbol{u} \in \mathbb{S}^{d-1}} R^{\mathrm{FFN}}(f_{A(S_n)}, \boldsymbol{W}_{A(S_n)}) \geq 1 - \frac{n}{Nd},$$

with probability 1 *over the training set S_n.*

The main intuition in the proof of the above theorem follows from the stationarity property of Definition 41. With this property, the rows of the trained $\boldsymbol{W}$ will always be in the span of the training data $\boldsymbol{x}^{(i)}$ for $i \in [n]$, and this subspace can be too small to predict y, which by randomizing $\boldsymbol{u}$, can depend on all Nd target directions.

6. Conclusion

In this paper, we established a sample complexity separation between Transformers and baseline architectures, namely feedforward and recurrent networks, for learning sequence-to-sequence models where the output at each position depends on a sparse subset of input tokens described in the input itself, coined the qSTR model. We proved that Transformers can learn such a model with sample complexity almost independent of the length of the input sequence N, while feedforward and recurrent networks have sample complexity lower bounds of N and $N^{\Omega(1)}$, respectively. Further, we established a separation between FFNs and RNNs by proving that recurrent networks can learn the subset of simple-qSTR models where the output at all positions is identical, whereas feedforward networks require at least N samples. An important direction for future work is to develop an understanding of the optimization dynamics of Transformers to learn qSTR models, and to study sample complexity separations that highlight the role of depth in Transformers.

References

[1] Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 2552–2623. PMLR, 2023.

[2] Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 37:45614–45650, 2023.

[3] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

[4] Noah Amsel, Gilad Yehudai, and Joan Bruna. On the benefits of rank in attention layers. *arXiv preprint arXiv:2407.16153*, 2024.

[5] Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, 35:38546–38556, 2022.

[6] Jimmy Ba, Murat A Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation. *arXiv preprint arXiv:2205.01445*, 2022.

[7] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

[8] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36, 2023.

[9] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

[10] Alberto Bietti, Joan Bruna, Clayton Sanford, and Min Jae Song. Learning single-index models with shallow neural networks. In *Advances in Neural Information Processing Systems*, 2022.

[11] Minshuo Chen, Xingguo Li, and Tuo Zhao. On generalization bounds of a family of recurrent neural networks. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1233–1243. PMLR, 2020.

[12] Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural Networks can Learn Representations with Gradient Descent. In *Conference on Learning Theory*, 2022.

[13] Yatin Dandi, Florent Krzakala, Bruno Loureiro, Luca Pesce, and Ludovic Stephan. Learning two-layer neural networks, one (giant) step at a time. *arXiv preprint arXiv:2305.18270*, 2023.

[14] Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pages 5793–5831. PMLR, 2022.

[15] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36, 2023.

[16] Hengyu Fu, Tianyu Guo, Yu Bai, and Song Mei. What can a single attention layer learn? a study through the random features lens. *Advances in Neural Information Processing Systems*, 36, 2023.

[17] Juno Kim and Taiji Suzuki. Transformers provably solve parity efficiently with chain of thought. *arXiv preprint arXiv:2410.08633*, 2024.

[18] Juno Kim, Tai Nakamaki, and Taiji Suzuki. Transformers are minimax optimal nonparametric in-context learners. In *ICML 2024 Workshop on In-Context Learning*, 2024. URL https://openreview.net/forum?id=WjrKBQTWKp.

[19] Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, pages 19565–19594. PMLR, 2023.

[20] Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations*, 2024.

[21] William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. In *The Twelfth International Conference on Learning Representations*, 2024.

[22] Alireza Mousavi-Hosseini, Denny Wu, and Murat A Erdogdu. Learning multi-index models with neural networks via mean-field langevin dynamics. *arXiv preprint arXiv:2408.07254*, 2024.

[23] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

[24] Clayton Sanford, Daniel J Hsu, and Matus Telgarsky. Representational strengths and limitations of transformers. *Advances in Neural Information Processing Systems*, 36, 2023.

[25] Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[27] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

[28] Zixuan Wang, Stanley Wei, Daniel Hsu, and Jason D. Lee. Transformers provably learn sparse token selection while fully-connected nets cannot. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[29] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

Appendix A. Detailed Assumptions and Learning Procedure

Here, we stay our precise assumptions on the data distribution.

Assumption 1 *Suppose* $\mathbb{E}[\|\boldsymbol{x}_i\|^r]^{1/r} \leq \sqrt{C_x d r}$ *and* $\mathbb{E}[|y_i|^r]^{1/r} \leq \sqrt{C_y r^s}$ *for all* $r \geq 1$, $i \in [N]$, *and some absolute constants* $s \geq 1$ *and* $C_x, C_y > 0$.

Learning the qSTR model requires two steps: (i) extracting the relevant tokens at each position, (ii) learning the link function g. We are interested in settings where the difficulty of learning is dominated by the first step, hence we assume g can be approximated by a two-layer feedforward network.

Assumption 2 *There exist* $m_g \in \mathbb{N}$, $\boldsymbol{a}_g, \boldsymbol{b}_g \in \mathbb{R}^{m_g}$ *and* $\boldsymbol{W}_g \in \mathbb{R}^{m_g \times qd}$, *such that* $\|\boldsymbol{a}_g\|_2 \leq r_a/\sqrt{m_g}$, *and* $\|(\boldsymbol{W}_g, \boldsymbol{b}_g)\|_{\mathrm{F}} \leq \sqrt{m_g} r_w$ *for some constants* $r_a, r_w > 0$, *and*

$$\sup_{\left\{\|\boldsymbol{x}_i\|_2 \leq \sqrt{Cd\log(nN)}, \forall i \in [q]\right\}} \left| g(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q) - \boldsymbol{a}_g^\top \sigma(\boldsymbol{W}_g(\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_q^\top)^\top + \boldsymbol{b}_g) \right|^2 \leq \varepsilon_{2\mathrm{NN}},$$

where $C = 3C_x e$ *and* $\varepsilon_{2\mathrm{NN}}$ *is some absolute constant.*

Ideally, $\varepsilon_{2\mathrm{NN}}$ above is a small constant denoting the approximation error. This assumption can be verified using various universal approximation results for ReLU networks. For example, when g is an additive model of P Lipschitz functions, where each function depends only on a k-dimensional projection of the input, the above holds for every $\varepsilon_{2\mathrm{NN}} > 0$ and $m_g = \tilde{\mathcal{O}}\big((P/\sqrt{\varepsilon_{2\mathrm{NN}}})^k\big)$, $r_a = \tilde{\mathcal{O}}\big((P/\sqrt{\varepsilon_{2\mathrm{NN}}})^{(k+1)/2}\big)$, and $r_w = 1$ (we can always have $r_w = 1$ by homogeneity) [7].

Before introducing the notions of ERM that we employ, we first state several sequential risk formulations to evaluate a predictor $\hat{\boldsymbol{y}}_{\mathtt{arc}}(\cdot; \boldsymbol{\Theta}) \in \mathcal{F}_{\mathtt{arc}}$ on i.i.d. training samples $\{\boldsymbol{p}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1}^n$, where `arc` denotes a general architecture. We define the *population risk*, *averaged empirical risk*, and *point-wise empirical risk* respectively as

$$R^{\mathtt{arc}}(\boldsymbol{\Theta}) \coloneqq \frac{1}{N}\,\mathbb{E}\left[\sum_{j=1}^N (\hat{y}_{\mathtt{arc}}(\boldsymbol{p}; \boldsymbol{\Theta})_j - y_j)^2\right] = \frac{1}{N}\,\mathbb{E}\Big[\|\hat{\boldsymbol{y}}_{\mathtt{arc}}(\boldsymbol{p}; \boldsymbol{\Theta}) - \boldsymbol{y}\|_2^2\Big], \tag{1}$$

$$\hat{R}^{\mathtt{arc}}_{n,N}(\boldsymbol{\Theta}) \coloneqq \tfrac{1}{nN} \textstyle\sum_{i=1}^n \sum_{j=1}^N \big(\hat{y}_{\mathtt{arc}}(\boldsymbol{p}^{(i)}; \boldsymbol{\Theta})_j - y_j^{(i)}\big)^2, \tag{2}$$

$$\hat{R}^{\mathtt{arc}}_{n}(\boldsymbol{\Theta}) \coloneqq \frac{1}{n} \sum_{i=1}^n \big(\hat{y}_{\mathtt{arc}}(\boldsymbol{p}^{(i)}; \boldsymbol{\Theta})_{j^{(i)}} - y_{j^{(i)}}^{(i)}\big)^2, \tag{3}$$

where $\{j^{(i)}\}_{i=1}^n$ are i.i.d. position indices drawn from $\mathrm{Unif}([N])$.The goal is to minimize the population risk $R^{\mathtt{arc}}(\boldsymbol{\Theta})$ by minimizing some empirical risk, potentially with weight regularization. We use three formalizations of learning algorithms to prove our results.

1. *Constrained ERM* minimizes an empirical risk $\hat{R}_n^{\mathtt{arc}}$ subject to the model parameters belonging on some (e.g., norm-constrained) set $\varTheta$. Concretely, let

   $$\hat{\boldsymbol{\Theta}} \in \arg\min_{\boldsymbol{\Theta} \in \varTheta} \hat{R}_n^{\mathtt{arc}}(\boldsymbol{\Theta}).$$

   Theorem 3 considers constrained ERM algorithms for bounded-weight transformers with point-wise risk $\hat{R}_n^{\mathtt{TR}}(\boldsymbol{\Theta})$, and Theorem 6 uses $\hat{R}_n^{\mathtt{RNN}}(\boldsymbol{\Theta})$ for RNNs. Note that upper bounds for training with point-wise empirical risk $\hat{R}_n^{\mathtt{arc}}$ readily transfer to training with averaged empirical risk $\hat{R}_{n,N}^{\mathtt{arc}}$.

2. *Min-norm ε-ERM* minimizes the norm of the parameters, subject to sufficiently small loss:

$$\hat{\boldsymbol{\Theta}}_\varepsilon \in \underset{\{\boldsymbol{\Theta}:\hat{R}_n^{\mathrm{arc}}(\boldsymbol{\Theta})-\min \hat{R}_n^{\mathrm{arc}}\leq\varepsilon\}}{\arg\min} \|\mathrm{vec}(\boldsymbol{\Theta})\|_2. \tag{4}$$

   Theorem 7 uses min-norm ε-ERM to place a sample complexity lower bound $\hat{R}_n^{\mathrm{RNN}}(\boldsymbol{\Theta})$.

3. Beyond ERM, Theorem 8 also considers *stationary points* of the averaged or point-wise loss, with ℓ_2 regularization. This learning algorithm is presented in greater detail in Definition 41.

If Θ is defined by a norm constraint, then min-norm ε-ERM with a proper ε can be seen as an instance of constrained ERM. All three formulations are motivated by practical optimization algorithms that either minimize an explicitly regularized loss, or have an implicit bias towards min-norm solutions.

Appendix B. Details of Section 3

Here we present the omitted details and proofs of Section 3. We begin by presenting the architectural details before proving sample complexity upper bounds for Transformers.

B.1. Transformer Architectural Definition

We formally introduce the single-layer H-headed Transformer that appears in all Section 3 proofs.

Positional encoding. To break the permutation equivaraince of Transformers, we append positional information to the input tokens. Given a prompt $\boldsymbol{p}$, we consider an encoding given by

$$\boldsymbol{Z}(\boldsymbol{p}) = \begin{pmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_N \\ \mathrm{enc}(1,\boldsymbol{t}_1) & \dots & \mathrm{enc}(N,\boldsymbol{t}_N) \end{pmatrix} \in \mathbb{R}^{D_e\times N},$$

where $\mathrm{enc}:[N]\times[N]^q\to\mathbb{R}^{d_{\mathrm{enc}}}$ provides the encoding of the position and of $\boldsymbol{t}_i$, and $D_e \coloneqq d+d_{\mathrm{enc}}$. We use $\boldsymbol{z}_i$ to refer to the ith column above. We remark that allowing enc to take $\boldsymbol{t}_i$ as input allows specific encodings of the indices $\boldsymbol{t}_i$ that take advantage of the qSTR structure; examples of this have been considered in prior works [28]. In practice, we expect such useful encodings to be learned automatically by previous layers in the Transformer. We remark that for a fair comparison, in our lower bounds for other architectures we allow *arbitrary processing* of $\boldsymbol{t}_i$ in their encoding procedure. To specify enc, we use a set of vectors $\{\boldsymbol{\omega}_i\}_{i=1}^N$ in $\mathbb{R}^{d_e}$ that satisfy the following property.

Assumption 3 *We have $|\langle\boldsymbol{\omega}_i,\boldsymbol{\omega}_j\rangle|\leq\frac{1}{2}$ for all $i\neq j$, and $\|\boldsymbol{\omega}_i\|^2=1$ for all i, with $d_e=\Theta(\log N)$.*

Such a set of vectors can be obtained e.g., by sampling random Rademacher vectors from the unit cube $\{\pm1/\sqrt{d_e}\}^{d_e}$ which will satisfy the assumption with high probability. We define

$$\mathrm{enc}(i,\boldsymbol{t}_i) = \sqrt{d/q}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_{t_{i1}},\dots,\boldsymbol{\omega}_{t_{iq}})^\top \in \mathbb{R}^{(q+1)d_e},$$

hence $d_{\mathrm{enc}}=(q+1)d_e$ and $D_e=d+(q+1)d_e$. The $\sqrt{d/q}$ prefactor ensures that $\boldsymbol{x}_i$ and $\mathrm{enc}(i,\boldsymbol{t}_i)$ will roughly have the same ℓ_2 norm, resulting in a balanced input to the attention layer.

Multi-head attention. Given a sequence $\{\boldsymbol{z}_i\}_{i=1}^N$ where $\boldsymbol{z}_i \in \mathbb{R}^{D_e}$ with D_e as the embedding dimension, a single head of attention outputs another sequence of length N in $\mathbb{R}^{D_e}$, given by

$$f_{\texttt{Attn}}(\boldsymbol{p}; \boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V) = \left[\sum_{j=1}^N \boldsymbol{W}_V \boldsymbol{z}_j \frac{e^{\langle \boldsymbol{W}_Q \boldsymbol{z}_i, \boldsymbol{W}_K \boldsymbol{z}_j \rangle}}{\sum_{l=1}^N e^{\langle \boldsymbol{W}_Q \boldsymbol{z}_i, \boldsymbol{W}_K \boldsymbol{z}_l \rangle}}\right]_{i \in [N]}.$$

Where $\boldsymbol{W}_K, \boldsymbol{W}_Q, \boldsymbol{W}_V$ are the key, query, and value projection matrices respectively. We can simplify the presentation by replacing $\boldsymbol{W}_Q^\top \boldsymbol{W}_K$ with a single parameterizing matrix for query-key projections denoted by $\boldsymbol{W}_{\text{QK}} \in \mathbb{R}^{D_e \times D_e}$, and absorbing $\boldsymbol{W}_V$ into the weights of the feedforward layer. This provides us with a simplified parameterization of attention, which we denote by $f_{\texttt{Attn}}(\boldsymbol{p}; \boldsymbol{W}_{\text{QK}})$. This simplification is standard in theoretical works (see e.g. [2, 19, 28, 29]). Our main separation results still apply when maintaining separate trainable projections.

We can concatenate the output of H attention heads with separate key-query projection matrices to obtain a multi-head attention layer with H heads. We denote the output of head $h \in [H]$ with $f_{\texttt{Attn}}(\boldsymbol{p}; \boldsymbol{W}_{\text{QK}}^{(h)})$. The output of the multi-head attention at position i is then given by

$$f_{\texttt{Attn}}^{(H)}(\boldsymbol{p}; \boldsymbol{W}_{\text{QK}}^{(1)}, \ldots, \boldsymbol{W}_{\text{QK}}^{(H)})_i = (f_{\texttt{Attn}}(\boldsymbol{p}; \boldsymbol{W}_{\text{QK}}^{(1)})_i, \ldots, f_{\texttt{Attn}}(\boldsymbol{p}; \boldsymbol{W}_{\text{QK}}^{(H)})_i)^\top \in \mathbb{R}^{HD_e}.$$

We will denote by $\boldsymbol{\Theta}_{\text{QK}} = (\boldsymbol{W}_{\text{QK}}^{(1)}, \ldots, \boldsymbol{W}_{\text{QK}}^{(H)})$ the parameters of the multi-head attention.

Finally, a two-layer neural network acts on the output of the attention to generate labels. Given input $\boldsymbol{h} \in \mathbb{R}^{HD_e}$, the output of the network is given by

$$f_{\texttt{2NN}}(\boldsymbol{h}; \boldsymbol{a}_{\texttt{2NN}}, \boldsymbol{W}_{\texttt{2NN}}, \boldsymbol{b}_{\texttt{2NN}}) = \boldsymbol{a}_{\texttt{2NN}}^\top \sigma(\boldsymbol{W}_{\texttt{2NN}} \boldsymbol{h} + \boldsymbol{b}_{\texttt{2NN}}),$$

where $\boldsymbol{W}_{\texttt{2NN}} \in \mathbb{R}^{m \times HD_e}$ are the first layer weights, $\boldsymbol{b}_{\texttt{2NN}}, \boldsymbol{a}_{\texttt{2NN}} \in \mathbb{R}^m$ are the second layer weights and biases, and m is the width. We also use the summarized notation $\boldsymbol{\Theta}_{\texttt{2NN}} = (\boldsymbol{a}_{\texttt{2NN}}, \boldsymbol{W}_{\texttt{2NN}}, \boldsymbol{b}_{\texttt{2NN}})$ to refer to the feedforward layer weights. The prediction of the transformer at position i is given by

$$\hat{y}_{\texttt{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\texttt{TR}})_i = f_{\texttt{2NN}}(f_{\texttt{Attn}}^{(H)}(\boldsymbol{p}; \boldsymbol{\Theta}_{\text{QK}})_i; \boldsymbol{\Theta}_{\texttt{2NN}}),$$

where $\boldsymbol{\Theta}_{\texttt{TR}} = (\boldsymbol{\Theta}_{\text{QK}}, \boldsymbol{\Theta}_{\texttt{2NN}})$ denotes the overall trainable parameters of the Transformer. We use the notation $\hat{\boldsymbol{y}}_{\texttt{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\texttt{TR}}) = (\hat{y}_{\texttt{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\texttt{TR}})_1, \ldots, \hat{y}_{\texttt{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\texttt{TR}})_N)^\top \in \mathbb{R}^N$ to denote the vectorized output.

B.2. Proof of Theorem 3

To prove Theorem 3, we will prove the more general theorem below.

Theorem 9 *Let* $\hat{\boldsymbol{\Theta}} \coloneqq \arg\min_{\boldsymbol{\Theta} \in \Theta_{\texttt{TR}}} \hat{R}_n^{\texttt{TR}}(\boldsymbol{\Theta})$*, where*

$$\Theta_{\texttt{TR}} \coloneqq \left\{ \|\boldsymbol{a}_{\texttt{2NN}}\|_2 \le r_a/\sqrt{m}, \|(\boldsymbol{W}_{\texttt{2NN}}, \boldsymbol{b}_{\texttt{2NN}})\|_{\text{F}} \le r_w\sqrt{m}, \left\|\boldsymbol{W}_{\text{QK}}^{(h)}\right\|_{2,1} \le \alpha \ \forall h \in [H] \right\}.$$

Suppose $H = q$, $m = m_g$, *and* $\alpha = \tilde{\Theta}(1)$ *(given in Lemma 10). Then, under Assumptions 1 to 3, with probability at least* $1 - n^{-c}$ *for some absolute constant* $c > 0$*, we have*

$$R^{\texttt{TR}}(\hat{\boldsymbol{\Theta}}) \le \mathcal{O}(\varepsilon_{\text{NN}}^2) + \tilde{\mathcal{O}}\left(C_1 \sqrt{\frac{(m_g q(d+q) + r_z^6 r_a^2 r_w^2 q^2 \wedge q(q^2+d^2))}{n}} \right), \tag{5}$$

where $C_1 = q r_a^2 r_w^2 r_z^2$.

We begin with a lemma establishing the capability of Transformers in approximating qSTR models.

Lemma 10 *Suppose Assumption 2 holds. Let* $r_x = \sqrt{3C_x ed\log(nN)}$. *Assume* $H = q$ *and* $m_g = m$. *Then, there exists* $\boldsymbol{\Theta}_{\texttt{TR}}$ *such that*

$$\sup_{\{\|\boldsymbol{x}_j\|_2 \le r_x, \forall j\in[N]\}} \left|g(\boldsymbol{x}_{t_{i1}},\dots,\boldsymbol{x}_{t_{iq}}) - \hat{y}_{\texttt{TR}}(\boldsymbol{p};\boldsymbol{\Theta}_{\texttt{TR}})_i\right| \le 2\sqrt{\varepsilon_{\texttt{2NN}}},$$

and

$$\|\boldsymbol{a}_{\texttt{2NN}}\|_2 \le \frac{r_a}{\sqrt{m}}, \quad \|(\boldsymbol{W}_{\texttt{2NN}}, \boldsymbol{b}_{\texttt{2NN}})\|_{\mathrm{F}} \le \sqrt{m} r_w, \quad \left\|\boldsymbol{W}_{\mathrm{QK}}^{(h)\top}\right\|_{2,1} \le \frac{2d_e q}{d}\log\left(\frac{2r_a r_w r_x N\sqrt{q}}{\varepsilon_{\texttt{2NN}}}\right),$$

for all $h \in [H]$.

Proof In our construction, the goal of attention head h at position i will be to output $z_{t_{ih}}$. Namely, we want to achieve

$$f_{\texttt{Attn}}(\boldsymbol{p};\boldsymbol{W}_{\mathrm{QK}}^{(h)})_i \approx \boldsymbol{z}_{t_{ih}}.$$

Note that to do so, for each key token $\boldsymbol{z}_j$, we only need to compute $\langle \boldsymbol{\omega}_{t_{ih}}, \boldsymbol{\omega}_j\rangle$. Therefore, most entries in $\boldsymbol{W}_{\mathrm{QK}}^{(h)}$ can be zero. We only require a block of $d_e \times d_e$, which corresponds to comparing $\boldsymbol{\omega}_j$ and $\boldsymbol{\omega}_{t_{ih}}$ when comparing query $\boldsymbol{z}_i$ and key $\boldsymbol{z}_j$. Thus, we let

$$\boldsymbol{W}_{\mathrm{QK}}^{(h)} = \begin{pmatrix} \boldsymbol{0}_{(d+hd_e)\times d} & \boldsymbol{0}_{(d+hd_e)\times d_e} & \boldsymbol{0}_{(d+hd_e)\times qd_e} \\ \boldsymbol{0}_{d_e\times d} & \alpha\mathbf{I}_{d_e} & \boldsymbol{0}_{d_e\times qd_e} \\ \boldsymbol{0}_{(q-h)d_e\times d} & \boldsymbol{0}_{(q-h)d_e\times d_e} & \boldsymbol{0}_{(q-h)d_e\times qd_e} \end{pmatrix} \tag{6}$$

Then, we have $\left\langle \boldsymbol{z}_i, \boldsymbol{W}_{\mathrm{QK}}^{(h)}\boldsymbol{z}_j\right\rangle = \alpha\langle \boldsymbol{\omega}_{t_{ih}}, \boldsymbol{\omega}_j\rangle d/q$. We can then verify that

$$\left\|\boldsymbol{A}f_{\texttt{Attn}}(\boldsymbol{p};\boldsymbol{W}_{\mathrm{QK}}^{(h)})_i - \boldsymbol{A}\boldsymbol{z}_{t_{ih}}\right\|_2 \le \sum_{j\neq t_{ih}} e^{-\alpha d/(2q)}(\|\boldsymbol{A}\boldsymbol{z}_j\| + \|\boldsymbol{A}\boldsymbol{z}_{t_{ih}}\|_2)$$

for every matrix $\boldsymbol{A}$. We will specifically choose $\boldsymbol{A}$ to be the projection onto the first d coordinates in the following. Hence, α will control the error in the softmax attention approximating a "hard-max" attention that would exactly choose $\boldsymbol{z}_{t_{ih}}$.

To construct the weights of the feedforward layer $\boldsymbol{a}_{\texttt{2NN}}, \boldsymbol{W}_{\texttt{2NN}}, \boldsymbol{b}_{\texttt{2NN}}$, we let $\boldsymbol{a}_{\texttt{2NN}} = \boldsymbol{a}_g$ and $\boldsymbol{b}_{\texttt{2NN}} = \boldsymbol{b}_g$ from Assumption 2, and define $\boldsymbol{W}_{\texttt{2NN}}$ by extending $\boldsymbol{W}_g$ with zero entries such that

$$\boldsymbol{W}_{\texttt{2NN}}\begin{pmatrix}\boldsymbol{z}_{t_{i1}}\\ \dots \\ \boldsymbol{z}_{t_{iq}}\end{pmatrix} = \boldsymbol{W}_g\begin{pmatrix}\boldsymbol{x}_{t_{i1}}\\ \dots \\ \boldsymbol{x}_{t_{iq}}\end{pmatrix}.$$

Then $\|\boldsymbol{W}_{\texttt{2NN}}\|_{\mathrm{F}} = \|\boldsymbol{W}_g\|_{\mathrm{F}}$. Notice that $\cdot \mapsto \boldsymbol{a}^\top\sigma(\boldsymbol{W}(\cdot)+\boldsymbol{b})$ is $r_a r_w$ Lipschitz. As a result, for any $\boldsymbol{x}$ with $\|\boldsymbol{x}\| \le r_x$ we have

$$\left|g(\boldsymbol{x}_{t_{i1}},\dots,\boldsymbol{x}_{t_{iq}}) - \hat{y}_{\texttt{TR}}(\boldsymbol{p};\boldsymbol{\Theta}_{\texttt{TR}})_i\right| \le \sqrt{\varepsilon_{\texttt{2NN}}} + \varepsilon_{\texttt{Attn}},$$

where we recall

$$\left|g(\boldsymbol{x}_{t_{i1}},\dots,\boldsymbol{x}_{t_{iq}}) - f_{\texttt{2NN}}((\boldsymbol{z}_{t_{i1}},\dots,\boldsymbol{z}_{t_{iq}});\boldsymbol{a}_{\texttt{2NN}},\boldsymbol{W}_{\texttt{2NN}},\boldsymbol{b}_{\texttt{2NN}})\right| \le \sqrt{\varepsilon_{\texttt{2NN}}},$$

and

$$\begin{aligned}\varepsilon_{\texttt{Attn}} &= \left| f_{\texttt{2NN}}((\boldsymbol{z}_{t_{i1}}, \ldots, \boldsymbol{z}_{t_{iq}}); \boldsymbol{\Theta}_{\texttt{2NN}}) - f_{\texttt{2NN}}(f^{(q)}_{\texttt{Attn}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\mathrm{QK}}); \boldsymbol{\Theta}_{\texttt{2NN}}) \right| \\ &\leq r_a r_w \sqrt{\sum_{h=1}^{q} \left\| \boldsymbol{A} f_{\texttt{Attn}}(\boldsymbol{p}; \boldsymbol{W}^{(h)}_{\mathrm{QK}})_i - \boldsymbol{A}\boldsymbol{z}_{t_{ih}} \right\|_2^2} \\ &\leq 2 r_a r_w r_x N \sqrt{q} e^{-\alpha d/(2q)},\end{aligned}$$

where we recall $\boldsymbol{A}\boldsymbol{z}_j = \boldsymbol{x}_j$. Thus, with

$$\alpha = 2q \log(2 r_a r_w r_x N \sqrt{q} / \sqrt{\varepsilon_{\texttt{2NN}}})/d$$

we can guarantee the distance is at most $2\sqrt{\varepsilon_{\texttt{2NN}}}$. ■

Before proceeding to obtain statistical guarantees, we will show that we can consider the encodings $\boldsymbol{z}_j^{(i)}$ to be bounded with high probability. This will be a useful event to consider throughout the proofs of various sections.

Lemma 11 *Suppose $\{\boldsymbol{p}^{(i)}\}_{i=1}^n$ are n input prompts (not necessarily independent) drawn from the input distribution, with tokens denoted by $\{(\boldsymbol{x}_j^{(i)})_{j=1}^N\}_{i=1}^n$. Under Assumption 1, for any $r_x > 0$ we have*

$$\mathbb{P}\left(\max_{i \in [n], j \in [N]} \left\| \boldsymbol{x}_j^{(i)} \right\|_2 \geq r_x \right) \leq n N e^{-r_x^2/(2C_x e d)}.$$

In particular, for $r_x = \sqrt{3 C_x e d \log(nN)}$ we have

$$\mathbb{P}\left(\max_{i \in [n], j \in [N]} \left\| \boldsymbol{x}_j^{(i)} \right\|_2 \geq r_x \right) \leq \sqrt{\frac{1}{nN}}.$$

Proof Via Markov's inequality, for any $p > 0$ and $r_x > 0$, we have

$$\mathbb{P}\left(\max_{i,j} \left\| \boldsymbol{x}_j^{(i)} \right\|_2 \geq r_x \right) \leq \frac{\mathbb{E}\left[\max_{i,j} \left\| \boldsymbol{x}_j^{(i)} \right\|_2^p \right]}{r_x^p} \leq \frac{\mathbb{E}\left[\sum_{i,j} \left\| \boldsymbol{x}_j^{(i)} \right\|_2^p \right]}{r_x^p} \leq \frac{Nn(C_x p d)^{p/2}}{r_x^p}.$$

Let $p = r_x^2/(C_x e d)$. Then,

$$\mathbb{P}\left(\max_{i,j} \left\| \boldsymbol{x}_j^{(i)} \right\|_2 \geq r_x \right) \leq n N e^{-r_x^2/(2C_x e d)},$$

which proves the first statement, and the second statement follows by plugging in the specific value of r_x. ■

We are now ready to move to the generalization analysis of Transformers. First, we have to formally define the prediction function class of Transformers with a notation suitable for this section. We begin by defining the function class of attention. We have

$$\mathcal{F}_{\texttt{Attn}} = \{\boldsymbol{p}, j \mapsto f^{(H)}_{\texttt{Attn}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\mathrm{QK}})_j : \boldsymbol{\Theta}_{\mathrm{QK}} \in \varTheta_{\mathrm{QK}}\},$$

where we will later specify Θ_{QK}. Additionally, we define $\mathcal{F}_{\texttt{2NN}}$ by

$$\mathcal{F}_{\texttt{2NN}} = \{\boldsymbol{h} \mapsto f_{\texttt{2NN}}(\boldsymbol{h}; \boldsymbol{\Theta}_{\texttt{2NN}}) \,:\, \boldsymbol{\Theta}_{\texttt{2NN}} \in \Theta_{\texttt{2NN}}\},$$

where $\boldsymbol{\Theta}_{\texttt{2NN}} = (\boldsymbol{a}_{\texttt{2NN}}, \boldsymbol{W}_{\texttt{2NN}}, \boldsymbol{b}_{\texttt{2NN}})$, and we will later specify $\Theta_{\texttt{2NN}}$. Then the class $\mathcal{F}_{\texttt{TR}}$ can be defined as

$$\mathcal{F}_{\texttt{TR}} = \{\boldsymbol{p}, j \mapsto f_{\texttt{2NN}}(f_{\texttt{Attn}}(\boldsymbol{p})_j) \,:\, f_{\texttt{Attn}} \in \mathcal{F}_{\texttt{Attn}}, f_{\texttt{2NN}} \in \mathcal{F}_{\texttt{2NN}}\}.$$

Recall we use the S_n to denote the training set. To avoid extra indices, we will use the notation $\boldsymbol{p}, j \in S_n$ to go over $\{\boldsymbol{p}^{(i)}, j^{(i)}\}_{i=1}^n$. We can then define the following distances on the introduced function classes

$$\begin{aligned}
d_\infty^{\texttt{TR}}(f, f') &\coloneqq \sup_{\boldsymbol{p},j} \big|f(\boldsymbol{p})_j - f'(\boldsymbol{p})_j\big|, \quad \forall f, f' \in \mathcal{F}_{\texttt{TR}} \\
d_\infty^{\texttt{Attn}}(f, f') &\coloneqq \sup_{\boldsymbol{p},j} \big\|f(\boldsymbol{p})_j - f'(\boldsymbol{p})_j\big\|_2, \quad \forall f, f' \in \mathcal{F}_{\texttt{Attn}} \\
d_\infty^{\texttt{2NN}}(f, f') &\coloneqq \sup_{\|\cdot\|_2 \le \sqrt{H} r_z} \big|f(\cdot) - f'(\cdot)\big|, \quad \forall f, f' \in \mathcal{F}_{\texttt{2NN}}.
\end{aligned}$$

We choose the radius $\sqrt{H} r_z$ for defining $d_\infty^{\texttt{2NN}}$ since on the event of Lemma 11, this will be the norm bound on the output of the attention layer at every position.

Recall that for a distance d_∞ and a set $\mathcal{F}$, an ϵ-covering $\hat{\mathcal{F}}$ is a set such that for every $f \in \mathcal{F}$, there exists $\hat{f} \in \hat{\mathcal{F}}$ such that $d_\infty(f, \hat{f}) \le \epsilon$. The ϵ-covering number of $\mathcal{F}$, denoted by $\mathcal{C}(\mathcal{F}, d_\infty, \epsilon)$, is the number of elements of the smallest such $\hat{\mathcal{F}}$. The following lemma relates the covering number of $\mathcal{F}_{\texttt{TR}}$ to those of $\mathcal{F}_{\texttt{Attn}}$ and $\mathcal{F}_{\texttt{2NN}}$.

Lemma 12 *Suppose $f_{\texttt{2NN}}$ is L_f Lipschitz for every $f_{\texttt{2NN}} \in \mathcal{F}_{\texttt{2NN}}$. Then, for any $\epsilon_{\texttt{2NN}}, \epsilon_{\texttt{Attn}} > 0$, on the event of Lemma 11 we have*

$$\log \mathcal{C}(\mathcal{F}_{\texttt{TR}}, d_\infty^{\texttt{TR}}, \epsilon_{\texttt{2NN}} + L_f \epsilon_{\texttt{Attn}}) \le \log \mathcal{C}\big(\mathcal{F}_{\texttt{2NN}}, d_\infty^{\texttt{2NN}}, \epsilon_{\texttt{2NN}}\big) + \log \mathcal{C}\big(\mathcal{F}_{\texttt{Attn}}, d_\infty^{\texttt{Attn}}, \epsilon_{\texttt{Attn}}\big).$$

Proof The proof simply follows from the triangle inequality, namely

$$\begin{aligned}
\sup_{\boldsymbol{p},j} \Big|f_{\texttt{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\texttt{TR}})_j - f_{\texttt{TR}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}}_{\texttt{TR}})_j\Big| \le & \sup_{\|\boldsymbol{h}\|_2 \le \sqrt{H} r_z} \Big\|f_{\texttt{2NN}}(\boldsymbol{h}; \boldsymbol{\Theta}_{\mathrm{NN}}) - f_{\texttt{2NN}}(\boldsymbol{h}; \hat{\boldsymbol{\Theta}}_{\mathrm{NN}})\Big\|_2 \\
& + L_f \sup_{\boldsymbol{p},j} \Big\|f^{(H)}_{\texttt{Attn}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\mathrm{QK}})_j - f^{(H)}_{\texttt{Attn}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}}_{\mathrm{QK}})_j\Big\|_2.
\end{aligned}$$

■

We have the following estimate for the covering number of $\mathcal{F}_{\texttt{2NN}}$.

Lemma 13 *Suppose $\|\mathrm{vec}(\boldsymbol{\Theta}_{\mathrm{RNN}})\|_2 \le R$ and $\left\|\boldsymbol{z}_j^{(i)}\right\|_2 \le R$ for all $i \in [n]$ and $j \in [N]$. Then,*

$$\log \mathcal{C}\big(\mathcal{F}_{\texttt{2NN}}, d_\infty^{\texttt{2NN}}, \epsilon\big) \lesssim m_g H D_e \log(1 + \mathrm{poly}(R)/\epsilon).$$

This is a special case of Lemma 29, proved in Appendix C.

For the next step, define the distance

$$d_\infty^{\mathrm{QK}}(\boldsymbol{\Theta}_{\mathrm{QK}}, \boldsymbol{\Theta}'_{\mathrm{QK}}) \coloneqq \sup_{\boldsymbol{p},j} \Big\|\boldsymbol{\Theta}_{\mathrm{QK}}^\top \boldsymbol{z}_j - \boldsymbol{\Theta}'^\top_{\mathrm{QK}} \boldsymbol{z}_j\Big\|_2$$

on Θ_{QK}, where we recall $\boldsymbol{\Theta}_{\mathrm{QK}} = (\boldsymbol{W}_{\mathrm{QK}}^{(1)}, \ldots, \boldsymbol{W}_{\mathrm{QK}}^{(H)}) \in \mathbb{R}^{D_e \times HD_e}$. The following lemma relates the covering number of the multi-head attention layer to the matrix covering number of the class of attention parameters.

Lemma 14 *Suppose* $\left\|\boldsymbol{z}_j^{(i)}\right\|_2 \leq r_z$ *for all* $i \in [n]$ *and* $j \in [N]$. *Then,*

$$\log \mathcal{C}(\mathcal{F}_{\texttt{Attn}}, d_\infty^{\texttt{Attn}}, \epsilon) \leq \log \mathcal{C}\left(\Theta_{\mathrm{QK}}, d_\infty^{\mathrm{QK}}, \frac{\epsilon}{2r_z^2}\right).$$

Proof We recall that $\boldsymbol{Z} \in \mathbb{R}^{N \times D_e}$ denotes the encoded prompt, and $\mathrm{softmax}$ is applied row-wise. For conciseness, Let $\Delta := \sup_{\boldsymbol{p},j} \left\| f_{\texttt{Attn}}^{(H)}(\boldsymbol{p}; \boldsymbol{\Theta}_{\mathrm{QK}})_j - f_{\texttt{Attn}}^{(H)}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}}_{\mathrm{QK}})_j \right\|_2^2$. Then we have

$$\begin{aligned}
\Delta &= \sup_{\boldsymbol{p}, j \in S_n} \sum_{h \in [H]} \left\| f_{\texttt{Attn}}(\boldsymbol{p}; \boldsymbol{W}_{\mathrm{QK}}^{(h)})_j - f_{\texttt{Attn}}(\boldsymbol{p}; \hat{\boldsymbol{W}}_{\mathrm{QK}}^{(h)})_j \right\|_2^2 \\
&= \sup_{\boldsymbol{p}, j \in S_n} \sum_{h \in [H]} \left\| \mathrm{softmax}\left(\boldsymbol{z}_j^\top \boldsymbol{W}_{\mathrm{QK}}^{(h)} \boldsymbol{Z}^\top\right) \boldsymbol{Z} - \mathrm{softmax}\left(\boldsymbol{z}_j^\top \hat{\boldsymbol{W}}_{\mathrm{QK}}^{(h)} \boldsymbol{Z}^\top\right) \boldsymbol{Z} \right\|_2^2 \\
&\leq \sup_{\boldsymbol{p}, j \in S_n} \sum_{h \in [H]} \left\| \boldsymbol{Z}^\top \right\|_{2,\infty}^2 \left\| \mathrm{softmax}(\boldsymbol{z}_j^\top \boldsymbol{W}_{\mathrm{QK}}^{(h)} \boldsymbol{Z}^\top)^\top - \mathrm{softmax}(\boldsymbol{z}_j^\top \hat{\boldsymbol{W}}_{\mathrm{QK}}^{(h)} \boldsymbol{Z}^\top)^\top \right\|_1^2,
\end{aligned}$$

where we used Lemma 42 for the last inequality. Moreover, by [14, Corollary A.7],

$$\begin{aligned}
\left\| \mathrm{softmax}\left(\boldsymbol{z}_j^\top \boldsymbol{W}_{\mathrm{QK}}^{(h)} \boldsymbol{Z}^\top\right)^\top - \mathrm{softmax}\left(\boldsymbol{z}_j^\top \hat{\boldsymbol{W}}_{\mathrm{QK}}^{(h)} \boldsymbol{Z}^\top\right) \right\|_1 &\leq 2 \left\| \boldsymbol{Z} \boldsymbol{W}^{(h)\top}_{\mathrm{QK}} \boldsymbol{z}_j - \boldsymbol{Z} \hat{\boldsymbol{W}}^{(h)\top}_{\mathrm{QK}} \boldsymbol{z}_j \right\|_\infty \\
&\leq 2 \left\| \boldsymbol{Z}^\top \right\|_{2,\infty} \left\| \boldsymbol{W}^{(h)\top}_{\mathrm{QK}} \boldsymbol{z}_j - \hat{\boldsymbol{W}}^{(h)\top}_{\mathrm{QK}} \boldsymbol{z}_j \right\|_2.
\end{aligned}$$

Consequently,

$$\begin{aligned}
\Delta &\leq 4r_z^4 \sup_{\boldsymbol{p}, j \in S_n} \sum_{h \in [H]} \left\| \boldsymbol{W}_{\mathrm{QK}}^{(h)\top} \boldsymbol{z}_j - \hat{\boldsymbol{W}}^{(h)\top}_{\mathrm{QK}} \boldsymbol{z}_j \right\|_2^2 \\
&= 4r_z^4 \sup_{\boldsymbol{p}, j \in S_n} \left\| \boldsymbol{\Theta}_{\mathrm{QK}}^\top \boldsymbol{z}_j - \hat{\boldsymbol{\Theta}}^\top_{\mathrm{QK}} \boldsymbol{z}_j \right\|_2^2,
\end{aligned}$$

which completes the proof. ■

Further, we have the following covering number estimate for Θ_{QK}.

Lemma 15 *Suppose* $\Theta_{\mathrm{QK}} = \{\|\boldsymbol{\Theta}_{\mathrm{QK}}\|_{2,1} \leq R_{2,1}, \|\boldsymbol{\Theta}_{\mathrm{QK}}\|_{\mathrm{F}} \leq R_F\}$ *and* $\left\|\boldsymbol{z}_j^{(i)}\right\|_2 \leq r_z$ *for all* $i \in [n]$ *and* $j \in [N]$. *Then,*

$$\log \mathcal{C}\big(\Theta_{\mathrm{QK}}, d_\infty^{\mathrm{QK}}, \epsilon\big) \lesssim \min\left(\frac{r_z^2 R_{2,1}^2 \log(2HD_e^2)}{\epsilon^2}, HD_e^2 \log\left(1 + \frac{2R_F r_z}{\epsilon}\right) \right).$$

Proof The first estimate comes from Maurey's sparsification lemma [9, Lemma 3.2], while the second estimate is based on the inequality

$$\left\|\boldsymbol{\Theta}_{\mathrm{QK}}^{\top}\boldsymbol{z}_j - \hat{\boldsymbol{\Theta}}_{\mathrm{QK}}^{\top}\boldsymbol{z}_j\right\|_2 \leq r_z\left\|\boldsymbol{\Theta}_{\mathrm{QK}} - \hat{\boldsymbol{\Theta}}_{\mathrm{QK}}\right\|_{\mathrm{F}},$$

and covering Θ_{QK} with the Frobenius norm, see e.g. Lemma 44. ■

Finally, we obtain the following covering number for $\mathcal{F}_{\mathtt{TR}}$.

Proposition 16 *Suppose* $\|\boldsymbol{a}_{2\mathtt{NN}}\|_2 \leq r_{m,a}$, $\|(\boldsymbol{W}_{2\mathtt{NN}}, \boldsymbol{b}_{2\mathtt{NN}})\|_{\mathrm{F}} \leq R_{m,w}$, *and* $\left\|\boldsymbol{W}_{\mathrm{QK}}^{(h)}\right\|_{2,1} \leq r_{\mathrm{QK}}$ *for all* $h \in [H]$. *Further assume* $\left\|\boldsymbol{z}_j^{(i)}\right\|_2 \leq r_z$ *for all* $i \in [n]$ *and* $j \in [N]$. *Let* $R := \max(r_{m,a}, R_{m,w}, r_z)$. *Then,*

$$\begin{aligned}\log \mathcal{C}(\mathcal{F}_{\mathtt{TR}}, d_{\mathcal{F}}, \epsilon) \lesssim & m_g H D_e \log(1 + R/\epsilon) \\ & + \min\left(\frac{r_z^6 r_{m,a}^2 R_{m,w}^2 H^2 r_{QK}^2 \log(HD_e^2)}{\epsilon^2}, HD_e^2 \log\left(1 + \frac{\sqrt{H} r_{\mathrm{QK}} r_z^3 r_{m,a} R_{m,w}}{\epsilon}\right)\right).\end{aligned}$$

Proof The proof follows from a number of observations. First, given the parameterization in the statement of the proposition, we have $L_f = r_{m,a}R_{m,w}$ in Lemma 12. Moreover, we have $R_F \leq \sqrt{H} r_{\mathrm{QK}}$ and $R_{2,1} \leq H r_{\mathrm{QK}}$ in Lemma 15. The rest follows from combining the statements of the previous lemmas. ■

Next, we will use the covering number bound to provide a bound for Rademacher complexity. Recall that for a class of loss functions $\mathcal{L}$, the empirical and population Rademacher complexities are defined as

$$\hat{\mathfrak{R}}_n(\mathcal{L}) := \mathbb{E}\left[\sup_{\ell \in \mathcal{L}} \frac{1}{n}\sum_{i=1}^{n} \xi_i \ell(\boldsymbol{p}^{(i)}, \boldsymbol{y}^{(i)}, j^{(i)})\right], \quad \mathfrak{R}_n(\mathcal{L}) := \mathbb{E}_{(\boldsymbol{p},\boldsymbol{y},j)}\left[\hat{\mathfrak{R}}_n(\mathcal{L})\right]$$

respectively, where (ξ_i) are i.i.d. Rademacher random variables. Let the class of loss functions be defined by

$$\mathcal{L}_\tau := \{(\boldsymbol{p}, \boldsymbol{y}, j) \mapsto (f_{\mathtt{TR}}(\boldsymbol{p})_j - y_j)^2 \wedge \tau \,:\, f_{\mathtt{TR}} \in \mathcal{F}_{\mathtt{TR}}\}, \tag{7}$$

for some constant $\tau > 0$ to be fixed later. We then have the following bound on Rademacher complexity.

Lemma 17 *Suppose* $\max_{i\in[n],j\in[N]}\left\|\boldsymbol{z}_j^{(i)}\right\|_2 \leq r_z$. *For the loss class* $\mathcal{L}_\tau$ *given by* (7), *we have*

$$\hat{\mathfrak{R}}_n(\mathcal{L}_\tau) \leq \tilde{\mathcal{O}}\left(\tau\sqrt{\frac{C_1 + (C_2 \wedge C_3)}{n}}\right),$$

where $C_1 = m_g H D_e$, $C_2 = r_z^6 r_{m,a}^2 R_{m,w}^2 H^2 r_{QK}^2$, *and* $C_3 = HD_e^2$.

Proof Let $\mathcal{C}(\mathcal{L}, d_\infty^{\mathcal{L}}, \epsilon)$ denote the ϵ-covering number of $\mathcal{L}$, where $\ell(\boldsymbol{p}, \boldsymbol{y}, j) = (f(\boldsymbol{p})_j - y_j)^2 \wedge \tau$ and $\ell'(\boldsymbol{p}, \boldsymbol{y}, j) = (f'(\boldsymbol{p})_j - y_j)^2 \wedge \tau$. Then, for any $\alpha \geq 0$, by a standard chaining argument,

$$\begin{aligned}
\hat{\mathfrak{R}}_n(\mathcal{L}_\tau) &\lesssim \alpha + \int_\alpha^\tau \sqrt{\frac{\log \mathcal{C}(\mathcal{L}, d_\infty^{\mathcal{L}}, \epsilon)}{n}} \mathrm{d}\epsilon. \\
&\lesssim \alpha + \int_\alpha^\tau \sqrt{\frac{\log \mathcal{C}(\mathcal{F}, d_\infty^{\mathrm{TR}}, \epsilon/(2\sqrt{\tau}))}{n}} \\
&\lesssim \alpha + \int_\alpha^\tau \sqrt{\frac{C_1 \log(R\sqrt{\tau}/\epsilon)}{n}} \mathrm{d}\epsilon + \left\{ \int_\alpha^\tau \sqrt{\frac{\tau C_2 \log(H D_e^2)}{n\epsilon^2}} \mathrm{d}\epsilon \right\} \wedge \left\{ \int_\alpha^\tau \sqrt{\frac{C_3 \log(1 + C_4\sqrt{\tau}/\epsilon)}{n}} \mathrm{d}\epsilon \right\} \\
&\lesssim \alpha + \sqrt{\frac{\tau^2 C_1 \log(R\sqrt{\tau}/\alpha)}{n}} + \left\{ \sqrt{\frac{\tau C_2 \log(H D_e^2)}{n}} \log\Big(\frac{\tau}{\alpha}\Big) \right\} \wedge \left\{ \sqrt{\frac{\tau^2 C_3 \log(1 + C_4\sqrt{\tau}/\alpha)}{n}} \right\},
\end{aligned}$$

where $(C_i)_{i=1}^3$ are given in the statement of the lemma and $C_4 = \sqrt{H} r_{\mathrm{QK}} r_z^3 r_{m,a} R_{m,w}$. Choosing $\alpha = 1/\sqrt{n}$ completes the proof. ■

Using standard symmetrization techniques, the above immediately yields a high probability upper bound for the expected truncated loss of any estimator in Θ_{TR}.

Corollary 18 *Let $\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta} \in \Theta_{\mathrm{TR}}} \hat{R}_n^{\mathrm{TR}}(\boldsymbol{\Theta})$, where Θ_{TR} is described in Proposition 16. Define $r_z = \sqrt{r_x^2 + d(1 + 1/q)}$ where r_x is defined in Lemma 11. Let C_1, C_2, and C_3 be defined as in Lemma 17. Then, with probability at least $1 - \delta - (nN)^{-1/2}$ over S_n, we have*

$$R_\tau^{\mathrm{TR}}(\hat{\boldsymbol{\Theta}}) - \hat{R}_n^{\mathrm{TR}}(\hat{\boldsymbol{\Theta}}) \leq \tilde{\mathcal{O}}\left(\tau \sqrt{\frac{(C_1 + C_2 \wedge C_3)}{n}} \right) + \mathcal{O}\left(\tau \sqrt{\frac{\log(1/\delta)}{n}} \right),$$

where $R_\tau^{\mathrm{RNN}}(\hat{\boldsymbol{\Theta}}) := \mathbb{E}_{\boldsymbol{p},j,y}\Big[(\hat{y}_{\mathrm{TR}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j - y_j)^2 \wedge \tau \Big]$

Proof The proof is a standard consequence of Rademacher-based generalization bounds, with the additional observation that

$$\frac{1}{n} \sum_{i=1}^n \left(\hat{y}_{\mathrm{TR}}(\boldsymbol{p}^{(i)}; \hat{\boldsymbol{\Theta}})_{j^{(i)}} - y_{j^{(i)}}^{(i)} \right)^2 \wedge \tau \leq \hat{R}_n^{\mathrm{TR}}(\hat{\boldsymbol{\Theta}}).$$

■

The last step in the proof of the generalization bound is to bound $R^{\mathrm{TR}}(\hat{\boldsymbol{\Theta}})$ with $R_\tau^{\mathrm{TR}}(\hat{\boldsymbol{\Theta}})$. This is achieved by the following lemma.

Lemma 19 *Define $\kappa^2 := H r_{m,a}^2 R_{m,w}^2 r_z^2$. Then, under Assumption 1, for $\tau \asymp \kappa^2 \log(\kappa^2 N\sqrt{n}) + \log(\kappa^2\sqrt{n})^s$, we have*

$$R^{\mathrm{TR}}(\hat{\boldsymbol{\Theta}}) - R_\tau^{\mathrm{TR}}(\hat{\boldsymbol{\Theta}}) \leq \sqrt{\frac{1}{n}}.$$

Proof For conciseness, define $\Delta_y := \left|\hat{y}_{\text{TR}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j - y_j\right|$. By the Cuachy-Schwartz inequality, we have

$$\begin{aligned} R^{\text{TR}}(\hat{\boldsymbol{\Theta}}) &= \mathbb{E}\big[\Delta_y^2 \mathbb{1}\big[\Delta_y \le \sqrt{\tau}\big]\big] + \mathbb{E}\big[\Delta_y^2 \mathbb{1}\big[\Delta_y > \sqrt{\tau}\big]\big] \\ &\le R_\tau^{\text{TR}}(\hat{\boldsymbol{\Theta}}) + \mathbb{E}\big[\Delta_y^4\big]^{1/2} \mathbb{P}\big(\Delta_y \ge \sqrt{\tau}\big)^{1/2}. \end{aligned}$$

Moreover,

$$\mathbb{E}\big[\Delta_y^4\big]^{1/2} \le 2\,\mathbb{E}\big[y_j^4\big]^{1/2} + 2\,\mathbb{E}\Big[\hat{y}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j^4\Big]^{1/2}.$$

By Assumption 1, we have $\mathbb{E}\Big[y_j^4\Big]^{1/2} \lesssim 1$. Additionally, note that

$$\begin{aligned} \left|\hat{y}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j\right| &\le \|\boldsymbol{a}_{2\text{NN}}\|_2 (\sqrt{H} \|\boldsymbol{W}_{2\text{NN}}\|_{\text{F}} \max_{l\in[N]} \|\boldsymbol{z}_l\|_2 + \|\boldsymbol{b}_{2\text{NN}}\|_2) \\ &\le \sqrt{H} r_{m,a} R_{m,w} (1 + \max_{l\in[N]} \|\boldsymbol{z}_l\|_2). \end{aligned}$$

To bound $\max_{l\in[N]} \|\boldsymbol{z}_l\|_2$, we use the subGaussianity of $\|\boldsymbol{x}_l\|_2$ characterized in Assumption 1. Specifically, for all $r \ge 1$

$$\begin{aligned} \mathbb{E}\left[\max_{l\in[N]} \|\boldsymbol{x}_l\|_2^4\right] \le \mathbb{E}\left[\max_{l\in[N]} \|\boldsymbol{x}_l\|_2^{4r}\right]^{1/r} &\le \mathbb{E}\left[\sum_{l=1}^N \|\boldsymbol{x}_l\|_2^{4r}\right]^{1/r} \\ &\le N^{1/r}\, \mathbb{E}\Big[\|\boldsymbol{x}_1\|_2^{4r}\Big]^{1/r} \\ &\lesssim N^{1/r} C_x^2 d^2 r^2 \\ &\lesssim (C_x d \log(N))^2, \end{aligned}$$

where the last inequality follows from choosing $r = \log N$. As a result,

$$\mathbb{E}\Big[\hat{y}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j^4\Big]^{1/2} \lesssim H r_{m,a}^2 R_{m,w}^2 r_z^2 \log(N)^2 =: \kappa^2 \log(N)^2.$$

We now turn to bounding the probability. We have

$$\begin{aligned} \mathbb{P}\big(\Delta_y \ge \sqrt{\tau}\big) &\le \mathbb{P}\left(|y_j| \ge \frac{\sqrt{\tau}}{2}\right) + \mathbb{P}\left(\left|\hat{y}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j\right| \ge \frac{\sqrt{\tau}}{2}\right) \\ &\le \exp\Big(-\Omega(\tau^{1/s})\Big) + N \exp\Big(-\Omega\Big(\frac{\tau}{H r_{m,a}^2 R_{m,w}^2 r_z^2}\Big)\Big), \end{aligned}$$

where the second inequality follows from sub-Weibull concentration bounds for y and Lemma 11. Choosing $\tau = \Theta(\kappa^2 \log(\kappa^2 N\sqrt{n}) + \log(\kappa^2\sqrt{n})^s)$ completes the proof. ■

Proof of Theorem 9. The theorem follows immediately from the approximation guarantee of Lemma 10, the generalization bound of Corollary 18, and the truncation control of Lemma 19. ■

B.3. Details on Limitations of Transformers with Few Heads

While Proposition 4 is only meaningful in the setting of $d = \Omega(q)$, the following proposition provides an exact lower bound $H \geq q$ on the number of heads for all d, at the expense of additional restrictions on the attention matrix.

Proposition 20 *Consider the qSTR data model. Suppose $d = 1$ and $y_i = \frac{1}{\sqrt{q}} \sum_{j=1}^{q} (x_{t_{ij}}^2 - \mathbb{E}[x_{t_{ij}}^2])$. Assume $x_i \sim \mathcal{N}(0, \sigma_i^2)$ independently, such that $\sigma_i = 1$ for $i < N/2$ and $\sigma_i = 0$ for $i \geq N/2$. Further, assume the attention weights between the data and positional encoding parts of the tokens are fixed at zero, i.e. $\boldsymbol{W}_{\mathrm{QK}}^{(h)} = \begin{pmatrix} \boldsymbol{W}_{\boldsymbol{x}}^{(h)} & \boldsymbol{0}_{d\times(q+1)d_e} \\ \boldsymbol{0}_{(q+1)d_e\times d} & \boldsymbol{W}_{\boldsymbol{\omega}}^{(h)} \end{pmatrix}$ where $\boldsymbol{W}_{\boldsymbol{x}}^{(h)} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{W}_{\boldsymbol{\omega}}^{(h)} \in \mathbb{R}^{(q+1)d_e\times(q+1)d_e}$ are the attention parameters, for $i \in [H]$. Then, there exists a distribution over $(\boldsymbol{t}_i)_{i\in[N]}$ such that for any choice of $\boldsymbol{\Theta}_{\mathrm{TR}}$, we have*

$$\frac{1}{N}\mathbb{E}\Big[\|\boldsymbol{y} - \hat{\boldsymbol{y}}_{\mathrm{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\mathrm{TR}})\|_2^2\Big] \geq 1 - \frac{H}{q}.$$

Note that in our approximation constructions for learning qSTR, we always fixed the attention weights between data and positional components to be zero, which is why we assume the same in Proposition 20.

Proof of Proposition 20. We will simply choose $\boldsymbol{t}_i = (1, \ldots, q)$ deterministically for $i \geq \frac{N}{2}$ and draw $\boldsymbol{t}_i$ from an arbitrary distribution for $i < N/2$. Note that we have

$$R^{\mathrm{TR}}(\boldsymbol{\Theta}_{\mathrm{TR}}) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\big[(y_i - \hat{y}_{\mathrm{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\mathrm{TR}})_i)^2\big] \geq \frac{1}{N}\sum_{i=N/2}^{N}\mathbb{E}\big[(y_i - \hat{y}_{\mathrm{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\mathrm{TR}})_i)^2\big].$$

Let $\phi : \mathbb{R}^{HD_e} \to \mathbb{R}$ denote the mapping by the feedforward layer. Fix some $i \geq N/2$. Note that

$$\begin{aligned}
\hat{y}_{\mathrm{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\mathrm{TR}})_i &= \phi(f_{\mathtt{Attn}}^{(H)}(\boldsymbol{p}; \boldsymbol{\Theta}_{\mathrm{QK}})_i) \\
&= \phi\Big(\sum_{j=1}^{N}\alpha_{ij}^{(1)}\boldsymbol{z}_j, \ldots, \sum_{j=1}^{N}\alpha_{ij}^{(H)}\boldsymbol{z}_j\Big) \\
&= \tilde{\phi}\Big(\sum_{j=1}^{q}\alpha_{ij}^{(1)}x_j, \ldots, \sum_{j=1}^{q}\alpha_{ij}^{(H)}x_j, (\boldsymbol{z}_l)_{l=q+1}^{N}\Big),
\end{aligned}$$

for some real-valued function $\tilde{\phi}$, where

$$\alpha_{ij}^{(h)} = \frac{e^{\left\langle \boldsymbol{z}_i, \boldsymbol{W}_{\mathrm{QK}}^{(h)}\boldsymbol{z}_j\right\rangle}}{\sum_{l=1}^{N} e^{\left\langle \boldsymbol{z}_i, \boldsymbol{W}_{\mathrm{QK}}^{(h)}\boldsymbol{z}_j\right\rangle}},$$

are the attention scores. Let $\boldsymbol{A}^{(i)} \in \mathbb{R}^{H\times q}$ be the matrix such that $A^{(i)}_{hj} = \alpha^{(h)}_{ij}$. Let $\boldsymbol{x}_{1:q} = (x_1, \ldots, x_q)^\top \in \mathbb{R}^q$. Then,

$$\begin{aligned} R^{\text{TR}}(\boldsymbol{\Theta}_{\text{TR}}) &\geq \frac{1}{N}\sum_{i=N/2}^{N} \mathbb{E}\Big[\Big(y_i - \tilde{\phi}\Big(\boldsymbol{A}^{(i)}\boldsymbol{x}_{1:q}, (\boldsymbol{z}_l)_{l=q+1}^N\Big)\Big)^2\Big] \\ &\geq \frac{1}{Nq}\sum_{i=N/2}^{N} \mathbb{E}\Big[\mathrm{Var}\Big(\|\boldsymbol{x}_{1:q}\|^2 \,|\, \boldsymbol{V}^{(i)}\boldsymbol{x}_{1:q}\Big)\Big] \end{aligned} \tag{8}$$

where $\boldsymbol{V}^{(i)} \in \mathbb{R}^{H\times q}$ is a matrix whose rows form an orthonormal basis of $\mathrm{span}(\boldsymbol{\alpha}_i^{(1)}, \ldots, \boldsymbol{\alpha}_i^{(H)})$ where $\boldsymbol{\alpha}_i^{(h)} = (\alpha_{i1}^{(h)}, \ldots, \alpha_{iq}^{(h)})^\top \in \mathbb{R}^q$ (note that $\boldsymbol{V}^{(i)}$ may have fewer than H rows, we consider the worst-case for the lower bound which is having H rows). The second inequality follows from the fact that $\boldsymbol{z}_l$ is independent of $\boldsymbol{x}_{1:q}$ for $l \geq q+1$, and the fact that best predictor of y_i (in L_2 error) given $\boldsymbol{A}^{(i)}\boldsymbol{x}_{1:q}$ is $\mathbb{E}\Big[y_i \,|\, \boldsymbol{V}^{(i)}\boldsymbol{x}_{1:q}\Big]$.

Next, thanks to the structural property of $\boldsymbol{W}_{\text{QK}}^{(h)}$ in the assumption of the proposition and the fact that $x_i = 0$ for $i \geq N/2$, $\alpha_{ij}^{(h)}$ does not depend on $(x_l)_{l\in[q]}$ for all $h \in [H]$, $i \geq N/2$, and $j \in [q]$. As a result, $\boldsymbol{V}^{(i)}$ is independent of $\boldsymbol{x}_{1:q}$. Therefore,

$$\boldsymbol{x}_{1:q} \,|\, \boldsymbol{V}^{(i)}\boldsymbol{x}_{1:q} \sim \mathcal{N}({\boldsymbol{V}^{(i)}}^\top \boldsymbol{V}^{(i)}\boldsymbol{x}_{1:q}, \mathbf{I}_q - {\boldsymbol{V}^{(i)}}^\top \boldsymbol{V}^{(i)}).$$

By Lemma 43, we have $\mathrm{Var}(\|\boldsymbol{x}_{1:q}\|^2 \,|\, \boldsymbol{V}^{(i)}\boldsymbol{x}_{1:q}) = 2(q-H)$, which combined with (8) completes the proof. ■

We now present the similarly structured proof of Proposition 4.

Proof of Proposition 4. The choice of distribution over $(\boldsymbol{t}_i)_{i\geq N/2}$ is similar to the one presented above, i.e. we let $\boldsymbol{t}_i = (1, \ldots, q)$ deterministically for $i \geq \frac{N}{2}$. However, for $i < \frac{N}{2}$, we draw $\boldsymbol{t}_i$ such that they are independent from $\boldsymbol{x}$. Once again, we use the fact that

$$R^{\text{TR}}(\boldsymbol{\Theta}_{\text{TR}}) \geq \frac{1}{N}\sum_{i=N/2}^{N} \mathbb{E}\big[(y_i - \hat{y}_{\text{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\text{TR}})_i)^2\big].$$

Recall $\boldsymbol{z}_i = (\boldsymbol{x}_i^\top, \mathrm{enc}(i, \boldsymbol{t}_i)^\top)$. Fix some $i \geq N/2$, and define

$$\tilde{\alpha}_{ij}^{(h)} = e^{\left\langle \mathrm{enc}(i,\boldsymbol{t}_i), \boldsymbol{W}_{\text{QK}}^{(h,e,x)}\boldsymbol{x}_j\right\rangle + \left\langle \mathrm{enc}(i,\boldsymbol{t}_i), \boldsymbol{W}_{\text{QK}}^{(h,e,e)}\,\mathrm{enc}(j,\boldsymbol{t}_j)\right\rangle},$$

where we use the notation

$$\boldsymbol{W}_{\text{QK}}^{(h)} = \begin{pmatrix} \boldsymbol{W}_{\text{QK}}^{(h,x,x)} & \boldsymbol{W}_{\text{QK}}^{(h,x,e)} \\ \boldsymbol{W}_{\text{QK}}^{(h,e,x)} & \boldsymbol{W}_{\text{QK}}^{(h,e,e)} \end{pmatrix},$$

for the query-key matrix of each head. Recall that $\boldsymbol{x}_i = 0$ for $i < N/2$, thus the attention weights are given by

$$\alpha_{ij}^{(h)} = \frac{\tilde{\alpha}_{ij}^{(h)}}{\sum_{l=1}^{N}\tilde{\alpha}_{il}^{(h)}}.$$

Recall from the proof of Proposition 20 that we denote the feedforward layer by $\phi : \mathbb{R}^{HD_e} \to \mathbb{R}$. With this notation, we have

$$\begin{aligned}\hat{y}_{\text{TR}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\text{TR}})_i &= \phi(\sum_{j=1}^{N} \alpha_{ij}^{(1)} \boldsymbol{z}_j, \ldots, \sum_{j=1}^{N} \alpha_{ij}^{(H)} \boldsymbol{z}_j) \\ &= \tilde{\phi}\Big(\sum_{j=1}^{q} \alpha_{ij}^{(1)} \boldsymbol{x}_j, \ldots, \sum_{j=1}^{q} \alpha_{ij}^{(H)} \boldsymbol{x}_j, (\tilde{\alpha}_{ij}^{(h)})_{h=1,j=1}^{h=H,j=N}, (\boldsymbol{z}_j)_{j=l+1}^{N}\Big).\end{aligned}$$

Therefore, using the fact that $\boldsymbol{z}_j$ and $\tilde{\alpha}_{ij}^{(h)}$ are independent of $\boldsymbol{x}_{1:q}$ for $j \geq l+1$, we have

$$\begin{aligned}R^{\text{TR}}(\boldsymbol{\Theta}_{\text{TR}}) &= \frac{1}{N} \sum_{i=N/2}^{N} \mathbb{E}\left[\left(y_i - \tilde{\phi}\Big(\sum_{j=1}^{q} \alpha_{ij}^{(1)} \boldsymbol{x}_j, \ldots, \sum_{j=1}^{q} \alpha_{ij}^{(H)} \boldsymbol{x}_j, (\tilde{\alpha}_{ij}^{(h)})_{h=1,j=1}^{h=H,j=N}, (\boldsymbol{z}_j)_{j=l+1}^{N}\Big)\right)^2\right] \\ &\geq \frac{1}{Nqd} \sum_{i=N/2}^{N} \mathbb{E}\left[\text{Var}\left(\|\boldsymbol{x}_{1:q}\|^2 \mid \left(\left\langle \boldsymbol{\alpha}_i^{(h,r)}, \boldsymbol{x}_{1:q} \right\rangle\right)_{h=1,r=1}^{h=H,r=d}, (\tilde{\alpha}_{ij}^{(h)})_{h=1,j=1}^{h=H,j=q}\right)\right] \\ &\geq \frac{1}{Nqd} \sum_{i=N/2}^{N} \mathbb{E}\left[\text{Var}\left(\|\boldsymbol{x}_{1:q}\|^2 \mid \left(\left\langle \boldsymbol{\alpha}_i^{(h,r)}, \boldsymbol{x}_{1:q} \right\rangle\right)_{h=1,r=1}^{h=H,r=d}, \left(\left\langle \boldsymbol{w}_{i,j}^{(h)}, \boldsymbol{x}_{1:q} \right\rangle\right)_{h=1,j=1}^{H,q}\right)\right] \\ &= \frac{1}{Nqd} \sum_{i=N/2}^{N} \mathbb{E}\Big[\text{var}\Big(\|\boldsymbol{x}_{1:q}\|^2 \mid \boldsymbol{V}^{(i)} \boldsymbol{x}_{1:q}\Big)\Big],\end{aligned}$$

where $\boldsymbol{\alpha}_i^{(h,r)} \in \mathbb{R}^{qd}$ such that

$$(\alpha_i^{(h,r)})_{jl} = \begin{cases} \alpha_{ij}^{(h)}, & \text{if } l = r \\ 0, & \text{if } l \neq r, \end{cases}$$

which yields $\left\langle \boldsymbol{\alpha}_i^{(h,r)}, \boldsymbol{x}_{1:q} \right\rangle = \sum_{j=1}^{q} \alpha_{ij}^{(h)} x_{jr}$, and $\boldsymbol{w}_{i,j}^{(h)} \in \mathbb{R}^{qd}$ such that

$$(w_{i,j}^{(h)})_{sl} = \begin{cases} \big(\boldsymbol{W}_{\text{QK}}^{(h,e,x)^\top} \text{enc}(i, \boldsymbol{t}_i)\big)_l, & \text{if } s = j \\ 0 & \text{if } s \neq j, \end{cases}$$

which yields $\left\langle \boldsymbol{w}_{i,j}^{(h)}, \boldsymbol{x}_{1:q} \right\rangle = \left\langle \boldsymbol{W}^{(h,e,x)^\top} \text{enc}(i, \boldsymbol{t}_i), \boldsymbol{x}_j \right\rangle$. Finally, $\boldsymbol{V}^{(i)}$ is a matrix whose rows form an orthonormal basis of $\text{span}\Big(\big(\boldsymbol{\alpha}_i^{(h,r)}\big)_{h=1,r=1}^{h=H,r=d}, \big(\boldsymbol{w}_{i,j}^{(h)}\big)_{h=1,j=1}^{h=H,j=q}\Big)$. Namely, $\boldsymbol{V}^{(i)}$ has at most $H(d+q)$ rows. Recall that

$$\boldsymbol{x}_{1:q} \mid \boldsymbol{V}^{(i)} \boldsymbol{x}_{1:q} \sim \mathcal{N}(\boldsymbol{V}^{(i)^\top} \boldsymbol{V}^{(i)} \boldsymbol{x}_{1:q}, \mathbf{I}_{qd} - \boldsymbol{V}^{(i)^\top} \boldsymbol{V}^{(i)}).$$

Once again, by Lemma 43, we conclude that $\text{var}(\|\boldsymbol{x}_{1:q}\|^2 \mid \boldsymbol{V}^{(i)} \boldsymbol{x}_{1:q}) \geq 2(qd - H(q+d))$, which completes the proof. ■

Appendix C. Details and Proofs of Section 4

Before presenting the proofs, we state the omitted setup and parameterization of the network in the next section.

C.1. Complete Setup of RNNs in the Upper Bound

A bidirectional RNN maintains, for each position in the sequence, a forward and a reverse hidden state, denoted by $(\boldsymbol{h}_i^{\rightarrow})_{i=1}^N$ and $(\boldsymbol{h}_i^{\leftarrow})_{i=1}^N$, where $\boldsymbol{h}_i^{\rightarrow}, \boldsymbol{h}_i^{\leftarrow} \in \mathbb{R}^{d_h}$. These hidden states are obtained by initializing $\boldsymbol{h}_1^{\rightarrow} = \boldsymbol{h}_N^{\leftarrow} = \boldsymbol{0}_{d_h}$ and recursively applying

$$\begin{aligned}\boldsymbol{h}_i^{\rightarrow} &= \Pi_{r_h}\big(\boldsymbol{h}_{i-1}^{\rightarrow} + f_h^{\rightarrow}(\boldsymbol{h}_{i-1}^{\rightarrow}, \boldsymbol{z}_{i-1}; \boldsymbol{\Theta}_h^{\rightarrow})\big), \quad \forall i \in \{2, \dots, N\} \\ \boldsymbol{h}_i^{\leftarrow} &= \Pi_{r_h}\big(\boldsymbol{h}_{i+1}^{\leftarrow} + f_h^{\leftarrow}(\boldsymbol{h}_{i+1}^{\leftarrow}, \boldsymbol{z}_{i+1}; \boldsymbol{\Theta}_h^{\leftarrow})\big), \quad \forall i \in \{1, \dots, N-1\},\end{aligned}$$

where $\Pi_{r_h} : \mathbb{R}^{d_h} \to \mathbb{R}^{d_h}$ is the projection $\Pi_{r_h}\boldsymbol{h} = (1 \wedge r_h/\|\boldsymbol{h}\|_2)\boldsymbol{h}$, and $f_h^{\rightarrow}$ and $f_h^{\leftarrow}$ are implemented by feedforward networks, parameterized by $\boldsymbol{\Theta}_h^{\rightarrow}$ and $\boldsymbol{\Theta}_h^{\leftarrow}$ respectively. Recall $\boldsymbol{z}_i = (\boldsymbol{x}_i^\top, \mathrm{enc}(i, \boldsymbol{t}_i)^\top)^\top$ is the encoding of $\boldsymbol{x}_i$. We remark that while we add Π_{r_h} for technical reasons, it resembles layer normalization which ensures stability of the state transitions on very long inputs; a more involved analysis can replace Π_{r_h} with standard formulations of layer normalization. Additionally, directly adding $\boldsymbol{h}_{i-1}^{\rightarrow}$ and $\boldsymbol{h}_{i+1}^{\leftarrow}$ to the output of transition functions represents residual or skip connections. The output at position i is generated by

$$y_i = f_y(\boldsymbol{h}_i^{\rightarrow}, \boldsymbol{h}_i^{\leftarrow}, \boldsymbol{z}_i; \boldsymbol{\Theta}_y),$$

which is an L_y-layer feedforward network. Specifically, we consider an RNN with deep transitions [23] and let $f_h^{\rightarrow}(\cdot; \boldsymbol{\Theta}_h^{\rightarrow})$ be an L_h-layer feedforward network, given by

$$f_h^{\rightarrow}(\cdot; \boldsymbol{\Theta}_h^{\rightarrow}) = \boldsymbol{W}_{L_h}^{\rightarrow}\sigma\big(\boldsymbol{W}_{L_h-1}^{\rightarrow} \dots \sigma(\boldsymbol{W}_2^{\rightarrow}\sigma(\boldsymbol{W}_1^{\rightarrow}(\cdot) + \boldsymbol{b}_1^{\rightarrow}) + \boldsymbol{b}_2^{\rightarrow}) \dots + \boldsymbol{b}_{L_h-1}^{\rightarrow}\big), \tag{9}$$

with $\boldsymbol{\Theta}_h^{\rightarrow} = (\boldsymbol{W}_1^{\rightarrow}, \boldsymbol{b}_1^{\rightarrow}, \dots, \boldsymbol{W}_{L_h-1}^{\rightarrow}, \boldsymbol{b}_{L_h-1}^{\rightarrow}, \boldsymbol{W}_{L_h}^{\rightarrow})$ and a similar equation for $f^{\leftarrow}(\cdot; \boldsymbol{\Theta}_h^{\leftarrow})$. As will be evident from its proof, Theorem 6 only requires $L_h, L_y = \mathcal{O}(1)$.

We denote the complete output of the RNN via

$$\hat{\boldsymbol{y}}_{\mathrm{RNN}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\mathrm{RNN}}) = (f_y(\boldsymbol{h}_1^{\rightarrow}, \boldsymbol{h}_1^{\leftarrow}, \boldsymbol{z}_1; \boldsymbol{\Theta}_y), \dots, f_y(\boldsymbol{h}_N^{\rightarrow}, \boldsymbol{h}_N^{\leftarrow}, \boldsymbol{z}_N; \boldsymbol{\Theta}_y)) \in \mathbb{R}^N.$$

We now define the constraint set of this architecture. Let

$$\Theta_{\mathrm{RNN}} = \Big\{\boldsymbol{\Theta} \,:\, \|\mathrm{vec}(\boldsymbol{\Theta})\|_2 \le R, \big\|\boldsymbol{W}_{L_h}^{\rightarrow}\big\|_{\mathrm{op}} \dots \big\|\boldsymbol{W}_{1,h}^{\rightarrow}\big\|_{\mathrm{op}} \le \alpha_N, \big\|\boldsymbol{W}_{L_h}^{\leftarrow}\big\|_{\mathrm{op}} \dots \big\|\boldsymbol{W}_{1,h}^{\leftarrow}\big\|_{\mathrm{op}} \le \alpha_N\Big\}, \tag{10}$$

where $\boldsymbol{W}_{1,h}^{\rightarrow}$ contains the first d_h columns of $\boldsymbol{W}_1^{\rightarrow}$, and the conditions above are introduced to ensure $f_h^{\rightarrow}$ and $f_h^{\leftarrow}$ are at most α_N-Lipschitz with respect to the hidden state input. One way to meet this requirement is to multiply $\boldsymbol{W}_{1,h}^{\rightarrow}$ by a factor of $\alpha_N / \prod_{l=2}^{L_h}\|\boldsymbol{W}_l^{\rightarrow}\|_{\mathrm{op}}$ in the forward pass. Without this Lipschitzness constraint, current techniques for proving uniform RNN generalization bounds will suffer from a sample complexity linear in N, see e.g. [11].

For Theorem 6 we only require $\alpha_N \le N^{-1}$. In particular, we can choose $\alpha_N = 0$ and fix $\boldsymbol{W}_{1,h}^{\rightarrow} = \boldsymbol{W}_{1,h}^{\leftarrow} = \boldsymbol{0}$, which would simplify the parameterization of the network. Namely, in our construction $f^{\rightarrow}$ and $f^{\leftarrow}$ do not need to depend on $\boldsymbol{h}^{\rightarrow}$ and $\boldsymbol{h}^{\leftarrow}$ respectively.

C.2. Overview of the Proof of Theorem 6

The following is the roadmap we will take for the proof of Theorem 6. The goal here is to implement a bi-directional RNN in such a way that

$$\boldsymbol{h}_i^{\rightarrow} \approx \big(\boldsymbol{x}_{t_1}\mathbb{1}[t_1 < i], \ldots, \boldsymbol{x}_{t_q}\mathbb{1}[t_q < i]\big),$$

and

$$\boldsymbol{h}_i^{\leftarrow} \approx \big(\boldsymbol{x}_{t_1}\mathbb{1}[t_1 > i], \ldots, \boldsymbol{x}_{t_q}\mathbb{1}[t_q > i]\big).$$

Throughout this section, we will use the notation

$$\Psi(\boldsymbol{x}, \boldsymbol{t}, i) = (\boldsymbol{x}^\top \mathbb{1}[t_1 = i], \ldots, \boldsymbol{x}^\top \mathbb{1}[t_q = i])^\top.$$

We can obtain the hidden states above through the following updates

$$\boldsymbol{h}_{i+1}^{\rightarrow} = \boldsymbol{h}_i^{\rightarrow} + \Psi(\boldsymbol{x}_i, \boldsymbol{\omega_t}, \boldsymbol{\omega}_i),$$

and

$$\boldsymbol{h}_{i-1}^{\leftarrow} = \boldsymbol{h}_i^{\leftarrow} + \Psi(\boldsymbol{x}_i, \boldsymbol{\omega_t}, \boldsymbol{\omega}_i).$$

where

$$\Psi(\boldsymbol{x}_i, \boldsymbol{\omega_t}, \boldsymbol{\omega}_i)_l = \frac{\boldsymbol{x}_i \sigma(\langle \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_l}\rangle - \delta)}{1-\delta} = \boldsymbol{x}_i \mathbb{1}[t_l = i], \quad \forall\, l \in [q]$$

where we recall $\boldsymbol{\omega_t} = (\boldsymbol{\omega}_{t_1}, \ldots, \boldsymbol{\omega}_{t_q})$, and σ is ReLU. As a result, our network must approximate

$$f_h^{\rightarrow}(\boldsymbol{h}_i^{\rightarrow}, \boldsymbol{x}_i, \boldsymbol{\omega_t}, \boldsymbol{\omega}_i; \boldsymbol{\Theta}_h^{\rightarrow}) = f_h^{\leftarrow}(\boldsymbol{h}_i^{\leftarrow}, \boldsymbol{x}_i, \boldsymbol{\omega_t}, \boldsymbol{\omega}_i; \boldsymbol{\Theta}_h^{\leftarrow}) \approx \Psi(\boldsymbol{x}_i, \boldsymbol{\omega_t}, \boldsymbol{\omega}_i).$$

A core challenge in this approximation is that if we simply control

$$\|f_h^{\rightarrow}(\boldsymbol{h}_i^{\rightarrow}, \boldsymbol{z}_i; \boldsymbol{\Theta}_h^{\rightarrow}) - \Psi(\boldsymbol{x}_i, \boldsymbol{\omega_t}, \boldsymbol{\omega}_i)\|_2 \le \varepsilon, \tag{11}$$

this error will propoagte through the forward pass, and we will have

$$\left\| \boldsymbol{h}_i^{\rightarrow} - \sum_{j=1}^{i-1} \Psi(\boldsymbol{x}_j, \boldsymbol{\omega_t}, \boldsymbol{\omega}_j) \right\|_2 \lesssim N\varepsilon.$$

As a result, we would like an implementation that satisfies the following

$$\|f_h^{\rightarrow}(\boldsymbol{h}_i^{\rightarrow}, \boldsymbol{z}_i; \boldsymbol{\Theta}_h^{\rightarrow})_l - \Psi(\boldsymbol{x}_i, \boldsymbol{\omega_t}, \boldsymbol{\omega}_i)_l\|_2 \le \begin{cases} 0 & t_l \neq i \\ \varepsilon & t_l = i. \end{cases} \tag{12}$$

Note that

$$\boldsymbol{h}_i^{\rightarrow} = \sum_{j=1}^{i-1} f_h^{\rightarrow}(\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{z}_j; \boldsymbol{\Theta}_h^{\rightarrow}).$$

Since for each $l \in [q]$, $t_l = j$ is possible for at most one $j \in [N]$, (12) implies

$$\left\| \boldsymbol{h}_i^{\rightarrow} - \sum_{j=1}^{i-1} \Psi(\boldsymbol{x}_j, \boldsymbol{\omega_t}, \boldsymbol{\omega}_j) \right\|_2 \le \sqrt{q}\varepsilon,$$

for all $i \in [N]$, hence, we can avoid dependence on N.

We can implmenet $f_h^{\rightarrow}$ to satisfy (11) with a depth three network, where the first two layers implements $\langle \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_j} \rangle$ (as a sum of Lipschitz 2-dimensional functions, an example of their approximation is given by [7, Proposition 6]), and the third performs coordinate-wise product between $\boldsymbol{x}_i$ and $\sigma(\langle \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_j} \rangle - 1/2)$ (which for each coordinate is a Lipschitz two-dimensional function). To ensure $f_h^{\rightarrow}$ satisfies (12), we can pass the outputs to a fourth layer which rectifies its input near zero to be exactly zero using ReLU activations.

To generate y_i from $\boldsymbol{h}_i^{\rightarrow}$ and $\boldsymbol{h}_i^{\leftarrow}$, we first calculate

$$\begin{aligned}\boldsymbol{h}_i &= f_{hh}(\boldsymbol{h}_i^{\rightarrow}, \boldsymbol{h}_i^{\leftarrow}, \boldsymbol{x}_i, \boldsymbol{\omega}_i, \boldsymbol{\omega}_{\boldsymbol{t}}) \\ &\approx \boldsymbol{h}_i^{\rightarrow} + \boldsymbol{h}_i^{\leftarrow} + \Psi(\boldsymbol{x}_i, \boldsymbol{\omega}_{\boldsymbol{t}}, \boldsymbol{\omega}_i) \\ &\approx (\boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_q}).\end{aligned}$$

Finally, y_i can be generated from $\boldsymbol{h}_i$ by applying the two-layer neural network from Assumption 2 that approximates $y_i = g(\boldsymbol{x}_{\boldsymbol{t}})$.

Note that the construction above has a complexity $\mathrm{poly}(d, q, \log(nN))$ (both in terms of number and weight of parameters), only depending on N up to log factors. As a result, by a simple parameter-counting approach, the sample complexity of regularized ERM would also be (almost) independent of N. We also simply use the encoding

$$\boldsymbol{z}_i = (\boldsymbol{x}_i, \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_{i1}}, \ldots, \boldsymbol{\omega}_{t_{iq}})^\top,$$

for the RNN positive result. The scaling difference with the encoding for Transofrmers is only made to simplify the exposition, as we no longer keep explicit dependence on d and q.

C.3. Approximation Upper Bounds for RNNs

As explained above, to implement $f_h^{\rightarrow}$ we first construct a depth three neural network (with two layers of non-linearity) which approximately performs the following mapping

$$\begin{pmatrix} \boldsymbol{h} \\ \boldsymbol{x} \\ \boldsymbol{\omega}_i \\ \boldsymbol{\omega}_{t_1} \\ \vdots \\ \boldsymbol{\omega}_{t_q} \end{pmatrix} \mapsto \begin{pmatrix} \boldsymbol{x} \\ \langle \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_1} \rangle \\ \vdots \\ \langle \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_q} \rangle \end{pmatrix} \mapsto \begin{pmatrix} 2\boldsymbol{x}\sigma(\langle \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_1} \rangle - 1/2) \\ \vdots \\ 2\boldsymbol{x}\sigma(\langle \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_q} \rangle - 1/2) \end{pmatrix}.$$

The first mapping will be provided by

$$\boldsymbol{\chi}_1 = \boldsymbol{A}_1 \sigma(\boldsymbol{W}_1 \boldsymbol{\chi}_0 + \boldsymbol{b}_1),$$

where $\boldsymbol{\chi}_0 = (\boldsymbol{h}^\top, \boldsymbol{x}^\top, \boldsymbol{\omega}_i^\top, \boldsymbol{\omega}_{t_1}^\top, \ldots, \boldsymbol{\omega}_{t_q}^\top)^\top \in \mathbb{R}^{d_h + d + (q+1)d_e}$, $\boldsymbol{W}_1 \in \mathbb{R}^{m_1 \times (d_h + d + (q+1)d_e)}$, $\boldsymbol{b}_1 \in \mathbb{R}^{m_1}$, and $\boldsymbol{A}_1 \in \mathbb{R}^{(d+q) \times m_1}$, with m_1 as the width of the first layer. We will use the notation

$$\boldsymbol{\chi}_1 = (\boldsymbol{\chi}_1^{\boldsymbol{x}}, \chi_1^{\boldsymbol{\omega}}(1), \ldots, \chi_1^{\boldsymbol{\omega}}(q))$$

to refer for the first d coordinates and the rest of the q coordinates of $\boldsymbol{\chi}_1$ respectively, thus ideally $\boldsymbol{\chi}_1^{\boldsymbol{x}} = \boldsymbol{x}$ and $\chi_1^{\boldsymbol{\omega}}(l) = \langle \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_l} \rangle$. The second mapping is provided by

$$\boldsymbol{\chi}_2 = \boldsymbol{A}_2 \sigma(\boldsymbol{W}_2 \boldsymbol{\chi}_1 + \boldsymbol{b}_2),$$

where $\boldsymbol{W}_2 \in \mathbb{R}^{m_2 \times (d+q)}$, $\boldsymbol{b}_2 \in \mathbb{R}^{m_2}$, and $\boldsymbol{A}_2 \in \mathbb{R}^{dq \times m_2}$. We will similarly use the notation $\boldsymbol{\chi}_2 = (\boldsymbol{\chi}_2(1), \ldots, \boldsymbol{\chi}_2(q))$, where our goal is to have $\boldsymbol{\chi}_2(l) \approx 2\boldsymbol{x}\sigma(\langle \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_l} \rangle - 1/2)$. To implement the first mapping, we rely on the following lemma.

Lemma 21 *Let σ be the ReLU activation. For any $\varepsilon > 0$ and positive integer d_e, there exists $m = \mathcal{O}(d_e^3(\log(d_e/\varepsilon)/\varepsilon)^2)$, $\boldsymbol{a} \in \mathbb{R}^m$, $\boldsymbol{W} \in \mathbb{R}^{m \times 2d_e}$, and $\boldsymbol{b} \in \mathbb{R}^m$, such that*

$$\sup_{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2 \in \mathbb{S}^{d_e - 1}} \left| \langle \boldsymbol{\omega}_1, \boldsymbol{\omega}_2 \rangle - \boldsymbol{a}^\top \sigma\left(\boldsymbol{W} \begin{pmatrix} \boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \end{pmatrix} + \boldsymbol{b} \right) \right| \leq \varepsilon,$$

and

$$\|\boldsymbol{a}\|_2 \leq \mathcal{O}\Big(d_e^{5/2} (\log(d_e/\varepsilon)/\varepsilon)^{3/2} / \sqrt{m} \Big), \quad \left\| \boldsymbol{W}^\top \right\|_{1,\infty} \leq 1, \quad \|\boldsymbol{b}\|_\infty \leq 1.$$

Proof Consider the mapping $e_{1j}, e_{2j} \mapsto e_{1j} e_{2j}$. Note that when $|e_{1j}| \leq 1$ and $|e_{2j}| \leq 1$, this mapping is $\sqrt{2}$-Lipschitz, and the output is bounded between $[-1, 1]$. Then, by Lemma 45, for every $\varepsilon_j > 0$, there exists $m_j \leq \mathcal{O}((1/\varepsilon_j \log(1/\varepsilon_j))^2)$, $\boldsymbol{a}_j \in \mathbb{R}^{m_j}$, $\boldsymbol{W}_j \in \mathbb{R}^{m_j \times 2d_e}$, and $\boldsymbol{b}_j \in \mathbb{R}^{m_j}$, such that

$$\sup_{|e_{1j}| \leq 1, |e_{2j}| \leq 1} \left| e_{1j} e_{2j} - \sum_{l=1}^{m} a_{jl} \sigma\Big(\Big\langle \boldsymbol{w}_{jl}, (\boldsymbol{\omega}_1^\top, \boldsymbol{\omega}_2^\top)^\top \Big\rangle + b_{jl} \Big) \right| \leq \varepsilon_j,$$

$\|\boldsymbol{a}_j\|_2 \leq \mathcal{O}\big((\log(1/\varepsilon_j)/\varepsilon_j)^{3/2} / \sqrt{m_j} \big)$, $\|\boldsymbol{b}_j\|_\infty \leq 1$, and $\|\boldsymbol{w}_{jl}\|_1 \leq 1$. Specifically, the only non-zero coordinates of $\boldsymbol{w}_{jl}$ are the jth and $d_e + j$th coordinates.

Let $\varepsilon_j = \varepsilon / d_e$ and $m = \sum_{j=1}^{d_e} m_j = \mathcal{O}(d_e^3(\log(d_e/\varepsilon)/\varepsilon)^2)$. Construct $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^m$ and $\boldsymbol{W} \in \mathbb{R}^{m \times 2d_e}$ by concatenating $(\boldsymbol{a}_j)$, $(\boldsymbol{b}_j)$, and $(\boldsymbol{W}_j)$ respectively. The resulting network satisfies

$$\sup_{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2 \in \mathbb{S}^{d_e - 1}} \left| \langle \boldsymbol{\omega}_1, \boldsymbol{\omega}_2 \rangle - \boldsymbol{a}^\top \sigma\left(\boldsymbol{W} \begin{pmatrix} \boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \end{pmatrix} + \boldsymbol{b} \right) \right| \leq \varepsilon,$$

while $\|\boldsymbol{a}\|_2 \leq \mathcal{O}\big(d_e^{5/2} (\log(d_e/\varepsilon)/\varepsilon)^{3/2} / \sqrt{m} \big)$, $\|\boldsymbol{b}\|_\infty \leq 1$, and $\left\| \boldsymbol{W}^\top \right\|_{1,\infty} \leq 1$, completing the proof. ■

We can now specify $\boldsymbol{A}_1, \boldsymbol{W}_1$, and $\boldsymbol{b}_1$ in our construction.

Lemma 22 *For any $\varepsilon > 0$, let $\bar{m}_1 = \mathcal{O}(d_e^3(\log(d_e/\varepsilon)/\varepsilon)^2)$ and $m_1 = 2d + q\bar{m}_1$. Then, there exist $\boldsymbol{A}_1 \in \mathbb{R}^{(d+q) \times m_1}$, $\boldsymbol{W}_1 \in \mathbb{R}^{m_1 \times (d_h + d + (q+1)d_e)}$, and $\boldsymbol{b}_1 \in \mathbb{R}^{m_1}$, given by Equations (13) to (17), such that*

$$\boldsymbol{\chi}_1^{\boldsymbol{x}} = \boldsymbol{x}, \quad |\chi_1^{\boldsymbol{\omega}}(l) - \langle \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_l} \rangle| \leq \varepsilon,$$

for all $\boldsymbol{h} \in \mathbb{R}^{d_h}$, $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{\omega}_i, (\boldsymbol{\omega}_{t_j})_{j \in [q]} \in \mathbb{S}^{d_e - 1}$, and $l \in [q]$. Furthermore, we have the following guarantees

$$\left\| \boldsymbol{W}_1^\top \right\|_{1,\infty} \leq \mathcal{O}(1), \quad \|\boldsymbol{b}_1\|_\infty \leq \mathcal{O}(1), \quad \left\| \boldsymbol{A}_1^\top \right\|_{1,\infty} \leq \mathcal{O}(d_e^{5/2} (\log(d_e/\varepsilon)/\varepsilon)^{3/2}).$$

Proof We define the decompositions

$$\boldsymbol{W}_1 = \begin{pmatrix} \boldsymbol{W}_{11} \\ \boldsymbol{W}_{12} \end{pmatrix}, \quad \boldsymbol{b}_1 = \begin{pmatrix} \boldsymbol{b}_{11} \\ \boldsymbol{b}_{12} \end{pmatrix}, \quad \boldsymbol{A}_1 = \begin{pmatrix} \boldsymbol{A}_{11} \\ \boldsymbol{A}_{12} \end{pmatrix}, \tag{13}$$

where $\boldsymbol{W}_{11} \in \mathbb{R}^{2d\times(d_h+d+d_e)}$, $\boldsymbol{W}_{12} \in \mathbb{R}^{q\bar{m}_1\times(d_h+d+d_e)}$, $\boldsymbol{b}_{11} \in \mathbb{R}^{2d}$, $\boldsymbol{b}_{12} \in \mathbb{R}^{q\bar{m}_1}$, $\boldsymbol{A}_{11} \in \mathbb{R}^{d\times m_1}$, and $\boldsymbol{A}_{12} \in \mathbb{R}^{q\times m_1}$. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d$ denote the standard basis of $\mathbb{R}^d$, and notice that $\sigma(z) - \sigma(-z) = z$. Therefore, we can implement the identity part of the mapping by letting

$$\boldsymbol{W}_{11} = \begin{pmatrix} \boldsymbol{0}_{d_h} & \boldsymbol{v}_1^\top & \boldsymbol{0}_{(q+1)d_e}^\top \\ \boldsymbol{0}_{d_h} & -\boldsymbol{v}_1^\top & \boldsymbol{0}_{(q+1)d_e}^\top \\ \vdots & \vdots & \\ \boldsymbol{0}_{d_h} & \boldsymbol{v}_d^\top & \boldsymbol{0}_{(q+1)d_e}^\top \\ \boldsymbol{0}_{d_h} & -\boldsymbol{v}_d^\top & \boldsymbol{0}_{(q+1)d_e}^\top \end{pmatrix}, \tag{14}$$

as well as

$$\boldsymbol{b}_1 = \boldsymbol{0}_{2d}, \quad \text{and} \quad \boldsymbol{A}_{11} = \begin{pmatrix} 1 & -1 & 0 & 0 & \ldots & 0 & \boldsymbol{0}_{q\bar{m}_1}^\top \\ 0 & 0 & 1 & -1 & \ldots & 0 & \boldsymbol{0}_{q\bar{m}_1}^\top \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & 0 & 1 & -1 & \boldsymbol{0}_{q\bar{m}_1}^\top \end{pmatrix} \tag{15}$$

Notice that $\left\|\boldsymbol{W}_{11}^\top\right\|_{1,\infty} = 1$ and $\left\|\boldsymbol{A}_{11}^\top\right\|_{1,\infty} = 2$. To implement the inner product part of the mapping, we take the construction of weights, biases, and second layer weights from Lemma 21, and rename them as $\tilde{\boldsymbol{W}}_1 \in \mathbb{R}^{\bar{m}_1\times 2d_e}$, $\tilde{\boldsymbol{b}}_1 \in \mathbb{R}^{\bar{m}_1}$, and $\tilde{\boldsymbol{a}}_1 \in \mathbb{R}^{\bar{m}_1}$. Let us introduce the decomposition $\tilde{\boldsymbol{W}}_1 = \begin{pmatrix} \tilde{\boldsymbol{W}}_{11} & \tilde{\boldsymbol{W}}_{12} \end{pmatrix}$, where $\tilde{\boldsymbol{W}}_{11}, \tilde{\boldsymbol{W}}_{12} \in \mathbb{R}^{\bar{m}_1\times d_e}$. With this decomposition, we can separate the projections applied to the first and second vectors in Lemma 21. We can then define

$$\boldsymbol{W}_{12} = \begin{pmatrix} \boldsymbol{0}_{\bar{m}_1\times(d_h+d)} & \tilde{\boldsymbol{W}}_{11} & \tilde{\boldsymbol{W}}_{12} & \boldsymbol{0}_{\bar{m}_1\times d_e} & \ldots & \boldsymbol{0}_{\bar{m}_1\times d_e} \\ \boldsymbol{0}_{\bar{m}_1\times(d_h+d)} & \tilde{\boldsymbol{W}}_{11} & \boldsymbol{0}_{\bar{m}_1\times d_e} & \tilde{\boldsymbol{W}}_{12} & \ldots & \boldsymbol{0}_{\bar{m}_1\times d_e} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{0}_{\bar{m}_1\times(d_h+d)} & \tilde{\boldsymbol{W}}_{11} & \boldsymbol{0}_{\bar{m}_1\times d_e} & \boldsymbol{0}_{\bar{m}_1\times d_e} & \ldots & \tilde{\boldsymbol{W}}_{12} \end{pmatrix}, \tag{16}$$

as well as

$$\boldsymbol{b}_{12} = \begin{pmatrix} \tilde{\boldsymbol{b}}_1 \\ \vdots \\ \tilde{\boldsymbol{b}}_1 \end{pmatrix}, \quad \text{and} \quad \boldsymbol{A}_{12} = \begin{pmatrix} \boldsymbol{0}_{2d}^\top & \tilde{\boldsymbol{a}}_1^\top & \boldsymbol{0}_{\bar{m}_1}^\top & \ldots & \boldsymbol{0}_{\bar{m}_1}^\top \\ \boldsymbol{0}_{2d}^\top & \boldsymbol{0}_{\bar{m}_1}^\top & \tilde{\boldsymbol{a}}_1^\top & \ldots & \boldsymbol{0}_{\bar{m}_1}^\top \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{0}_{2d}^\top & \boldsymbol{0}_{\bar{m}_1}^\top & \ldots & \boldsymbol{0}_{\bar{m}_1}^\top & \tilde{\boldsymbol{a}}_1^\top \end{pmatrix}. \tag{17}$$

From Lemma 21, we have $\left\|\boldsymbol{W}_{12}^\top\right\|_{1,\infty} \leq 1$, $\|\boldsymbol{b}_{12}\|_\infty \leq 1$, and

$$\left\|\boldsymbol{A}_{12}^\top\right\|_{1,\infty} = \|\tilde{\boldsymbol{a}}_1\|_1 \leq \mathcal{O}(d_e^{5/2}(\log(d_e/\varepsilon)/\varepsilon)^{3/2}),$$

which completes the proof. ■

To introduce the construction of the next layer, we rely on the following lemma which establishes the desired approximation for a single coordinate, the proof of which is similar to that of Lemma 21.

Lemma 23 *Let σ be the ReLU activation. Suppose $|h| \leq r_\infty^h$, $|x| \leq r_\infty^x$ and $|z| \leq 1$. Let $R \coloneqq \sqrt{1 + {r_\infty^x}^2 + {r_\infty^h}^2}$. For any $\varepsilon > 0$, there exists $m = \mathcal{O}(R^6(\log(R/\varepsilon)/\varepsilon)^3)$, $\boldsymbol{a} \in \mathbb{R}^m$,*

$\boldsymbol{W} \in \mathbb{R}^{m\times 2}$, and $\boldsymbol{b} \in \mathbb{R}^m$, such that

$$\sup_{|h|\le r_\infty^h, |x|\le r_\infty^x, |z|\le 1} \left| h + 2x\sigma(z-1/2) - \boldsymbol{a}^\top \sigma\Big(\boldsymbol{W}(h,x,z)^\top + \boldsymbol{b}\Big)\right| \le \varepsilon$$

and

$$\|\boldsymbol{a}\|_2 \le \mathcal{O}\big(R^6(\log(R/\varepsilon)/\varepsilon)^2/\sqrt{m}\big), \quad \left\|\boldsymbol{W}^\top\right\|_{1,\infty} \le R^{-1}, \quad \|\boldsymbol{b}\|_\infty \le 1.$$

Additionally, if $r_\infty^h = 0$, we have the improved bounds

$$m = \mathcal{O}\big(R^4(\log(R/\varepsilon)/\varepsilon)^2\big), \quad \|\boldsymbol{a}\|_2 \le \mathcal{O}\big(R^5(\log(R/\varepsilon)/\varepsilon)^{3/2}/\sqrt{m}\big)$$

Proof Note that $(h,x,z) \mapsto h + 2x\sigma(z-1/2)$ is $2R$-Lipschitz, and $|h + 2x\sigma(z-1/2)| \le R$. The proof follows from Lemma 45 with dimension 3 when $r_\infty^h \neq 0$ and dimension 2 otherwise. ■

With that, we can now construct the weights for the second mapping in the network.

Lemma 24 *Suppose $\|\boldsymbol{\chi}_1^{\boldsymbol{x}}\|_\infty \le r_x$ and $\max_l |\chi^{\boldsymbol{\omega}}(l)| \le 1$. Let $R \coloneqq \sqrt{1+r_x^2}$. Then, for every $\varepsilon > 0$ and absolute constant $\delta \in (0,1)$, there exists $\bar{m}_2 \le \mathcal{O}(R^4(\log(R/\varepsilon)/\varepsilon)^{3/2})$, $m_2 \coloneqq qd\bar{m}_2$, and $\boldsymbol{A}_2 \in \mathbb{R}^{d_h \times m_2}$, $\boldsymbol{W}_2 \in \mathbb{R}^{m_2\times(d+q)}$, and $\boldsymbol{b}_2 \in \mathbb{R}^{m_2}$ given by Equations (18) and (19) such that*

$$\|\boldsymbol{\chi}_2(l) - 2\boldsymbol{\chi}_1^{\boldsymbol{x}}\sigma(\chi_1^{\boldsymbol{\omega}}(l) - 1/2)\|_\infty \le \varepsilon,$$

for all such $\boldsymbol{\chi}_1$ and $l \in [q]$, where we recall $\boldsymbol{\chi}_2 = \boldsymbol{A}_2\sigma(\boldsymbol{W}_2\boldsymbol{\chi}_1 + \boldsymbol{b}_2)$. Moreover, we have

$$\left\|\boldsymbol{A}_2^\top\right\|_{1,\infty} \le \mathcal{O}(R^4(\log(R/\varepsilon)/\varepsilon)^{3/2}), \quad \left\|\boldsymbol{W}_2^\top\right\|_{1,\infty} \le R^{-1}, \quad \|\boldsymbol{b}_2\|_\infty \le 1.$$

Proof Let $\tilde{\boldsymbol{W}} = \begin{pmatrix}\tilde{\boldsymbol{w}}_{21} & \tilde{\boldsymbol{w}}_{22}\end{pmatrix}$, $\tilde{\boldsymbol{b}}$, and $\tilde{\boldsymbol{a}}$ be the weights obtained from Lemma 23, where $\tilde{\boldsymbol{w}}_{21}, \tilde{\boldsymbol{w}}_{22}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{a}} \in \mathbb{R}^{\bar{m}_2}$. To construct $\boldsymbol{W}_2$ and $\boldsymbol{b}_2$, we let

$$\boldsymbol{W}_2 = \begin{pmatrix}\boldsymbol{W}_2(1,1)\\ \vdots \\ \boldsymbol{W}_2(1,d)\\ \vdots \\ \boldsymbol{W}_2(q,1)\\ \vdots \\ \boldsymbol{W}_2(q,d)\end{pmatrix}, \quad \boldsymbol{b}_{22} = \begin{pmatrix}\boldsymbol{b}_2(1,1)\\ \vdots \\ \boldsymbol{b}_2(1,d)\\ \vdots \\ \boldsymbol{b}_2(q,1)\\ \vdots \\ \boldsymbol{b}_2(q,d)\end{pmatrix}. \tag{18}$$

where $\boldsymbol{W}_2(l,j) \in \mathbb{R}^{\bar{m}_2\times(d+q)}$ is given by

$$\boldsymbol{W}_2(l,j) = \begin{pmatrix}\boldsymbol{0}_{\bar{m}_2\times(j-1)} & \tilde{\boldsymbol{w}}_{21} & \boldsymbol{0}_{\bar{m}_2\times(d-j)} & \boldsymbol{0}_{\bar{m}_2\times(l-1)} & \tilde{\boldsymbol{w}}_{22} & \boldsymbol{0}_{\bar{m}_2\times(q-l)}\end{pmatrix},$$

and $\boldsymbol{b}_2(l,j) = \tilde{\boldsymbol{b}}_2$. Consequently, $\left\|\boldsymbol{W}_2^\top\right\|_{1,\infty} \le 1$ and $\|\boldsymbol{b}_2\|_\infty \le 1$. Finally, we have

$$\boldsymbol{A}_2 = \begin{pmatrix}\tilde{\boldsymbol{a}}_2^\top & \boldsymbol{0}_{\bar{m}_2}^\top & \dots & \boldsymbol{0}_{\bar{m}_2}^\top\\ \boldsymbol{0}_{\bar{m}_2}^\top & \tilde{\boldsymbol{a}}_2^\top & \dots & \boldsymbol{0}_{\bar{m}_2}^\top\\ \vdots & \vdots & \vdots & \vdots\\ \boldsymbol{0}_{\bar{m}_2}^\top & \dots & \boldsymbol{0}_{\bar{m}_2}^\top & \tilde{\boldsymbol{a}}_2^\top\end{pmatrix}. \tag{19}$$

Consequently, we obtain $\left\|\boldsymbol{A}_2^\top\right\|_{1,\infty} \le \mathcal{O}(R^4(\log(R/\varepsilon)/\varepsilon)^{3/2})$, completing the proof. ■

We are now ready to provide the four-layer feedforward construction of $f^{\rightarrow}(\boldsymbol{h}, \boldsymbol{x}, \boldsymbol{t}; \boldsymbol{\Theta}_h^{\rightarrow})$.

Proposition 25 *Let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_1}, \ldots, \boldsymbol{\omega}_{t_q})$. Then, for every $\varepsilon > 0$, there exists a feedforward network with $L_h = 4$ layers given by*

$$f^{\rightarrow}(\boldsymbol{h}, \boldsymbol{z}; \boldsymbol{\Theta}_h^{\rightarrow}) = \boldsymbol{W}_{L_h}\sigma\Big(\ldots\sigma\big(\boldsymbol{W}_2\sigma\big(\boldsymbol{W}_1(\boldsymbol{h}^\top, \boldsymbol{z}^\top)^\top + \boldsymbol{b}_1\big) + \boldsymbol{b}_2\big)\ldots\Big)$$

where $\boldsymbol{W}_i \in \mathbb{R}^{m_i \times m_{i-1}}, \boldsymbol{b}_i \in \mathbb{R}_i^m$ for $i \in \{2, \ldots, L_h - 1\}$, $\boldsymbol{W}_1 \in \mathbb{R}^{m_1 \times d_h + d + (q+1)d_e}, \boldsymbol{b}_1 \in \mathbb{R}^{m_1}$, and $\boldsymbol{W}_{L_h} \in \mathbb{R}^{d_h \times m_{L_h - 1}}$ that satisfies the following:

1. *If $t_l = i$, then*
$$\Big\|f^{\rightarrow}(\boldsymbol{h}, \boldsymbol{z}; \hat{\boldsymbol{\Theta}}_h^{\rightarrow})_l - \boldsymbol{x}\Big\|_2 \leq \varepsilon$$
2. *Else $f^{\rightarrow}(\boldsymbol{h}, \boldsymbol{z}; \hat{\boldsymbol{\Theta}}^{\rightarrow})_l = \boldsymbol{0}_d$,*

for all $l \in [q]$, $\boldsymbol{h} \in \mathbb{R}^{d_h}$ and $\|\boldsymbol{x}\|_2 \leq r_x$. Additionally $\|\boldsymbol{W}_i\|_{\mathrm{F}} \leq \mathrm{poly}(r_x, D_e, \varepsilon^{-1})$ for all $i \in [L_h]$ and $m_i, \|\boldsymbol{b}_i\|_2 \leq \mathrm{poly}(r_x, D_e, \varepsilon^{-1})$ for all $i \in [L_h - 1]$, where we recall $D_e = d + (q+1)d_e$.

Proof Let $\tilde{\boldsymbol{A}}_1 \in \mathbb{R}^{(d+q)\times m_1}, \tilde{\boldsymbol{W}}_1 \in \mathbb{R}^{m_1 \times (d_h + d + (q+1)d_e)}, \tilde{\boldsymbol{b}}_1 \in \mathbb{R}^{m_1}$ be given by Lemma 22 with error parameter ε_1 and $\tilde{\boldsymbol{A}}_2 \in \mathbb{R}^{d_h \times m_2}, \tilde{\boldsymbol{W}}_2 \in \mathbb{R}^{m_2 \times (d+q)}, \tilde{\boldsymbol{b}}_2 \in \mathbb{R}^{m_2}$ be given by Lemma 24 with error parameter ε_2. Recall that

$$\boldsymbol{\chi}_1 = \tilde{\boldsymbol{A}}_1\sigma\big(\tilde{\boldsymbol{W}}_1\boldsymbol{\chi}_0 + \tilde{\boldsymbol{b}}_1\big), \quad \boldsymbol{\chi}_2 = \tilde{\boldsymbol{A}}_2\sigma\big(\tilde{\boldsymbol{W}}_2\boldsymbol{\chi}_1 + \tilde{\boldsymbol{b}}_2\big).$$

By the triangle inequality,

$$\begin{aligned}
\Big\|\Psi(\boldsymbol{x}, \boldsymbol{t}, i) - \tilde{\boldsymbol{A}}_2\sigma\big(\tilde{\boldsymbol{W}}_2\boldsymbol{\chi}_1 + \tilde{\boldsymbol{b}}_2\big)\Big\|_\infty \leq& \Big\|\Psi(\boldsymbol{x}, \boldsymbol{t}, i) - \tilde{\boldsymbol{A}}_2\sigma\big(\tilde{\boldsymbol{W}}_2\bar{\boldsymbol{\chi}}_1 + \tilde{\boldsymbol{b}}_2\big)\Big\|_\infty \\
&+ \Big\|\tilde{\boldsymbol{A}}_2\sigma\big(\tilde{\boldsymbol{W}}_2\bar{\boldsymbol{\chi}}_1 + \tilde{\boldsymbol{b}}_2\big) - \tilde{\boldsymbol{A}}_2\sigma\big(\tilde{\boldsymbol{W}}_2\boldsymbol{\chi}_1 + \tilde{\boldsymbol{b}}_2\big)\Big\|_\infty \\
\leq& \varepsilon_2 + \Big\|\tilde{\boldsymbol{A}}_2^\top\Big\|_{1,\infty}\Big\|\tilde{\boldsymbol{W}}_2\Big\|_{1,\infty}\|\boldsymbol{\chi}_1 - \bar{\boldsymbol{\chi}}_1\|_\infty \\
\leq& \varepsilon_2 + \Big\|\tilde{\boldsymbol{A}}_2\Big\|_{1,\infty}\Big\|\tilde{\boldsymbol{W}}_2\Big\|_{1,\infty}\varepsilon_1,
\end{aligned}$$

where $\bar{\boldsymbol{\chi}}_1 = (\boldsymbol{x}^\top, \langle\boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_1}\rangle, \ldots, \langle\boldsymbol{\omega}_i, \boldsymbol{\omega}_{t_q}\rangle)^\top$. By letting $\varepsilon_2 = \varepsilon/4$, we obtain

$$m_2, \Big\|\tilde{\boldsymbol{A}}_2\Big\|_{\mathrm{F}}, \Big\|\tilde{\boldsymbol{W}}_2\Big\|_{\mathrm{F}}, \Big\|\tilde{\boldsymbol{b}}_2\Big\|_2 \leq \mathrm{poly}(r_x, D_e, \varepsilon^{-1}).$$

Similarly, we can let $\varepsilon_1 = \varepsilon/\big(4\Big\|\tilde{\boldsymbol{A}}_2\Big\|_{1,\infty}\Big\|\tilde{\boldsymbol{W}}_2\Big\|_{1,\infty}\big)$, which yields

$$m_1, \Big\|\tilde{\boldsymbol{A}}_2\Big\|_{\mathrm{F}}, \Big\|\tilde{\boldsymbol{W}}_2\Big\|_{\mathrm{F}}, \Big\|\tilde{\boldsymbol{b}}_2\Big\|_2 \leq \mathrm{poly}(r_x, D_e, \varepsilon^{-1}).$$

Let

$$\boldsymbol{W}_2 = \tilde{\boldsymbol{W}}_2\tilde{\boldsymbol{A}}_1, \qquad \boldsymbol{W}_1 = \tilde{\boldsymbol{W}}_1, \qquad \boldsymbol{b}_1 = \tilde{\boldsymbol{b}}_1, \qquad \boldsymbol{b}_2 = \tilde{\boldsymbol{b}}_2.$$

Then,

$$\boldsymbol{\chi}_2 = \tilde{\boldsymbol{A}}_2\sigma(\boldsymbol{W}_2\sigma(\boldsymbol{W}_1(\boldsymbol{h}^\top\boldsymbol{z}^\top)^\top + \boldsymbol{b}_1) + \boldsymbol{b}_2),$$

satisfies $\|\boldsymbol{\chi}_2 - \Psi(\boldsymbol{x}, \boldsymbol{t}, i)\|_\infty \leq \varepsilon/2$ for all $\|\boldsymbol{x}\|_2 \leq r_x$.

Recall that when $t_l \neq i$ for some $l \in [q]$, we would like to guarantee the output of the network to be equal to $\Psi(\boldsymbol{x}, \boldsymbol{t}, i)_l = \mathbf{0}_d$. To do so, we rely on the fact that $z \mapsto \sigma(z-b) - \sigma(-z-b)$ is zero for $|z| \leq b$, and has an L_∞ distance of b from the identity, i.e. $|z - \sigma(z-b) + \sigma(-z-b)| \leq b$. This mapping needs to be applied element-wise to $\boldsymbol{\chi}_2$. Let $\tilde{\boldsymbol{W}}_3 \in \mathbb{R}^{2d_h \times d_h}, \boldsymbol{b}_3 \in \mathbb{R}^{2d_h}$, and $\boldsymbol{W}_4 \in \mathbb{R}^{d_h \times 2d_h}$ via

$$\tilde{\boldsymbol{W}}_3 = \begin{pmatrix} \boldsymbol{v}_1^\top \\ -\boldsymbol{v}_1^\top \\ \vdots \\ \boldsymbol{v}_d^\top \\ -\boldsymbol{v}_d^\top \end{pmatrix}, \quad \boldsymbol{b}_3 = -\frac{\varepsilon}{2}\mathbf{1}_{2d_h}, \quad \boldsymbol{W}_4 = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix}.$$

As a result, $\boldsymbol{\chi}_3 = \boldsymbol{W}_4 \sigma(\tilde{\boldsymbol{W}}_3 \boldsymbol{\chi}_2 + \boldsymbol{b}_3)$ satisfies

$$|(\chi_3)_j - (\chi_2)_j| \leq \begin{cases} 0 & |(\chi_2)_j| \leq \varepsilon/2 \\ \varepsilon/2 & |(\chi_2)_j| > \varepsilon/2 \end{cases}, \quad \forall j \in [d_h]. \tag{20}$$

We thus make two observations. First, $\|\boldsymbol{\chi}_3 - \boldsymbol{\chi}_2\|_\infty \leq \varepsilon/2$, and consequently $\|\boldsymbol{\chi}_3(l) - \Psi(\boldsymbol{x}, \boldsymbol{t}, i)_l\|_\infty \leq \varepsilon$ for all $l \in [q]$. Second, when $t_l \neq i$, we have $\Psi(\boldsymbol{x}, \boldsymbol{t}, i)_l = \mathbf{0}_d$ and $|\chi_2(l)_j| \leq \varepsilon/2$ for all $j \in [d]$ since $\|\boldsymbol{\chi}_2(l) - \Psi(\boldsymbol{x}, \boldsymbol{t}, i)_l\|_\infty \leq \varepsilon/2$. Consequently, by the first case in (20), we have $\chi_3(l)_j = 0$ for all $j \in [d]$. We can summarize these two observations as follows

$$\|\boldsymbol{\chi}_3(l) - \Psi(\boldsymbol{x}, \boldsymbol{t}, i)_l\|_\infty \leq \begin{cases} 0 & t_l \neq i \\ \varepsilon & t_l = i \end{cases},$$

which completes the proof. ■

With the above implementation of $f^{\rightarrow}(\boldsymbol{h}, \boldsymbol{z}; \boldsymbol{\Theta}_h^{\rightarrow})$, we have the following guarantee on $\boldsymbol{h}_i^{\rightarrow}$ for all $i \in [N]$.

Corollary 26 *Let $f_h^{\rightarrow}$ be given by the construction in Proposition 25, and suppose $r_h \geq \sqrt{q}(r_x + \sqrt{d}\varepsilon)$. Then, $\boldsymbol{h}_i^{\rightarrow}$ satisfies the following guarantees for all $i \in [N]$ and $l \in [q]$:*

1. *If $t_l \geq i$, then $\boldsymbol{h}_i^{\rightarrow}(l) = \mathbf{0}_d$*
2. *If $t_l < i$, then $\|\boldsymbol{h}_i^{\rightarrow}(l) - \boldsymbol{x}_{t_l}\|_\infty \leq \varepsilon$.*

Proof We can prove the statement by induction. Note that it holds for $i = 1$ since $\boldsymbol{h}_1^{\rightarrow} = \mathbf{0}_d$. For the induction step, suppose it holds up to some i, and recall

$$\boldsymbol{h}_{i+1}^{\rightarrow} = \boldsymbol{h}_i^{\rightarrow} + f_h^{\rightarrow}(\boldsymbol{h}_i^{\rightarrow}, \boldsymbol{z}_i; \boldsymbol{\Theta}_h^{\rightarrow}).$$

- If $t_l \geq i+1$, then $\boldsymbol{h}_i^{\rightarrow}(l) = \mathbf{0}_d$ and $f_h^{\rightarrow}(\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{z}_i; \boldsymbol{\Theta}_h^{\rightarrow}) = \mathbf{0}_d$ by Proposition 25.
- If $t_l < i < i+1$, then $\|\boldsymbol{h}_i^{\rightarrow}(l) - \boldsymbol{x}_{t_l}\|_\infty \leq \varepsilon$ by induction hypothesis, and $f_h^{\rightarrow}(\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{z}_j; \boldsymbol{\Theta}_h^{\rightarrow}) = \mathbf{0}_d$.

- Finally, if $t_l = i < i+1$, then $\boldsymbol{h}_i^{\rightarrow}(l) = 0$ and $\|f_h^{\rightarrow}(\boldsymbol{h}_i^{\rightarrow}, \boldsymbol{z}_i; \boldsymbol{\Theta}_h^{\rightarrow}) - \boldsymbol{x}_{t_l}\|_\infty \le \varepsilon$.

Note that since $\left\|\boldsymbol{h}_j^{\rightarrow}\right\|_2 \le r_h$ for all $j \in [N]$, the projection Π_{r_h} will always be identity through the forward pass, concluding the proof. ■

By symmetry, the same construction for $f_h^{\leftarrow}$ would yield a similar guarantee on $\boldsymbol{h}_j^{\leftarrow}$.
The last step is to design $f_y(\boldsymbol{h}^{\rightarrow}, \boldsymbol{h}^{\leftarrow}, \boldsymbol{z}; \boldsymbol{\Theta}_y)$ such that

$$f_y(\boldsymbol{h}^{\rightarrow}, \boldsymbol{h}^{\leftarrow}, \boldsymbol{z}_i; \boldsymbol{\Theta}_y) \approx g\big(\boldsymbol{h}^{\rightarrow} + \boldsymbol{h}^{\leftarrow} + (\boldsymbol{x}_i^\top \mathbb{1}[t_1 = i], \ldots, \boldsymbol{x}_i^\top \mathbb{1}[t_q = i])^\top\big).$$

The following proposition provides the end-to-end RNN guarantee for approximating simple qSTR models.

Proposition 27 *Suppose g satisfies Assumption 2. Then there exist RNN weights $\boldsymbol{\Theta}_{\text{RNN}}$ with $\text{vec}(\boldsymbol{\Theta}_{\text{RNN}}) \in \mathbb{R}^p$ (i.e. with p parameters) and $r_h \ge \sqrt{q} r_x + \sqrt{\varepsilon_{\text{2NN}}}/(r_a r_w)$, such that*

$$\sup_{i \in [N]} \left| g(\boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_q}) - \hat{y}(\boldsymbol{p}; \boldsymbol{\Theta}_{\text{RNN}})_i \right|^2 \le 4\varepsilon_{\text{2NN}} \tag{21}$$

for all $\boldsymbol{t} \in [N]^q$ and $\|\boldsymbol{x}_j\|_2 \le r_x$ for all $j \in [N]$. Additionally, we have

$$\|\text{vec}(\boldsymbol{\Theta}_{\text{RNN}})\|_2 \le \text{poly}(r_x, D_e, r_w, r_a, \varepsilon_{\text{2NN}}^{-1}), \quad p \le \text{poly}(r_x, D_e, m_g, r_w, r_a, \varepsilon_{\text{2NN}}^{-1}), \tag{22}$$

and $f_h^{\rightarrow}, f_h^{\leftarrow}$ do not depend on $\boldsymbol{h}^{\rightarrow}$ and $\boldsymbol{h}^{\leftarrow}$, namely the first d_h columns of $\boldsymbol{W}_1^{\rightarrow}$ and $\boldsymbol{W}_1^{\leftarrow}$ that are multiplied by $\boldsymbol{h}^{\rightarrow}$ and $\boldsymbol{h}^{\leftarrow}$ respectively are zero.

Proof As the proof of this proposition mostly follows from the previous proofs in this section, we only state the procedure for obtaining the desired weights.

Let $(\boldsymbol{v}_j)_{j=1}^{d_h}$ denote the standard basis of $\mathbb{R}^{d_h}$. Since $\sigma(z) - \sigma(-z) = z$, we can implement the identity mapping in $\mathbb{R}^{d_h}$ via a two-layer feedforward network with the following weights

$$\boldsymbol{W}_{\text{id}} = \begin{pmatrix} \boldsymbol{v}_1^\top \\ -\boldsymbol{v}_1^\top \\ \vdots \\ \boldsymbol{v}_{d_h}^\top \\ -\boldsymbol{v}_{d_h}^\top \end{pmatrix}, \quad \boldsymbol{b}_{\text{id}} = \boldsymbol{0}_{2d_h}, \boldsymbol{A}_{\text{id}} = \begin{pmatrix} 1 & -1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & -1 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 1 & -1 \end{pmatrix},$$

where $\boldsymbol{W}_{\text{id}} \in \mathbb{R}^{2d_h \times d_h}$, $\boldsymbol{b}_{\text{id}} \in \mathbb{R}^{2d_h}$, and $\boldsymbol{A}_{\text{id}} \in \mathbb{R}^{d_h \times 2d_h}$. Let $\boldsymbol{W}_1, \boldsymbol{b}_1, \tilde{\boldsymbol{A}}_1, \tilde{\boldsymbol{W}}_2, \boldsymbol{b}_2, \tilde{\boldsymbol{A}}_2$ be given as in the proof of Proposition 25, for achieving an L_∞ error of $\tilde{\varepsilon}$, to be fixed later. Recall $\boldsymbol{z}_i = (\boldsymbol{x}_i^\top, \boldsymbol{\omega}_i^\top, \boldsymbol{\omega}_{t_1}^\top, \ldots, \boldsymbol{\omega}_{t_q}^\top)^\top$. In the following, we remove the zero columns of $\boldsymbol{W}_1$ corresponding to the $\boldsymbol{h}$ part of the input (see Lemma 22), which does not change the resulting function. Our construction can then be denoted by

$$\begin{array}{llllll} \boldsymbol{h}_i^{\rightarrow} & \xrightarrow{\boldsymbol{A}_{\text{id}}\sigma(\boldsymbol{W}_{\text{id}}\cdot)} & \boldsymbol{h}_i^{\rightarrow} & \xrightarrow{\boldsymbol{A}_{\text{id}}\sigma(\boldsymbol{W}_{\text{id}}\cdot)} & \boldsymbol{h}_i^{\rightarrow} & \searrow \\ \boldsymbol{h}_i^{\leftarrow} & \xrightarrow{\boldsymbol{A}_{\text{id}}\sigma(\boldsymbol{W}_{\text{id}}\cdot)} & \boldsymbol{h}_i^{\leftarrow} & \xrightarrow{\boldsymbol{A}_{\text{id}}\sigma(\boldsymbol{W}_{\text{id}}\cdot)} & \boldsymbol{h}_i^{\leftarrow} & \rightarrow \quad \boldsymbol{h}_i^{\rightarrow} + \boldsymbol{h}_i^{\leftarrow} + \boldsymbol{\chi}_2 \quad \xrightarrow{\boldsymbol{a}_g^\top \sigma(\boldsymbol{W}_g \cdot + \boldsymbol{b}_g)} \quad \hat{y}_{\text{RNN}}(\boldsymbol{p}; \boldsymbol{\Theta}_{\text{RNN}})_i \\ \boldsymbol{z}_i & \xrightarrow{\tilde{\boldsymbol{A}}_1\sigma(\boldsymbol{W}_1\cdot + \boldsymbol{b}_1)} & \boldsymbol{\chi}_1 & \xrightarrow{\tilde{\boldsymbol{A}}_2\sigma(\tilde{\boldsymbol{W}}_2\cdot + \boldsymbol{b}_2)} & \boldsymbol{\chi}_2 & \nearrow \end{array}$$

Note that the addition above can be implemented exactly by using the fact that $\sigma(z_1 + z_2 + z_3) - \sigma(-z_1 - z_2 - z_3) = z_1 + z_2 + z_3$. Specifically, the weights of this layer are given by

$$\boldsymbol{W}_{\text{add}} = \begin{pmatrix} \boldsymbol{v}_1^\top & \boldsymbol{v}_1^\top & \boldsymbol{v}_1^\top \\ -\boldsymbol{v}_1^\top & -\boldsymbol{v}_1^\top & -\boldsymbol{v}_1^\top \\ \vdots & \vdots & \vdots \\ \boldsymbol{v}_{d_h}^\top & \boldsymbol{v}_{d_h}^\top & \boldsymbol{v}_{d_h}^\top \\ -\boldsymbol{v}_{d_h}^\top & -\boldsymbol{v}_{d_h}^\top & -\boldsymbol{v}_{d_h}^\top \end{pmatrix}, \quad \boldsymbol{b}_{\text{add}} = \boldsymbol{0}_{2d_h}, \quad \boldsymbol{A}_{\text{add}} = \boldsymbol{A}_{\text{id}},$$

where $\boldsymbol{W}_{\text{add}} \in \mathbb{R}^{2d_h \times 3d_h}$, $\boldsymbol{b}_{\text{add}} \in \mathbb{R}^{2d_h}$, $\boldsymbol{A}_{\text{add}} \in \mathbb{R}^{d_h \times 2d_h}$.

Let $\boldsymbol{\Theta}_h^{\rightarrow}$ (and similarly $\boldsymbol{\Theta}_h^{\leftarrow}$) be given by Proposition 25 with corresponding error ε_h. Using the shorthand notation $\boldsymbol{x_t} = (\boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_q}) \in \mathbb{R}^{dq}$ and $\hat{\boldsymbol{x}}_{\boldsymbol{t}} = \boldsymbol{h}_i^{\rightarrow} + \boldsymbol{h}_i^{\leftarrow} + \boldsymbol{\chi}_2$, we have

$$\begin{aligned} \|\boldsymbol{h}_i^{\rightarrow} + \boldsymbol{h}_i^{\leftarrow} + \boldsymbol{\chi}_2 - \hat{\boldsymbol{x}}_{\boldsymbol{t}}\|_2 &\leq \left\| \boldsymbol{h}_i^{\rightarrow} - \sum_{j=1}^{i-1} \Psi(\boldsymbol{x}_j, \boldsymbol{t}, j) \right\|_2 + \left\| \boldsymbol{h}_i^{\leftarrow} - \sum_{j=N}^{i+1} \Psi(\boldsymbol{x}_j, \boldsymbol{t}, j) \right\|_2 + \|\boldsymbol{\chi}_2 - \Psi(\boldsymbol{x}_i, \boldsymbol{t}, i)\|_2 \\ &\leq \sqrt{qd}(2\varepsilon_h + \tilde{\varepsilon}), \end{aligned}$$

which holds for all input prompts $\boldsymbol{p}$ with $\|\boldsymbol{x}_j\|_2 \leq r_x$ for all $j \in [N]$. Finally, we have

$$\begin{aligned} \sup_{\|\boldsymbol{x}_j\|_2 \leq r_x, \forall j \in [N]} \left| g(\boldsymbol{x_t}) - \boldsymbol{a}_g^\top \sigma(\boldsymbol{W}_g \hat{\boldsymbol{x}}_{\boldsymbol{t}} + \boldsymbol{b}_g) \right| &\leq \sup_{\|\boldsymbol{x}_j\|_2 \leq r_x, \forall j \in [N]} \left| g(\boldsymbol{x_t}) - \boldsymbol{a}_g^\top \sigma(\boldsymbol{W}_g \boldsymbol{x_t} + \boldsymbol{b}_g) \right| \\ &\quad + \sup_{\|\boldsymbol{x}_j\|_2 \leq r_x, \forall j \in [N]} \left| \boldsymbol{a}_g^\top \sigma(\boldsymbol{W}_g \boldsymbol{x_t} + \boldsymbol{b}_g) - \boldsymbol{a}_g^\top \sigma(\boldsymbol{W}_g \hat{\boldsymbol{x}}_{\boldsymbol{t}} + \boldsymbol{b}_g) \right| \\ &\leq \sqrt{\varepsilon_{\text{2NN}}} + r_a r_w \sqrt{qd}(2\varepsilon_h + \tilde{\varepsilon}). \end{aligned}$$

Choosing $\varepsilon_h = \sqrt{\varepsilon_{\text{2NN}}}/(4\sqrt{qd} r_a r_w)$ and $\tilde{\varepsilon} = \sqrt{\varepsilon_{\text{2NN}}}/(2\sqrt{qd} r_a r_w)$, we obtain RNN weights that saitsfy $\|\text{vec}(\boldsymbol{\Theta}_{\text{RNN}})\|_2 \leq \text{poly}(r_x, D_e, r_a, r_w, \varepsilon_{\text{2NN}}^{-1})$, completing the proof. ■

C.4. Generalization Upper Bounds for RNNs

Recall the state transitions

$$\begin{aligned} \boldsymbol{h}_{j+1}^{\rightarrow} &= \Pi_{r_h}\big(\boldsymbol{h}_j^{\rightarrow} + f_h^{\rightarrow}(\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{z}_j; \boldsymbol{\Theta}_h^{\rightarrow})\big) \\ \boldsymbol{h}_{j-1}^{\leftarrow} &= \Pi_{r_h}\big(\boldsymbol{h}_j^{\leftarrow} + f^{\leftarrow}(\boldsymbol{h}^{\leftarrow}, \boldsymbol{z}_j; \boldsymbol{\Theta}^{\leftarrow})\big). \end{aligned}$$

We will use the notation $\boldsymbol{h}_j^{\rightarrow}(\boldsymbol{p}; \boldsymbol{\Theta}_h^{\rightarrow})$ and $\boldsymbol{h}_j^{\leftarrow}(\boldsymbol{p}; \boldsymbol{\Theta}_j^{\leftarrow})$ to highlight the dependence of the hidden states on the prompt $\boldsymbol{p}$ and parameters $\boldsymbol{\Theta}_h^{\rightarrow}$ and $\boldsymbol{\Theta}_h^{\leftarrow}$. We then define the prediction function as $F(\boldsymbol{p}; \boldsymbol{\Theta}_h^{\rightarrow}, \boldsymbol{\Theta}_h^{\leftarrow}, \boldsymbol{\Theta}_y)$ where

$$F(\boldsymbol{p}; \boldsymbol{\Theta}_h^{\rightarrow}, \boldsymbol{\Theta}_h^{\leftarrow}, \boldsymbol{\Theta}_y)_j = f_y(\boldsymbol{h}_j^{\rightarrow}(\boldsymbol{p}; \boldsymbol{\Theta}_h^{\rightarrow}), \boldsymbol{h}_j^{\leftarrow}(\boldsymbol{p}; \boldsymbol{\Theta}_h^{\leftarrow}), \boldsymbol{z}_j; \boldsymbol{\Theta}_y).$$

We can now define the function class

$$\mathcal{F}_{\text{RNN}} = \{\boldsymbol{p}, j \mapsto F(\boldsymbol{p}; \boldsymbol{\Theta}_h^{\rightarrow}, \boldsymbol{\Theta}_h^{\leftarrow}, \boldsymbol{\Theta}_y)_j \,:\, \boldsymbol{\Theta}_h^{\rightarrow}, \boldsymbol{\Theta}_h^{\leftarrow}, \boldsymbol{\Theta}_y \in \Theta_{\text{RNN}}\}.$$

We can then define our distance function by going over $\{\boldsymbol{p}, j \in S_n\}$,

$$d_\infty(F, \hat{F}) = \sup_{\boldsymbol{p}, j \in S_n} \left| F(\boldsymbol{p}; \boldsymbol{\Theta}_h^{\rightarrow}, \boldsymbol{\Theta}_h^{\leftarrow}, \boldsymbol{\Theta}_y)_j - F(\boldsymbol{p}; \hat{\boldsymbol{\Theta}}_h^{\rightarrow}, \hat{\boldsymbol{\Theta}}_h^{\leftarrow}, \boldsymbol{\Theta}_y)_j \right|.$$

We will further use the notation

$$f_y(\cdot; \boldsymbol{\Theta}_y) = \boldsymbol{W}_{L_y}^y \sigma\big(\boldsymbol{W}_{L_y-1}^y \ldots \sigma(\boldsymbol{W}_{L_1}^1(\cdot) + \boldsymbol{b}_1^y) \ldots + \boldsymbol{b}_{L_y-1}^y\big) \in \mathcal{F}_{\text{NN},L_y}^y,$$

and

$$f_h^{\rightarrow}(\cdot; \boldsymbol{\Theta}_h^{\rightarrow}) = \boldsymbol{W}_{L_h}^{\rightarrow} \sigma(\boldsymbol{W}_{L_h-1}^{\rightarrow} \ldots \sigma(\boldsymbol{W}_1^{\rightarrow}(\cdot) + \boldsymbol{b}_1^{\rightarrow}) \ldots + \boldsymbol{b}_{L_h-1}^{\rightarrow}) \in \mathcal{F}_{\text{NN},L_h}^{\rightarrow}.$$

We similarly define $\mathcal{F}_{\text{NN},L_h}^{\leftarrow}$. The covering number of $\mathcal{F}_{\text{RNN}}$ can be related to that of $\mathcal{F}_{\text{NN},L_y}^y$, $\mathcal{F}_{\text{NN},L_h}^{\rightarrow}$, and $\mathcal{F}_{\text{NN},L_y}^{\rightarrow}$, through the following lemma.

Lemma 28 *Suppose for every $\boldsymbol{\Theta}_h^{\rightarrow}, \boldsymbol{\Theta}_h^{\leftarrow}, \boldsymbol{\Theta}_y \in \Theta_{\text{RNN}}$ we have*

$$\left\|\boldsymbol{W}_{L_y}^y \ldots \boldsymbol{W}_1^y\right\|_{\text{op}} \leq C_W^y, \quad \left\|\boldsymbol{W}_{L_h}^{\rightarrow}\right\|_{\text{op}} \ldots \left\|\boldsymbol{W}_{1,h}^{\rightarrow}\right\|_{\text{op}} \leq \alpha_N, \quad \left\|\boldsymbol{W}_{L_h}^{\leftarrow}\right\|_{\text{op}} \ldots \left\|\boldsymbol{W}_{1,h}^{\leftarrow}\right\|_{\text{op}} \leq \alpha_N,$$

where $\alpha_N \leq N^{-1}$. Then,

$$\begin{aligned}\log \mathcal{C}(\mathcal{F}_{\text{RNN}}, d_\infty, \epsilon) \leq \log \mathcal{C}(\mathcal{F}_{\text{NN},L_y}^y, d_\infty, \epsilon/2) &+ \log \mathcal{C}\left(\mathcal{F}_{\text{NN},L_h}^{\rightarrow}, d_\infty, \frac{\epsilon}{4eC_w^y N}\right) \\ &+ \log \mathcal{C}\left(\mathcal{F}_{\text{NN},L_h}^{\leftarrow}, d_\infty, \frac{\epsilon}{4eC_w^y N}\right)\end{aligned}$$

Proof Throughout the proof, we will use the shorthand notation $\boldsymbol{h}_j^{\rightarrow} = \boldsymbol{h}_j^{\rightarrow}(\boldsymbol{p}; \boldsymbol{\Theta}_h^{\rightarrow})$ and $\hat{\boldsymbol{h}}_j^{\rightarrow} = \boldsymbol{h}_j^{\rightarrow}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}}_h^{\rightarrow})$, with similarly define $\boldsymbol{h}_j^{\leftarrow}$ and $\hat{\boldsymbol{h}}_j^{\leftarrow}$. We begin by observing

$$\sup_{\boldsymbol{p}, j \in S_n} \left| f_y(\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{h}_j^{\leftarrow}, \boldsymbol{z}_j; \boldsymbol{\Theta}_y) - f_y(\hat{\boldsymbol{h}}_j^{\rightarrow}, \hat{\boldsymbol{h}}_j^{\leftarrow}, \boldsymbol{z}_j; \hat{\boldsymbol{\Theta}}_y) \right| \leq \mathcal{E}_1 + \mathcal{E}_2$$

where

$$\begin{aligned}\mathcal{E}_1 &\coloneqq \sup_{\boldsymbol{p}, j \in S_n} \left| f_y(\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{h}_j^{\leftarrow}, \boldsymbol{z}_j; \boldsymbol{\Theta}_y) - f_y(\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{h}_j^{\leftarrow}, \boldsymbol{z}_j; \hat{\boldsymbol{\Theta}}_y) \right| \\ \mathcal{E}_2 &\coloneqq \sup_{\boldsymbol{p}, j \in S_n} \left| f_y(\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{h}_j^{\leftarrow}, \boldsymbol{z}_j; \hat{\boldsymbol{\Theta}}_y) - f_y(\hat{\boldsymbol{h}}_j^{\rightarrow}, \hat{\boldsymbol{h}}_j^{\leftarrow}, \boldsymbol{z}_j; \hat{\boldsymbol{\Theta}}_y) \right|.\end{aligned}$$

Then, we observe that $\mathcal{E}_1 = d_\infty(f_y(\cdot; \boldsymbol{\Theta}_y), f_y(\cdot; \hat{\boldsymbol{\Theta}}_y))$.Thus, we can ensure $\mathcal{E}_1 \leq \epsilon/2$ with a covering $\{\hat{\boldsymbol{\Theta}}_y\}$ of size $\mathcal{C}(\mathcal{F}_{\text{NN},L_y}^y, d_\infty, \epsilon/2)$. Hence, we move to $\mathcal{E}_2$.

Using the Lipschitzness of f_y, we obtain

$$\begin{aligned}\mathcal{E}_2 &\leq \left\|\boldsymbol{W}_{L_y}^y \ldots \boldsymbol{W}_1^y\right\|_{\text{op}} \left(\sup_{\boldsymbol{p}, j} \left\|\boldsymbol{h}_j^{\rightarrow} - \hat{\boldsymbol{h}_j^{\rightarrow}}\right\|_2 + \sup_{\boldsymbol{p}, j} \left\|\boldsymbol{h}_j^{\leftarrow} - \hat{\boldsymbol{h}}_j^{\leftarrow}\right\|_2 \right) \\ &\leq C_W^y \left(\sup_{\boldsymbol{p}, j} \left\|\boldsymbol{h}_j^{\rightarrow} - \hat{\boldsymbol{h}}_j^{\rightarrow}\right\|_2 + \sup_{\boldsymbol{p}, j} \left\|\boldsymbol{h}_j^{\leftarrow} - \hat{\boldsymbol{h}}_j^{\leftarrow}\right\|_2 \right).\end{aligned}$$

Further, by Lipschitzness of Π_{r_h}, we have

$$\sup_{\boldsymbol{p},j}\Big\|\boldsymbol{h}_j^{\rightarrow}-\hat{\boldsymbol{h}}_j^{\rightarrow}\Big\|_2 \le \sup_{\boldsymbol{p},j}\Big\|\boldsymbol{h}_{j-1}^{\rightarrow}-\hat{\boldsymbol{h}}_{j-1}^{\rightarrow}\Big\|_2 + \underbrace{\sup_{\boldsymbol{p},j}\Big\|f_h^{\rightarrow}(\boldsymbol{h}_{j-1}^{\rightarrow},\boldsymbol{z}_{j-1};\hat{\boldsymbol{\Theta}}_h^{\rightarrow})-f_h^{\rightarrow}(\hat{\boldsymbol{h}}_{j-1}^{\rightarrow},\boldsymbol{z}_{j-1};\hat{\boldsymbol{\Theta}}_h^{\rightarrow})\Big\|_2}_{=:\mathcal{E}_1^h}$$
$$+\underbrace{\sup_{\boldsymbol{p},j}\Big\|f_h^{\rightarrow}(\boldsymbol{h}_{j-1}^{\rightarrow},\boldsymbol{z}_{j-1};\boldsymbol{\Theta}_h^{\rightarrow})-f_h^{\rightarrow}(\boldsymbol{h}_{j-1}^{\rightarrow},\boldsymbol{z}_{j-1};\hat{\boldsymbol{\Theta}}_h^{\rightarrow})\Big\|_2}_{=:\mathcal{E}_2^h}.$$

By the Lipschitzness of $f_h^{\rightarrow}$, for the second term we have

$$\mathcal{E}_1^h \le \Big\|\hat{\boldsymbol{W}}_{L_h}^{\rightarrow}\dots\hat{\boldsymbol{W}}_{1,h}^{\rightarrow}\Big\|_{\mathrm{op}}\Big\|\boldsymbol{h}_{j-1}^{\rightarrow}-\hat{\boldsymbol{h}}_{j-1}^{\rightarrow}\Big\|_2 \le \alpha_N\Big\|\boldsymbol{h}_{j-1}^{\rightarrow}-\hat{\boldsymbol{h}}_{j-1}^{\rightarrow}\Big\|_2.$$

Moreover, we have $\mathcal{E}_2^h \le d_\infty(f_h^{\rightarrow}(\cdot;\boldsymbol{\Theta}_h^{\rightarrow}), f_h^{\rightarrow}(\cdot;\hat{\boldsymbol{\Theta}}_h^{\rightarrow}))$. Consequently, we obtain

$$\begin{aligned}
\sup_{\boldsymbol{p},j}\Big\|\boldsymbol{h}_j^{\rightarrow}-\hat{\boldsymbol{h}}_j^{\rightarrow}\Big\|_2 &\le (1+\alpha_N)\sup_{\boldsymbol{p},j}\Big\|\boldsymbol{h}_{j-1}^{\rightarrow}-\hat{\boldsymbol{h}}_{j-1}^{\rightarrow}\Big\|_2 + d_\infty(f_h^{\rightarrow}(\cdot;\boldsymbol{\Theta}_h^{\rightarrow}), f_h^{\rightarrow}(\cdot;\hat{\boldsymbol{\Theta}}_h^{\rightarrow}))\\
&\le \sum_{l=0}^{j-2}(1+\alpha_N)^l d_\infty(f_h^{\rightarrow}(\cdot;\boldsymbol{\Theta}_h^{\rightarrow}), f^{\rightarrow}(\cdot;\hat{\boldsymbol{\Theta}}_h^{\rightarrow}))\\
&\le \frac{(1+\alpha_N)^{j-1}-1}{\alpha_N} d_\infty(f_h^{\rightarrow}(\cdot;\boldsymbol{\Theta}_h^{\rightarrow}), f_h^{\rightarrow}(\cdot;\hat{\boldsymbol{\Theta}}_h^{\rightarrow}))\\
&\le eN d_\infty(f_h^{\rightarrow}(\cdot;\boldsymbol{\Theta}_h^{\rightarrow}), f_h^{\rightarrow}(\cdot;\hat{\boldsymbol{\Theta}}_h^{\rightarrow})).
\end{aligned}$$

We can similarly obtain an upper bound on $\sup_{\boldsymbol{p},j}\Big\|\boldsymbol{h}_j^{\leftarrow}-\hat{\boldsymbol{h}}_j^{\leftarrow}\Big\|_2$. Hence, we have

$$\mathcal{E}_2 \le eC_w^y N\Big\{d_\infty(f_h^{\rightarrow}(\cdot;\boldsymbol{\Theta}_h^{\rightarrow}), f_h^{\rightarrow}(\cdot;\hat{\boldsymbol{\Theta}}_h^{\rightarrow})) + d_\infty(f_h^{\leftarrow}(\cdot;\boldsymbol{\Theta}_h^{\leftarrow}), f_h^{\leftarrow}(\cdot;\hat{\boldsymbol{\Theta}}_h^{\leftarrow}))\Big\}.$$

Therefore, by constructing $\epsilon/(2eC_w^y N)$ coverings $\{\hat{\boldsymbol{\Theta}}_h^{\rightarrow}\}$ and $\{\hat{\boldsymbol{\Theta}}_h^{\leftarrow}\}$ which have sizes

$$\mathcal{C}(\mathcal{F}_{\mathrm{NN},L_h}^{\rightarrow},\epsilon/(4eC_w^y N)),\quad \text{and},\quad \mathcal{C}(\mathcal{F}_{\mathrm{NN},L_h}^{\leftarrow},\epsilon/(4eC_w^y N))$$

respectively, we complete the covering of $\mathcal{F}_{\mathtt{RNN}}$. ■

The next step is to bound the covering number of the class of feedforward networks, as performed by the following lemma.

Lemma 29 *Let*

$$\mathcal{F}_{\mathrm{NN},L} = \{\boldsymbol{x}\mapsto \boldsymbol{W}_L\sigma(\boldsymbol{W}_{L-1}\sigma(\dots\boldsymbol{W}_2(\sigma(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1)\dots+\boldsymbol{b}_{L-1}) : \boldsymbol{\Theta}_{\mathrm{NN}}\in\Theta_{\mathrm{NN}}\},$$

where $\boldsymbol{\Theta}_{\mathrm{NN}} = (\boldsymbol{W}_1,\boldsymbol{b}_1,\dots,\boldsymbol{W}_{L-1},\boldsymbol{b}_{L-1},\boldsymbol{W}_L)$ *and* $\mathrm{vec}(\boldsymbol{\Theta}_{\mathrm{NN}})\in\mathbb{R}^p$*. Further, define the distance function*

$$d_\infty(f,f') = \sup_{\|\boldsymbol{x}\|\le R}\big|f(\boldsymbol{x})-f'(\boldsymbol{x})\big|,\quad \forall f,f'\in\mathcal{F}_{\mathrm{NN},L}.$$

Suppose $\|\boldsymbol{W}_l\|_{\mathrm{F}}, \|\boldsymbol{b}_l\|_2 \le R$ *for all* l*. Then, for any absolute constant depth* $L=\mathcal{O}(1)$*, we have*

$$\log\mathcal{C}(\mathcal{F}_{\mathrm{NN},L}, d_\infty, \epsilon) \le p\log(1+\mathrm{poly}(R)/\epsilon).$$

Proof Let $\boldsymbol{x}_0 = \boldsymbol{x}$, $\boldsymbol{x}_l = \sigma(\boldsymbol{W}_l \boldsymbol{x}_{l-1} + \boldsymbol{b}_l)$ for $l \in [L-1]$, and $\boldsymbol{x}_L = \boldsymbol{W}_L \boldsymbol{x}_{L-1}$. Also let $(\hat{\boldsymbol{x}}_l)$ be the corresponding definitions under weights and biases $(\hat{\boldsymbol{W}}_l)$ and $(\hat{\boldsymbol{b}}_l)$. First, we remark that for $l \in [L-1]$,

$$\begin{aligned}
\|\boldsymbol{x}_l\|_2 &\le \|\boldsymbol{W}_l\|_{\mathrm{op}} \|\boldsymbol{x}_{l-1}\|_2 + \|\boldsymbol{b}_l\|_2 \qquad (23)\\
&\le \prod_{i=1}^{l} \|\boldsymbol{W}_i\|_{\mathrm{op}} \|\boldsymbol{x}_0\|_2 + \sum_{i=0}^{l-1} \|\boldsymbol{b}_{l-i-1}\|_2 \prod_{j=0}^{i} \|\boldsymbol{W}_{l-j}\|_{\mathrm{op}} + \|\boldsymbol{b}_l\|_2 \\
&\le \mathrm{poly}(R), \qquad (24)
\end{aligned}$$

where we used the fact that L is an absolute constant. Next, for $l \in [L-1]$, we have

$$\begin{aligned}
\|\boldsymbol{x}_l - \hat{\boldsymbol{x}}_l\|_2 &\le \left\|\boldsymbol{W}_l \boldsymbol{x}_{l-1} - \hat{\boldsymbol{W}}_l \hat{\boldsymbol{x}}_{l-1}\right\|_2 + \left\|\boldsymbol{b}_l - \hat{\boldsymbol{b}}_l\right\|_2 \\
&\le \|\boldsymbol{W}_l\|_{\mathrm{op}} \|\boldsymbol{x}_{l-1} - \hat{\boldsymbol{x}}_{l-1}\|_2 + \|\hat{\boldsymbol{x}}_{l-1}\|_2 \left\|\boldsymbol{W}_l - \hat{\boldsymbol{W}}_l\right\|_{\mathrm{op}} + \left\|\boldsymbol{b}_l - \hat{\boldsymbol{b}}_l\right\|_2 \\
&\le \mathrm{poly}(R) \Big\{ \|\boldsymbol{x}_{l-1} - \hat{\boldsymbol{x}}_{l-1}\|_2 + \left\|\boldsymbol{W}_l - \hat{\boldsymbol{W}}_l\right\|_{\mathrm{F}} + \left\|\boldsymbol{b}_l - \hat{\boldsymbol{b}}_l\right\|_2 \Big\}.
\end{aligned}$$

Once again, using the fact that L is an absolute constant and by expnaind the above inequality, we obtain

$$\|\boldsymbol{x}_l - \hat{\boldsymbol{x}}_l\|_2 \le \mathrm{poly}(R) \left\{ \sum_{i=1}^{l} \left\|\boldsymbol{W}_i - \hat{\boldsymbol{W}}_i\right\|_{\mathrm{F}} + \left\|\boldsymbol{b}_i - \hat{\boldsymbol{b}}_i\right\|_2 \right\}.$$

Finally, we have the bound

$$\begin{aligned}
\|\boldsymbol{x}_L - \hat{\boldsymbol{x}}_L\|_2 &\le \|\boldsymbol{W}_L\|_{\mathrm{op}} \|\boldsymbol{x}_{L-1} - \hat{\boldsymbol{x}}_{L-1}\|_2 + \|\hat{\boldsymbol{x}}_{L-1}\|_2 \left\|\boldsymbol{W}_L - \hat{\boldsymbol{W}}_L\right\|_{\mathrm{op}} \\
&\le \mathrm{poly}(R) \left\|\mathrm{vec}(\boldsymbol{\Theta}_{\mathrm{NN}}) - \mathrm{vec}(\hat{\boldsymbol{\Theta}}_{\mathrm{NN}})\right\|_2.
\end{aligned}$$

Consequently, we have

$$\begin{aligned}
\log \mathcal{C}(\mathcal{F}_{\mathrm{NN},L}, d_\infty, \epsilon) &\le \log \mathcal{C}(\{\boldsymbol{\Theta} \in \mathbb{R}^p \,:\, \|\boldsymbol{\Theta}\|_2 \le \mathrm{poly}(d,q)\}, \|\cdot\|_2, \epsilon/\mathrm{poly}(R)) \\
&\le p \log(1 + \mathrm{poly}(R)/\epsilon),
\end{aligned}$$

where the last inequality follows from Lemma 44. ■

Therefore, we immediately obtain the following bound on the covering number of $\mathcal{F}_{\mathrm{RNN}}$.

Corollary 30 *Suppose* $\Theta_{\mathrm{RNN}} \subseteq \{\boldsymbol{\Theta} \in \mathbb{R}^p : \|\mathrm{vec}(\boldsymbol{\Theta})\|_2 \le R\}$ *and* $\left\|\boldsymbol{z}_j^{(i)}\right\|_2 \le R$ *for all* $i \in [n]$ *and* $j \in [N]$. *Then,*

$$\log \mathcal{C}(\mathcal{F}_{\mathrm{RNN}}, d_\infty, \epsilon) \le p \log(1 + \mathrm{poly}(R) N/\epsilon).$$

We can now proceed with standard Rademacher complexity based arguments. Similar to the argument in Appendix B.2, we define a truncated version of the loss by considering the loss class

$$\mathcal{L}_\tau^{\mathrm{RNN}} = \{(\boldsymbol{p}, \boldsymbol{y}, j) \mapsto (f_{\mathrm{RNN}}(\boldsymbol{p})_j - y_j)^2 \wedge \tau \,:\, f_{\mathrm{RNN}} \in \mathcal{F}_{\mathrm{RNN}}\},$$

where the constant $\tau > 0$ will be chosen later. We then have the following bound on the empirical Rademacher complexity of $\mathcal{L}_\tau^{\mathrm{RNN}}$.

Lemma 31 *In the same setting as Corollary 30 and with $\tau \geq 1$, we have*

$$\hat{\mathfrak{R}}_n(\mathcal{L}_\tau^{\text{RNN}}) \leq \mathcal{O}\left(\tau\sqrt{\frac{p\log(RNn\tau)}{n}}\right).$$

Proof By a standard discretization bound for Rademacher complexity, for all $\epsilon > 0$ we have

$$\begin{aligned}\hat{\mathfrak{R}}_n(\mathcal{L}_\tau^{\text{RNN}}) &\leq \epsilon + \tau\sqrt{\frac{2\log \mathcal{C}(\mathcal{L}_\tau^{\text{RNN}}, d_\infty, \epsilon)}{n}} \\ &\leq \epsilon + \tau\sqrt{\frac{2\log \mathcal{C}(\mathcal{F}_{\text{RNN}}, d_\infty, \epsilon/(2\sqrt{\tau}))}{n}} \\ &\leq \epsilon + \tau\sqrt{\frac{2p\log(1+\text{poly}(R)N\sqrt{\tau}/\epsilon)}{n}},\end{aligned}$$

where the second inequality follows from Lipschitzness of $(\cdot)^2 \wedge \tau$. We conclude the proof by choosing $\epsilon = 1/\sqrt{n}$. ■

We can directly turn the above bound on the empirical Rademacher complexity into a bound on generalization gap.

Corollary 32 *Let $\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}\in\Theta_{\text{RNN}}} \hat{R}_n^{\text{RNN}}(\boldsymbol{\Theta})$. Suppose $\Theta_{\text{RNN}} \subseteq \{\boldsymbol{\Theta} \in \mathbb{R}^p : \|\text{vec}(\boldsymbol{\Theta})\|_2 \leq R\}$, and additionally $\sqrt{3C_x ed\log(nN) + q + 1} \leq R$. Then, for every $\delta > 0$, with probability at least $1 - \delta - (nN)^{-1/2}$ over the training set, we have*

$$R_\tau^{\text{RNN}}(\hat{\boldsymbol{\Theta}}) - \hat{R}_\tau^{\text{RNN}}(\hat{\boldsymbol{\Theta}}) \leq \mathcal{O}\left(\tau\sqrt{\frac{p\log(RNn\tau)}{n}} + \tau\sqrt{\frac{\log(1/\delta)}{n}}\right).$$

Proof We highlight that for the specified R, Lemma 11 guarantees $\left\|\boldsymbol{z}_j^{(i)}\right\|_2 \leq R$ for all $i \in [n]$ and $j \in [N]$ with probability at least $1 - (nN)^{-1/2}$. Standard Rademacher complexity generalization arguments applied to Lemma 31 complete the proof. ■

Note that $\hat{R}_\tau^{\text{RNN}}(\hat{\boldsymbol{\Theta}}) \leq \hat{R}_n^{\text{RNN}}(\hat{\boldsymbol{\Theta}})$ which is further controlled in the approximation section by Proposition 27. Therefore, the last step is to demonstrate that choosing $\tau = \text{poly}(d, q, \log n)$ suffices to achieve a desirable bound on $R^{\text{RNN}}(\hat{\boldsymbol{\Theta}})$ through $R_\tau^{\text{RNN}}(\hat{\boldsymbol{\Theta}})$.

Lemma 33 *Consider the setting of Corollary 32, and additionally assume $R \geq r_h$. Then, for some $\tau = \text{poly}(R, \log n)$, we have*

$$R^{\text{RNN}}(\hat{\boldsymbol{\Theta}}) - R_\tau^{\text{RNN}}(\hat{\boldsymbol{\Theta}}) \leq \sqrt{\frac{1}{n}}.$$

.

Proof The proof of this lemma proceeds similarly to the proof of Lemma 19. By defining

$$\Delta_y := \left|\hat{y}_{\text{RNN}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j - y_j\right|$$

and following the same steps (where we recall $j \sim \mathrm{Unif}([N])$), we obtain

$$\begin{aligned} R^{\mathrm{RNN}}(\hat{\boldsymbol{\Theta}}) &= \mathbb{E}\big[\Delta_y^2 \mathbb{1}[\Delta_y \le \sqrt{\tau}]\big] + \mathbb{E}\big[\Delta_y^2 \mathbb{1}[\Delta_y > \sqrt{\tau}]\big] \\ &\le R_\tau^{\mathrm{RNN}}(\hat{\boldsymbol{\Theta}}) + \mathbb{E}\big[\Delta_y^4\big]^{1/2} \mathbb{P}\big(\Delta_y \ge \sqrt{\tau}\big)^{1/2}, \end{aligned}$$

where

$$\mathbb{E}\big[\Delta_y^4\big]^{1/2} \le 2\,\mathbb{E}\big[y_j^4\big]^{1/2} + 2\,\mathbb{E}\Big[\hat{y}_{\mathrm{RNN}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j^4\Big]^{1/2}$$

and

$$\mathbb{P}\big(\Delta_y > \sqrt{\tau}\big) \le \mathbb{P}\left(|y_j| \ge \frac{\sqrt{\tau}}{2}\right) + \mathbb{P}\left(\left|\hat{y}_{\mathrm{RNN}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j\right| \ge \frac{\sqrt{\tau}}{2}\right)$$

From Assumption 1, we have $\mathbb{E}\Big[y_j^4\Big]^{1/2} \lesssim 1$ and $\mathbb{P}(|y_j| \ge \sqrt{\tau}/2) \le e^{-\Omega(\tau^{1/s})}$. For the prediction of the RNN, we have the following bound (see (24) for the derivation)

$$\left|\hat{y}_{\mathrm{RNN}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j\right| \le \prod_{l=1}^{L_y} \left\|\boldsymbol{W}_l^y\right\|_{\mathrm{op}} \left\|(\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{h}_j^{\leftarrow}, \boldsymbol{z}_j)\right\|_2 + \sum_{i=0}^{L_y-1} \left\|\boldsymbol{b}_{L_y-i-1}^y\right\|_2 \prod_{l=0}^{i} \left\|\boldsymbol{W}_{L_y-l}^y\right\|_{\mathrm{op}}.$$

As a result,

$$\left|\hat{y}_{\mathrm{RNN}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j\right| \le \mathrm{poly}(R)(1 + r_h + \|\boldsymbol{z}_j\|).$$

As a result, by the fact that $r_h \le R$ and Assumption 1, after taking an expectation, we immediately have

$$\mathbb{E}\Big[\hat{y}_{\mathrm{RNN}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})_j^4\Big]^{1/2} \le \mathrm{poly}(R).$$

On the other hand, from Lemma 11 (with $n = N = 1$), we obtain

$$\mathbb{P}\left(\left|\hat{y}_{\mathrm{RNN}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}})\right| \ge \frac{\sqrt{\tau}}{2}\right) \le e^{-\Omega(\tau/\,\mathrm{poly}(R))}$$

Therefore, for some $\tau = \mathrm{poly}(R, \log n)$ we can obtain the bound stated in the lemma. ■

We can summarize the above facts into the proof of Theorem 6.

Proof of Theorem 6. From the approximation bound of Proposition 27, we know that for some $R = \mathrm{poly}(d, q, r_a, r_w, \varepsilon_{2\mathrm{NN}}^{-1}, \log(nN))$ and the constraint set

$$\Theta_{\mathrm{RNN}} = \left\{\boldsymbol{\Theta} : \|\mathrm{vec}(\boldsymbol{\Theta})\|_2 \le R, \left\|\boldsymbol{W}_{L_h}^{\rightarrow}\right\|_{\mathrm{op}} \cdots \left\|\boldsymbol{W}_{1,h}^{\rightarrow}\right\|_{\mathrm{op}} \le \alpha_N, \left\|\boldsymbol{W}_{L_h}^{\leftarrow}\right\|_{\mathrm{op}} \cdots \left\|\boldsymbol{W}_{1,h}^{\leftarrow}\right\|_{\mathrm{op}} \le \alpha_N\right\}$$

with any $\alpha_N \le N^{-1}$, we have $\hat{R}^{\mathrm{RNN}}(\hat{\boldsymbol{\Theta}}) \lesssim \varepsilon_{2\mathrm{NN}}$. The proof is then completed by letting $r_h = \sqrt{q} r_x + \sqrt{\varepsilon_{2\mathrm{NN}}}/(r_a r_w)$, invoking the generalization bound of Corollary 32, and the bound on truncation error given in Lemma 33, with $R = \mathrm{poly}(d, q, r_a, r_w, \varepsilon_{2\mathrm{NN}}^{-1}, \log(nN))$.

■

C.5. RNN Lower Bound Formulation and Details

For our lower bound, we will consider a broad class of recurrent networks, without restricting to a specific form of parametrization. Specifically, we consider bidirectional RNNs chracterized by

$$\begin{aligned}
\boldsymbol{h}^{\rightarrow}_{i+1} &= \mathrm{proj}_{r_h}\big(f^{\rightarrow}_h(\boldsymbol{h}^{\rightarrow}_i, \boldsymbol{x}_i, \boldsymbol{t}_i, i)\big), \quad \forall\, i \in \{1, \dots, N-1\} \\
\boldsymbol{h}^{\leftarrow}_{i-1} &= \mathrm{proj}_{r_h}\big(f^{\leftarrow}_h(\boldsymbol{h}^{\leftarrow}_i, \boldsymbol{x}_i, \boldsymbol{t}_i, i)\big), \quad \forall\, i \in \{2, \dots, N\} \\
y_i &= f_y(\boldsymbol{U}^{\rightarrow}\boldsymbol{h}^{\rightarrow}_i, \boldsymbol{U}^{\leftarrow}\boldsymbol{h}^{\leftarrow}_i, \boldsymbol{x}_i, \boldsymbol{t}_i, i), \quad \forall i \in [N]
\end{aligned}$$

where $f_y : \mathbb{R}^{d_h} \times \mathbb{R}^{d_h} \times \mathbb{R}^d \times [N]^{q+1} \to \mathbb{R}$, $f^{\rightarrow}_h, f^{\leftarrow}_h : \mathbb{R}^{d_h} \times \mathbb{R}^d \times [N]^{q+1} \to \mathbb{R}^{d_h}$, $\boldsymbol{U}^{\rightarrow}, \boldsymbol{U}^{\leftarrow} \in \mathbb{R}^{d_h \times d_h}$, d_h is the width of the model, and $r_h > 0$ is some constant. Moreover, $\mathrm{proj}_{r_h} : \mathbb{R}^{d_h} \to \mathbb{R}^{d_h}$ is any mapping that guarantees $\big\|\mathrm{proj}_{r_h}(\cdot)\big\|_2 \le r_h$. As mentioned before, this operation mirrors the layer normalization to ensure that $\boldsymbol{h}_i$ remains stable. Further, we assume $f_y(\cdot, \boldsymbol{x}, \boldsymbol{t})$ is $\mathfrak{L}/r_h$-Lipschitz for all $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{t} \in [N]^q$. This formulation covers different variants of (bidirectional) RNNs used in practice such as LSTM and GRU, and includes the RNN formulation of Theorem 6 as a special case. Define $\boldsymbol{U} \coloneqq (\boldsymbol{U}^{\rightarrow}, \boldsymbol{U}^{\leftarrow}) \in \mathbb{R}^{d_h \times 2d_h}$ for conciseness. Note that in practice $f_y, f^{\rightarrow}_h, f^{\leftarrow}_h$ are determined by additional parameters. However, the only weight that we explicitly denote in this formulation is $\boldsymbol{U}$, since our lower bound will directly involve this projection, and we keep the rest of the parameters implicit for our representational lower bound.

Our technique for proving the RNN lower bound differs significantly from that of FFNs. In particular, we will control the representation cost of the qSTR model, i.e., a lower bound on the norm of $\boldsymbol{\Theta}_{\text{RNN}}$.

We will now present the RNN lower bound, with its proof deferred to Section C.6.

Proposition 34 *Consider the* 1STR *model where $\boldsymbol{x} \sim \mathcal{N}(0, \mathbf{I}_{Nd})$ with a linear link function, i.e. $y_j = \langle \boldsymbol{u}, \boldsymbol{x}_{t_j} \rangle$ for some $\boldsymbol{u} \in \mathbb{S}^{d-1}$. Further, t_i is drawn independently from the rest of the prompt and uniformly from $[N]$ for all $i \in [N]$. Then, there exists an absolute constant $c > 0$, such that*

$$\frac{1}{N}\,\mathbb{E}\Big[\|\boldsymbol{y} - \hat{\boldsymbol{y}}_{\text{RNN}}(\boldsymbol{p})\|^2\Big] \le c,$$

implies

$$d_h \ge \Omega\Big(\frac{N}{\log(1 + \mathfrak{L}^2\|\boldsymbol{U}\|^2_{\text{op}})}\Big), \quad \textit{and} \quad \|\boldsymbol{U}\|^2_{\text{op}} \ge \Omega\Big(\frac{N}{\mathfrak{L}^2 \log(1 + d_h)}\Big).$$

Remark 35 *Note that the unboundedness of Gaussian random variables is not an issue for approximation here, since $(g(\boldsymbol{x}_1), \dots, g(\boldsymbol{x}_N))$ is highly concentrated around $\mathbb{S}^{N-1}(\sqrt{N})$. In fact, one can directly assume $(g(\boldsymbol{x}_1), \dots, g(\boldsymbol{x}_N)) \sim \mathrm{Unif}(\mathbb{S}^{N-1}(\sqrt{N}))$ and derive a similar lower bound. The choice of Gaussian above is only made to simplify the presentation of the proof.*

The above proposition has two implications. First, it has a *computational* consequence, implying that any RNN representing the qSTR models requires a width that grows at least linearly with the context-length N. A similar lower bound in terms of bit complexity was derived in [24] using different tools. More importantly, the norm lower bound $\|\boldsymbol{U}\|_{\text{F}} \ge \tilde{\Omega}(\sqrt{N})$ has a *generalization* consequence, which we discuss below.

To translate the above representational cost result to a sample complexity lower bound, we now introduce the parametrization of the output function f_y. The exact parametrization of the transition

functions will be unimportant, and we will use the notation $f_h^{\rightarrow}(\boldsymbol{h}, \boldsymbol{x}, \boldsymbol{t}; \boldsymbol{\Theta}_h^{\rightarrow})$ to denote a general parameterized function (similarly with $f^{\leftarrow}$). We will assume f_y is given by a feedforward network,

$$f_y(\boldsymbol{U}^{\rightarrow}\boldsymbol{h}^{\rightarrow}, \boldsymbol{U}^{\leftarrow}\boldsymbol{h}^{\leftarrow}, \boldsymbol{x}, \boldsymbol{t}; \boldsymbol{\Theta}_y) = \boldsymbol{W}_{L_y}\sigma\big(\dots\sigma(\boldsymbol{W}_2\sigma(\boldsymbol{U}\boldsymbol{h} + \boldsymbol{W}_y\boldsymbol{z} + \boldsymbol{b}_y) + \boldsymbol{b}_2)\dots\big),$$

where $\boldsymbol{h} = (\boldsymbol{h}^{\rightarrow}, \boldsymbol{h}^{\leftarrow}) \in \mathbb{R}^{2d_h}$, $\boldsymbol{z} = (\boldsymbol{x}_i, f_E(\boldsymbol{t}_i, i)) \in \mathbb{R}^{d+d_E}$. Here, $f_E(\boldsymbol{t}_i, i)$ is an arbitrary encoding function with arbitrary dimension d_E. Then $\boldsymbol{\Theta}_y = (\boldsymbol{U}, \boldsymbol{W}_y, \boldsymbol{b}_y, \boldsymbol{W}_2, \boldsymbol{b}_2, \dots, \boldsymbol{W}_{L_y})$, and $\boldsymbol{\Theta}_{\text{RNN}} = (\boldsymbol{U}, \boldsymbol{\Theta}_y, \boldsymbol{\Theta}_h^{\rightarrow}, \boldsymbol{\Theta}_h^{\leftarrow})$. Note that thanks to the homogeneity of ReLU, we can always reparameterize the network by taking $\bar{\boldsymbol{h}} = \boldsymbol{h}/r_h$, $\bar{\boldsymbol{W}}_y = \boldsymbol{W}_y/r_h$, $\bar{\boldsymbol{b}}_y = \boldsymbol{b}_y/r_h$, and $\bar{\boldsymbol{W}}_2 = \boldsymbol{W}_2/r_h$ without changing the prediction function. Thus, in Theorem 7, we take $r_h = 1$ without losing the expressive power of the network. We now state the rigorous version of Theorem 7.

Theorem 36 *Consider the* 1STR *model of Proposition 34. Suppose the size of the hidden state, the depth of the prediction function, and the weight norm respectively satisfy $d_h \le e^{N^c}$, $2 \le L_y \le C$, and $\|\text{vec}(\boldsymbol{\Theta}_{\text{RNN}})\|_2 \le e^{N^c/L_y}$ for some absolute constants $c < 1$ and $C \ge 2$, and recall we set $r_h = 1$ due to homogeneity of the network. Let $\hat{\boldsymbol{\Theta}}_\varepsilon$ be the min-norm ε-ERM of $\hat{R}_n^{\text{RNN}}$, defined in (4). Then, there exist absolute constants $c_1, c_2, c_3 > 0$ such that if $n \le \mathcal{O}(N^{c_1})$, for any $\varepsilon \ge 0$, with probability at least c_2 over the training set,*

$$\frac{1}{N}\mathbb{E}\bigg[\Big\|\hat{\boldsymbol{y}}_{\text{RNN}}(\boldsymbol{p}; \hat{\boldsymbol{\Theta}}_{n,\varepsilon}) - \boldsymbol{y}\Big\|_2^2\bigg] \ge c_3.$$

C.6. Proof of Proposition 34

The crux of the proof of Proposition 34 is to show the following position, which provides a lower bound on the prediction error at any fixed position in the prompt.

Proposition 37 *Consider the same setting as in Proposition 34. There exists an absolute constant $c > 0$, such that for any fixed $j \in [N]$, if*

$$\mathbb{E}\big[(\hat{y}_{\text{RNN}}(\boldsymbol{p})_j - y_j)^2\big] \le c,$$

then

$$d_h \ge \Omega\Big(\frac{N}{\log(1 + \mathfrak{L}^2\|\boldsymbol{U}\|_{\text{op}}^2)}\Big), \quad \textit{and} \quad \|\boldsymbol{U}\|_{\text{op}}^2 \ge \Omega\Big(\frac{N}{\mathfrak{L}^2\log(1+d_h)}\Big).$$

We shortly remark that the statement of Proposition 34 directly follows from that of Proposition 37.

Proof of Proposition 34. Let c be the constant given by Proposition 37. Suppose that

$$\frac{1}{N}\mathbb{E}\Big[\|\hat{\boldsymbol{y}}_{\text{RNN}}(\boldsymbol{p}) - \boldsymbol{y}\|_2^2\Big] \le c.$$

Then,

$$\min_{j\in[N]} \mathbb{E}\big[(\hat{y}_{\text{RNN}}(\boldsymbol{p})_j - y_j)^2\big] \le \frac{1}{N}\sum_{j=1}^{N}\mathbb{E}\big[(\hat{y}_{\text{RNN}}(\boldsymbol{p})_j - y_j)^2\big] \le c.$$

As a result, there exists some $j \in [N]$ such that $\mathbb{E}\big[(\hat{y}_{\text{RNN}}(\boldsymbol{p})_j - y_j)^2\big] \le c$. We can then invoke Proposition 37 to obtain lower bounds on d_h and $\|\boldsymbol{U}\|_{\text{op}}$, completing the proof of Proposition 34. ■

We now present the proof of Proposition 37.

Proof of Proposition 37. Let $\boldsymbol{h}_j = (\boldsymbol{U}^{\rightarrow}\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{U}^{\leftarrow}\boldsymbol{h}_j^{\leftarrow}) \in \mathbb{R}^{2d_h}$, and define

$$\Phi(\boldsymbol{h}_j) \coloneqq \Big(f_y(\boldsymbol{h}_j, \boldsymbol{x}_j, (1), j), \ldots, f_y(\boldsymbol{h}_j, \boldsymbol{x}_j, (j-1), j), f_y(\boldsymbol{h}_j, \boldsymbol{x}_j, (j+1), j), \ldots, f_y(\boldsymbol{h}_j, \boldsymbol{x}_j, (N), j)\Big)^\top \in \mathbb{R}^{N-1}.$$

In other words, $\Phi : \mathbb{R}^{2d_h} \to \mathbb{R}^{N-1}$ captures all possible outcomes of $\hat{y}_{\text{RNN}}(\boldsymbol{p})_j$ depending on the value of t_j (excluding the case where $t_j = j$). Ideally, we must have $f_y(\boldsymbol{h}_j, \boldsymbol{x}_j, (k), j) \approx g(\boldsymbol{x}_k)$.

Let $\boldsymbol{p}^{(1)}, \ldots, \boldsymbol{p}^{(P)}$ be an i.i.d. sequence of prompts, then modify them to share the jth input token, i.e. $\boldsymbol{x}_j^{(i)} = \boldsymbol{x}_j^{(1)}$ for all $i \in [P]$, with P to be determined later. Note that by our assumption on prompt distribution, this operation does not change the marginal distribution of each $\boldsymbol{p}^{(i)}$. Similarly, define

$$\boldsymbol{g}^{(i)} \coloneqq (\boldsymbol{g}(\boldsymbol{x}_1^{(i)}), \ldots, \boldsymbol{g}(\boldsymbol{x}_{j-1}^{(i)}), \boldsymbol{g}(\boldsymbol{x}_{j+1}^{(i)}), \ldots, \boldsymbol{g}(\boldsymbol{x}_N^{(i)}))^\top \in \mathbb{R}^{N-1}$$

for each prompt. We also let $\boldsymbol{h}^{(i)}{}_j^{\rightarrow}, \boldsymbol{h}^{(i)}{}_j^{\leftarrow}$ be the corresponding hidden states obtained from passing these prompts through the RNN, and define $\boldsymbol{h}_j^{(i)}$ using them. Note that $\boldsymbol{g}^{(1)}, \ldots, \boldsymbol{g}^{(P)}$ is an i.i.d. sequence of vectors drawn from $\mathcal{N}(0, \mathbf{I}_{N-1})$.

We now define two events E_1 and E_2, where

$$E_1 = \Big\{\forall\, i \neq k, \quad \Big\|\boldsymbol{g}^{(i)} - \boldsymbol{g}^{(k)}\Big\|_2 \geq \varepsilon_g\sqrt{N-1}\Big\},$$

and

$$E_2 = \left\{\sum_{i=1}^{P} \mathbb{1}\left[\Big\|\Phi(\boldsymbol{h}_j^{(i)}) - \boldsymbol{g}^{(i)}\Big\|_2 \geq \frac{\varepsilon\sqrt{N}}{\delta}\right] \leq 2\delta^2 P\right\},$$

where $\delta \in (0, 1)$ will be chosen later. In other words, E_1 is the event in which $\boldsymbol{g}^{(i)}$ are "packed" in the space, while E_2 is the event where the RNN will be "wrong" at position j on at most $2\delta^2$ fraction of the prompts. We will now attempt to lower bound $\mathbb{P}(E_1 \cap E_2)$.

Note that $\boldsymbol{g}^{(i)} - \boldsymbol{g}^{(k)} \overset{(d)}{=} \sqrt{2}\boldsymbol{g}$ where $\boldsymbol{g} \sim \mathcal{N}(0, \mathbf{I}_{N-1})$. By a union bound we have

$$\begin{aligned}
\mathbb{P}\big(E_1^C\big) &\leq \sum_{i \neq k} \mathbb{P}\Big(\Big\|\boldsymbol{g}^{(i)} - \boldsymbol{g}^{(k)}\Big\|_2 \leq \varepsilon_g\sqrt{N-1}\Big) \\
&\leq P^2\mathbb{P}\Big(\sqrt{2}\|\boldsymbol{g}\|_2 \leq \varepsilon_g\sqrt{N-1}\Big) \\
&\leq P^2\mathbb{P}\left(\|\boldsymbol{g}\|_2 - \mathbb{E}[\|\boldsymbol{g}\|_2] \leq \big(\frac{\varepsilon_g}{\sqrt{2}} - c\big)\sqrt{N-1}\right) \\
&\leq P^2 e^{-(c-\varepsilon_g/\sqrt{2})^2(N-1)/2},
\end{aligned}$$

for all $\varepsilon_g \leq c\sqrt{2}$, where $c > 0$ is an absolute constant such that $c\sqrt{N-1} \leq \mathbb{E}[\|\boldsymbol{g}\|]$, and the last inequality holds by subGaussianity of the norm of a standard Gaussian random vector. From here on, we will choose $\varepsilon_g = c/\sqrt{2}$ (and simply denote $\varepsilon_g \asymp 1$), which implies $\mathbb{P}\big(E_1^C\big) \leq P^2 e^{-c^2(N-1)/8}$.

To lower bound $\mathbb{P}(E_2)$, consider a random prompt-label pair $\boldsymbol{p}, \boldsymbol{y}$ and the corresponding $\boldsymbol{g}$. Note that in the prompt $\boldsymbol{p}$, the index t_j is drawn independently of the rest of $\boldsymbol{p}$, and has a uniform distribution in $[N]$. Let $\boldsymbol{p}[t_j \mapsto k]$ denote a modification of $\boldsymbol{p}$ where we set t_j equal to k, and let

$\boldsymbol{y}[t_j \mapsto k]$ be the labels corresponding to this modified prompt. We then have

$$\begin{aligned}\frac{1}{N}\|\Phi(\boldsymbol{h}_j) - \boldsymbol{g}\|_2^2 &= \frac{1}{N}\sum_{k\neq j}\left(\hat{y}_{\text{RNN}}(\boldsymbol{p}[t_j \mapsto k])_j - g(\boldsymbol{x}_k)\right)^2 \\ &\leq \frac{1}{N}\sum_{k=1}^{N}\left(\hat{y}_{\text{RNN}}(\boldsymbol{p}[t_j \mapsto k])_j - y(\boldsymbol{p}[t_j \mapsto k])_j\right)^2 \\ &= \mathbb{E}_{t_j}\left[(\hat{y}_{\text{RNN}}(\boldsymbol{p})_j - y_j)^2\right]\end{aligned}$$

As a result, via a Markov inequality, we obtain

$$\begin{aligned}\mathbb{P}\left(\frac{1}{N}\|\Phi(\boldsymbol{h}_j) - \boldsymbol{g}\|_2^2 \geq \frac{\varepsilon^2}{\delta^2}\right) &= \mathbb{P}\left(\mathbb{E}_{t_j}\left[(\hat{y}_{\text{RNN}}(\boldsymbol{p})_j - y_j)^2\right] \geq \frac{\varepsilon^2}{\delta^2}\right) \\ &\leq \frac{\delta^2\,\mathbb{E}\left[(\hat{y}_{\text{RNN}}(\boldsymbol{p})_j - y_j)^2\right]}{\varepsilon^2} \\ &\leq \delta^2.\end{aligned}$$

Going back to our lower bound on $\mathbb{P}(E_2)$, define the Bernoulli random variable

$$z^{(i)} = \mathbb{1}\left[\left\|\Phi(\boldsymbol{h}_j^{(i)}) - \boldsymbol{g}^{(i)}\right\|_2 \geq \frac{\varepsilon\sqrt{N}}{\delta}\right].$$

Note that $(z^{(i)})$ are i.i.d. since $\boldsymbol{h}_j^{(i)}$ and $\boldsymbol{g}^{(i)}$ do not depend on $\boldsymbol{x}_j$. Then, by Hoeffding's inequality,

$$\mathbb{P}\big(E_2^C\big) = \mathbb{P}\left(\sum_{j=1}^{P} z^{(i)} \geq 2\delta^2 P\right) \leq e^{-2P\delta^4}.$$

We now have our desired lower bound on $\mathbb{P}(E_1 \cap E_2)$, given by

$$\mathbb{P}(E_1 \cap E_2) \geq 1 - \mathbb{P}\big(E_1^C\big) - \mathbb{P}\big(E_2^C\big) \geq 1 - e^{-2P\delta^4} - P^2 e^{-c^2(N-1)/8}.$$

Suppose $\delta \geq e^{-c'N}$ for some absolute constant $c' > 0$. Then, choosing $P = \lfloor e^{c''N} \rfloor$ for some absolute constant $c'' > 0$ would ensure $\mathbb{P}(E_1 \cap E_2) > 0$, and allows us to look at this intersection.

Let $\mathcal{I} = \{i : z^{(i)} = 0\}$. On E_1, and for $i, k \in \mathcal{I}$ with $i \neq k$ we have

$$\begin{aligned}\left\|\Phi(\boldsymbol{h}_j^{(i)}) - \Phi(\boldsymbol{h}_j^{(k)})\right\|_2 &\geq \left\|\boldsymbol{g}^{(i)} - \boldsymbol{g}^{(k)}\right\|_2 - \left\|\Phi(\boldsymbol{h}_j^{(i)}) - \boldsymbol{g}^{(i)}\right\|_2 - \left\|\Phi(\boldsymbol{h}_j^{(k)}) - \boldsymbol{g}^{(k)}\right\|_2 \\ &\geq \varepsilon_g\sqrt{N-1} - \frac{2\varepsilon\sqrt{N}}{\delta} =: \mathfrak{L}\sqrt{N}\varepsilon_h.\end{aligned}$$

Note that from the Lipschitzness of f_y, we have $\left\|\Phi(\boldsymbol{h}_j^{(i)}) - \Phi(\boldsymbol{h}_j^{(k)})\right\|_2 \leq \frac{\mathfrak{L}\sqrt{N}}{r_h}\left\|\boldsymbol{h}_j^{(i)} - \boldsymbol{h}_j^{(k)}\right\|_2$. As a result, the set $\left\{\boldsymbol{h}_j^{(i)} : i \in \mathcal{I}\right\}$ is an $r_h\varepsilon_h$-packing for $\{\boldsymbol{h} : \|\boldsymbol{h}\|_2 \leq \sqrt{2}\|\boldsymbol{U}\|_{\text{op}} r_h\}$. Using Lemma 44, the log packing number can be bounded by

$$\log\mathcal{I} \leq \left\{d_h \log\left(1 + \frac{2\sqrt{2}\|\boldsymbol{U}\|_{\text{op}}}{\varepsilon_h}\right)\right\} \wedge \left\{\frac{2\|\boldsymbol{U}\|_{\text{op}}^2}{\varepsilon_h^2}\left(1 + \log\left(1 + \frac{M\varepsilon_h^2}{2\|\boldsymbol{U}\|_{\text{op}}^2}\right)\right)\right\}.$$

On $E_1 \cap E_2$, we have $\mathcal{I} \geq (1-2\delta^2)P \geq (1-2\delta^2)e^{cN}$ for some absolute constant $c > 0$. Therefore,

$$\frac{\log(1-2\delta^2) + cN}{\log(1 + 2\sqrt{2}\|\boldsymbol{U}\|_{\mathrm{op}}/\varepsilon_h)} \leq d_h,$$

and

$$\frac{\varepsilon_h^2\big(\log(1-2\delta^2) + cN\big)}{2 + 2\log(1 + d_h\varepsilon_h^2/(2\|\boldsymbol{U}\|_{\mathrm{op}}^2))} \leq \|\boldsymbol{U}\|_{\mathrm{op}}^2.$$

Choosing $\delta = 1/2$ and recalling $\varepsilon_g \asymp 1$, we obtain $\varepsilon_h \gtrsim (1 - C\varepsilon)/\mathfrak{L}$ for some absolute constant $C > 0$, which concludes the proof. ■

C.7. Proof of Theorem 7

We first provide an estimate for the capacity of two-layer feedforward networks to interpolate n samples.

Lemma 38 *Suppose* $\{\boldsymbol{x}^{(i)}\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I}_d)$ *and let* $y^{(i)} = \langle \boldsymbol{u}, \boldsymbol{x}_{t_i} \rangle$ *for arbitrary* $t_i \in [N]$ *and* $\boldsymbol{u} \in \mathbb{S}^{d-1}$. *Then, there exists an absolute constant* $c > 0$ *such that for all* $m \geq n$ *and with probability at least* c, *there exist data dependent weights* $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^m$ *and* $\boldsymbol{W} \in \mathbb{R}^{m \times d}$, *such that*

$$\boldsymbol{a}^\top \sigma(\boldsymbol{W}\boldsymbol{x}^{(i)} + \boldsymbol{b}) = y^{(i)}, \quad \forall\, i \in [n]$$

and

$$\|\boldsymbol{a}\|_2^2 + \|\boldsymbol{W}\|_{\mathrm{F}}^2 + \|\boldsymbol{b}\|_2^2 \leq \mathcal{O}(n^3).$$

Proof The proof of Lemma 38 is an immediate consequence of two lemmas.

1. Lemma 39 shows that the inputs $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ can be projected to sufficiently separated scalar values with a unit vector $\boldsymbol{v}$.
2. Lemma 40 perfectly fits n univariate samples using a two-layer ReLU neural network. When invoking this lemma, we use $\|\boldsymbol{z}\|_2 = \mathcal{O}(\sqrt{n})$ and $\epsilon = \Omega(1/n^2)$ as given by Lemma 39.

The only missing piece is to upper bound $\|\boldsymbol{y}\|_2$ appearing in the final bound of Lemma 40. To that end, we apply the following Markov inequality,

$$\mathbb{P}\Big(\|\boldsymbol{y}\|_2^2 \geq 6n\Big) \leq \frac{\mathbb{E}\Big[\|\boldsymbol{y}\|_2^2\Big]}{6n} \leq \frac{1}{6}.$$

As the statement of Lemma 39 holds with probability at least $\frac{1}{3}$, this suggests that the statement of Lemma 38 holds with probability at least $\frac{1}{6}$, concluding the proof. ■

Lemma 39 *Suppose* $\{\boldsymbol{x}^{(i)}\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I}_d)$. *Then, with probability at least* $1/3$, *there exists some* $\boldsymbol{v} \in \mathbb{S}^{d-1}$ *(dependent on* $\{\boldsymbol{x}^{(i)}\}$*) such that for all* $i \neq j$,

$$\Big|\boldsymbol{v}^\top \boldsymbol{x}^{(i)} - \boldsymbol{v}^\top \boldsymbol{x}^{(j)}\Big| = \Omega\left(\frac{1}{n^2}\right). \tag{25}$$

and $\sum_{i=1}^n (\boldsymbol{v}^\top \boldsymbol{x}^{(i)})^2 = \mathcal{O}(n)$.

Proof The proof follows the probabilistic method. Sample $\boldsymbol{v} \sim \text{Unif}(\mathbb{S}^{d-1})$ independent of $\{\boldsymbol{x}^{(i)}\}$. For each $i \neq j$, let

$$a_{i,j} = \boldsymbol{u}^\top(\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)})$$

and note that $a_{i,j} \,|\, \boldsymbol{v} \sim \mathcal{N}(0, 2)$. We apply basic Gaussian anti-concentration to place a lower bound on the probability of any $a_{i,j}$ being close to zero,

$$\mathbb{P}(\exists i, j \text{ s.t. } |a_{i,j}| \leq \epsilon) \leq \sum_{i \neq j} \mathbb{P}(|a_{i,j}| \leq \epsilon) = \sum_{i \neq j} \mathbb{E}[\mathbb{P}(|a_{i,j}| \leq \epsilon \,|\, \boldsymbol{v})] \leq \frac{n^2\epsilon}{\sqrt{\pi}} \leq \frac{1}{3},$$

where the last inequality follows by taking $\epsilon = \sqrt{\pi}/(3n^2)$. Furthermore,

$$\mathbb{P}\left(\sum_{i=1}^{n} (\boldsymbol{v}^\top \boldsymbol{x}^{(i)})^2 \geq 3n\right) \leq \frac{\sum_{i=1}^{n} \mathbb{E}\left[(\boldsymbol{v}^\top \boldsymbol{x}^{(i)})^2\right]}{3n} = \frac{1}{3},$$

by Markov's inequality. Combining the two events completes the proof. ■

Lemma 40 *Consider some $\boldsymbol{z} = (z^{(1)}, \ldots, z^{(n)})^\top \in \mathbb{R}^n$ and $\boldsymbol{y} = (y^{(1)}, \ldots, y^{(n)})^\top \in \mathbb{R}^n$, such that $\left|z^{(i)} - z^{(j)}\right| \geq \epsilon$ for all $i \neq j$. For simplicity, assume $\epsilon \leq 1$. Then, there exists a two-layer ReLU neural network*

$$g(t) = \sum_{j=1}^{m} a_j \sigma(w_j t + b_j)$$

that satisfies $g(z^{(i)}) = y^{(i)}$ for all $i \in [n]$, $m = n$, and

$$\|\boldsymbol{a}\|_2^2 + \|\boldsymbol{w}\|_2^2 + \|\boldsymbol{b}\|_2^2 = \mathcal{O}\left(\frac{\|\boldsymbol{y}\|_2 \sqrt{n + \|\boldsymbol{z}\|_2^2}}{\epsilon}\right). \tag{26}$$

Proof Without loss of generality, we assume that $z^{(1)} \leq \cdots \leq z^{(n)}$. Then, we define the neural network g as follows:

$$g(t) = \sum_{i=1}^{n} a_i' \sigma(w_i' t - b_i') = y^{(1)} \sigma(t - z^{(1)} + 1) + \left(\frac{y^{(2)} - y^{(1)}}{z^{(2)} - z^{(1)}} - y^{(1)}\right) \sigma(t - z^{(1)}) + \sum_{i=3}^{n} \left(\frac{y^{(i)} - y^{(i-1)}}{z^{(i)} - z^{(i-1)}} - \frac{y^{(i-1)} - y^{(i-2)}}{z^{(i-1)} - z^{(i-2)}}\right) \sigma(t - z^{(i-1)}).$$

One can verify by induction that $g(z^{(i)}) = y^{(i)}$ for every i by noting that the slope of g is

$$(y^{(i)} - y^{(i-1)})/(z^{(i)} - z^{(i-1)})$$

between $(z^{(i-1)}, y^{(i-1)})$ and $(z^{(i)}, y^{(i)})$. From the above, we have $w_i' = 1$, $\left\|\boldsymbol{b}'\right\|_2^2 \lesssim \|\boldsymbol{z}\|_2^2 + 1$, and $\|\boldsymbol{a}'\|_2^2 \lesssim \|\boldsymbol{y}\|_2^2/\epsilon^2$. For $\alpha = \left((\|\boldsymbol{z}\|_2^2 + n)\epsilon^2/\|\boldsymbol{y}\|_2^2\right)^{1/4}$, let $\boldsymbol{u} = \alpha \boldsymbol{u}'$, $\boldsymbol{w} = \boldsymbol{w}'/\alpha$, and $\boldsymbol{b} = \boldsymbol{b}'/\alpha$. By homogeneity, the neural network with weights $(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{b})$ has identical outputs to that of $(\boldsymbol{u}', \boldsymbol{w}', \boldsymbol{b}')$ and satisfies (26), completing the proof. ■

We are now ready to present the proof of the sample complexity lower bound for RNNs.

Proof of Theorem 7. First, consider the case where $d_h < n$. Note that as a function of $\boldsymbol{U}\boldsymbol{h} = (\boldsymbol{U}^{\rightarrow}\boldsymbol{h}^{\rightarrow}, \boldsymbol{U}^{\leftarrow}\boldsymbol{h}^{\leftarrow})$, f_y is $\mathfrak{L}$-Lipschitz with

$$\mathfrak{L} = \left\|\boldsymbol{W}_{L_y}\right\|_{\mathrm{op}} \left\|\boldsymbol{W}_{L_y-1}\right\|_{\mathrm{op}} \dots \|\boldsymbol{W}_2\|_{\mathrm{op}}.$$

Using the AM-GM inequality,

$$\left(\mathfrak{L}^2 \|\boldsymbol{U}\|_{\mathrm{op}}^2\right)^{1/L_y} \leq \frac{1}{L_y} \|\mathrm{vec}(\boldsymbol{\Theta})\|_2^2 \leq e^{N^c/L_y}.$$

As a result, we have $\mathfrak{L}\|\boldsymbol{U}\|_{\mathrm{op}} \leq e^{N^c/2}$. By invoking Proposition 25, to obtain population risk less than some absolute constant $c_3 > 0$, we need

$$d_h \geq \Omega\left(\frac{N}{\log(1 + \mathfrak{L}^2\|\boldsymbol{U}\|_{\mathrm{op}}^2)}\right) \geq \Omega(N^{1-c}).$$

This implies $n \geq d_h \geq \Omega(N^{1-c})$. By taking c_1 in the theorem statement to be less than $1 - c$, we obtain a contradiction. Therefore, we must have either a population risk at least c_3 or $d_h \geq n$.

Suppose now that $d_h \geq n$. We show that with constant probability, we can construct an RNN that interpolates the n training samples with norm independent of n. We simply let $\boldsymbol{\Theta}_h^{\rightarrow} = \boldsymbol{0}$, $\boldsymbol{\Theta}_h^{\leftarrow} = \boldsymbol{0}$, $\boldsymbol{U} = \boldsymbol{0}$, and describe the construction of $\boldsymbol{W}_{L_y}, \dots, \boldsymbol{W}_2, \boldsymbol{W}_y$, and $(\boldsymbol{b}_l)$ in the following. Using the construction of Lemma 38, we can let

$$\boldsymbol{W}_y = \begin{pmatrix} \boldsymbol{W} & \boldsymbol{0}_{n\times d_E} \\ \boldsymbol{0}_{(m-n)\times d} & \boldsymbol{0}_{(m-n)\times d_E} \end{pmatrix}, \quad \boldsymbol{b}_1 = \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{0}_{m-n} \end{pmatrix}, \quad \boldsymbol{W}_2 = \begin{pmatrix} \boldsymbol{a}^\top & \boldsymbol{0}_{m-n}^\top \\ -\boldsymbol{a}^\top & \boldsymbol{0}_{m-n}^\top \\ \boldsymbol{0}_{(m-2)\times n} & \boldsymbol{0}_{(m-2)\times(m-n)} \end{pmatrix},$$

where $\boldsymbol{W} \in \mathbb{R}^{n\times d}$, and $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ are given by Lemma 38. Then,

$$\boldsymbol{W}_2^\top \sigma(\boldsymbol{W}_y \boldsymbol{x}_{j^{(i)}}^{(i)} + \boldsymbol{b}_y) = (y_{j^{(i)}}^{(i)}, -y_{j^{(i)}}^{(i)}, 0, \dots, 0)^\top.$$

For $(\boldsymbol{W}_l)_{l=3}^{L_y-1}$, we let $(W_l)_{11} = (W_l)_{22} = 1$, and choose the rest of the coordinates of $\boldsymbol{W}_l$ to be zero. Therefore, the output of the lth layer is given by

$$(\sigma(y_{j^{(i)}}^{(i)}), \sigma(-y_{j^{(i)}}^{(i)}), 0, \dots, 0)^\top.$$

For the final layer, we let $\boldsymbol{W}_{L_y} = (1, -1, 0, \dots, 0)$. Using the fact that $\sigma(z) - \sigma(-z) = z$, we obtain

$$f_y(\boldsymbol{U}^{\rightarrow}\boldsymbol{h}_j^{\rightarrow}, \boldsymbol{U}^{\leftarrow}\boldsymbol{h}_j^{\leftarrow}, \boldsymbol{z}_{j^{(i)}}^{(i)}; \boldsymbol{\Theta}_y) = y_{j^{(i)}}^{(i)}$$

We have found $\boldsymbol{\Theta}$ such that $\hat{R}_n^{\mathrm{RNN}}(\boldsymbol{\Theta}) = 0$ and $\|\mathrm{vec}(\boldsymbol{\Theta})\|_2^2 \leq \mathcal{O}(n^3)$ (recall that $L_y \leq \mathcal{O}(1)$). As a result, $\hat{\boldsymbol{\Theta}}_\varepsilon$ must also satisfy $\left\|\mathrm{vec}(\hat{\boldsymbol{\Theta}}_\varepsilon)\right\|_2^2 \leq \mathcal{O}(n^3)$.

On the other hand, notice that as a function of $\boldsymbol{U}\boldsymbol{h} = (\boldsymbol{U}^{\rightarrow}\boldsymbol{h}^{\rightarrow}, \boldsymbol{U}^{\leftarrow}\boldsymbol{h}^{\leftarrow})$, f_y is $\mathfrak{L}$-Lipschitz with

$$\mathfrak{L} = \left\|\boldsymbol{W}_{L_y}\right\|_{\mathrm{op}} \left\|\boldsymbol{W}_{L_y-1}\right\|_{\mathrm{op}} \dots \|\boldsymbol{W}_2\|_{\mathrm{op}}.$$

From Proposition 34, using the fact that $\|\cdot\|_{\mathrm{op}} \le \|\cdot\|_{\mathrm{F}}$ and the AM-GM inequality, we obtain

$$\frac{1}{L_y}\|\mathrm{vec}(\boldsymbol{\Theta})\|_2^2 \ge \left(\mathfrak{L}^2\|\boldsymbol{U}\|_{\mathrm{op}}^2\right)^{1/L_y} \ge \Omega\left(\left(\frac{N}{\log d_h}\right)^{1/L_y}\right)$$

to achieve population risk less than some absolute constant $c_3 > 0$. Recall that $\log d_h \le N^c$ for some $c < 1$. The proof is completed by noticing that unless $n \ge \Omega(N^{c_1})$ for some absolute constant $c_1 > 0$, $\left\|\mathrm{vec}(\hat{\boldsymbol{\Theta}}_\varepsilon)\right\|_2$ will always be less than the lower bound above, with some absolute constant probability $c_2 > 0$ over the training set. ■

Appendix D. Details of Section 5

We first define the class of algorithms considered in Theorem 8. Importantly, this lower bound holds regardless of the loss function used for training; for some arbitrary loss $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, we define the empirical risk of the FFN as

$$\hat{\mathcal{L}}^{\mathrm{FFN}}(f, \boldsymbol{W}) \coloneqq \frac{1}{nN}\sum_{i=1}^{n}\sum_{j=1}^{N}\ell(y_j^{(i)}, f(\boldsymbol{T}^{(i)}, \boldsymbol{W}\boldsymbol{x}^{(i)})_j),$$

where $\boldsymbol{T}^{(i)} = (\boldsymbol{t}_1^{(i)}, \dots, \boldsymbol{t}_N^{(i)})$. We still use $R^{\mathrm{FFN}}(f, \boldsymbol{W})$ for expected squared loss. Our lower bound covers a broad set of algorithms, characterized by the following definition.

Definition 41 *Let $\mathcal{A}_{\mathrm{SP}}$ denote the set of algorithms that return a stationary point of the regularized empirical risk. Specifically, for every $A \in \mathcal{A}_{\mathrm{SP}}$, $A(S_n)$ returns $f_{A(S_n)}$, $\boldsymbol{W}_{A(S_n)}$, such that*

$$\nabla_{\boldsymbol{W}}\hat{\mathcal{L}}^{\mathrm{FFN}}(f_{A(S_n)}, \boldsymbol{W}_{A(S_n)}) + \lambda\boldsymbol{W}_{A(S_n)} = 0,$$

for some $\lambda > 0$ depending on A. S_n above denotes the training set. Let $\mathcal{A}_{\mathrm{ERM}}$ denote the set of algorithms that return the min-norm approximate ERM. Specifically, every $A \in \mathcal{A}_{\mathrm{ERM}}$ returns

$$A(S_n) = \operatorname*{arg\,min}_{\{f, \boldsymbol{W}:\hat{\mathcal{L}}^{\mathrm{FFN}}(f,\boldsymbol{W})\le\varepsilon\}} \|\boldsymbol{W}\|_{\mathrm{F}},$$

for some $\varepsilon \ge 0$. Define $\mathcal{A} \coloneqq \mathcal{A}_{\mathrm{SP}} \cup \mathcal{A}_{\mathrm{ERM}}$.

In particular, $\mathcal{A}$ goes beyond constrained ERM in that it also includes the (ideal) output of first-order optimization algorithms with weight decay, or ERM with additional ℓ_2 penalty on the weights.

D.1. Proof of Theorem 8

Let $\boldsymbol{u}$ be sampled uniformly from $\mathbb{S}^{d-1}$ independently from $\boldsymbol{p} = (t_1, \boldsymbol{x})$, and note that we have

$$\sup_{\boldsymbol{u}\in\mathbb{S}^{d-1}} \mathbb{E}\left[(y_j - f_{A(S_n)}(t_1, \boldsymbol{W}_{A(S_n)}\boldsymbol{x})_j)^2\right] \ge \mathbb{E}_{\boldsymbol{u}\sim\mathrm{Unif}(\mathbb{S}^{d-1}),j,y,\boldsymbol{p}\sim\mathcal{P}}\left[(y_j - f_{A(S_n)}(t_1, \boldsymbol{W}_{A(S_n)}\boldsymbol{x})_j)^2\right],$$

for all $A \in \mathcal{A}$. From this point, we will simply use f for $f_{A(S_n)}$ and $\boldsymbol{W}$ for $\boldsymbol{W}_{A(S_n)}$. Next, we argue that the output weights of any algorithm in $\mathcal{A}$ satisfy

$$\boldsymbol{w}_k = \sum_{i=1}^{n}\alpha_k^{(i)}\boldsymbol{x}^{(i)}, \quad \forall k \in [m_1],$$

for some coefficients $(\alpha_k^{(i)})_{i\in[n],k\in[m_1]}$. This is straightforward to verify for $A \in \mathcal{A}_{\mathrm{SP}}$, as

$$\nabla_{\boldsymbol{w}_k} \hat{\mathcal{L}}^{\mathrm{FFN}}(f, \boldsymbol{W}) \in \mathrm{span}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}).$$

For $A \in \mathcal{A}_{\mathrm{ERM}}$, note that $\hat{\mathcal{L}}^{\mathrm{FFN}}$ only depends on $\boldsymbol{w}_k$ through its projection on $\mathrm{span}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)})$. As a result, any minimum-norm ε-ERM would satisfy $\boldsymbol{w}_k \in \mathrm{span}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)})$.

Note that for $n \le Nd$, the span of $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ is n-dimensional with probability 1 over S_n. Let $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}$ denote an orthonormal basis of $\mathrm{span}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)})$, and let $\boldsymbol{V} = (\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)})^\top \in \mathbb{R}^{n\times Nd}$. Recall that for the simple-1STR model considered here, $y_j = y = \langle \boldsymbol{u}, \boldsymbol{x}_{t_q} \rangle$ for $j \in [N]$. Then,

$$\mathbb{E}_{\boldsymbol{u},y,j,\boldsymbol{p}}\big[(y_j - f(t_1, \boldsymbol{W}\boldsymbol{x})_j)^2\big] \ge \mathbb{E}_{\boldsymbol{u},t_1,\boldsymbol{V}\boldsymbol{x}}[\mathrm{Var}(y \,|\, \boldsymbol{u}, t_1, \boldsymbol{V}\boldsymbol{x})] = \mathbb{E}_{\boldsymbol{u},t_1,\boldsymbol{V}\boldsymbol{x}}[\mathrm{Var}(\langle \boldsymbol{P}_{t_1}\boldsymbol{u}, \boldsymbol{x}\rangle \,|\, \boldsymbol{u}, t_1, \boldsymbol{V}\boldsymbol{x})],$$

where $\boldsymbol{P}_{t_1} \in \mathbb{R}^{Nd\times d}$ has the form $\big(\underbrace{\boldsymbol{0}_d, \ldots, \mathbf{I}_d}_{t_1}, \ldots, \boldsymbol{0}_d\big)^\top$. The conditioning above comes from the fact that via training, f and $\boldsymbol{W}$ can depend on $\boldsymbol{u}$, but the prediction depends on $\boldsymbol{x}$ only through $\boldsymbol{V}\boldsymbol{x}$. Consequently, we replace the predicition of the FFN by the best predictor having access to $\boldsymbol{u}$, t_1, and $\boldsymbol{V}\boldsymbol{x}$. Note that t_1, $\boldsymbol{u}$, and $\boldsymbol{V}\boldsymbol{x}$ are jointly independent, and the joint distribution $\big(\langle \boldsymbol{P}_{t_1}\boldsymbol{u}, \boldsymbol{x}\rangle, \boldsymbol{V}\boldsymbol{x}\big)$ is given by $\mathcal{N}\left(0, \begin{pmatrix} 1 & \boldsymbol{V}\boldsymbol{P}_{t_1}\boldsymbol{u} \\ \boldsymbol{u}^\top \boldsymbol{P}_{t_1}^\top \boldsymbol{V}^\top & \mathbf{I}_n \end{pmatrix}\right)$, thus we have

$$\mathrm{Var}(\langle \boldsymbol{P}_{t_1}\boldsymbol{u}, \boldsymbol{x}\rangle \,|\, \boldsymbol{u}, t_1, \boldsymbol{V}\boldsymbol{x}) = 1 - \|\boldsymbol{V}\boldsymbol{P}_{t_1}\boldsymbol{u}\|^2.$$

In particular,

$$\mathbb{E}_{\boldsymbol{u}}[\mathrm{Var}(\langle \boldsymbol{P}_{t_1}\boldsymbol{u}, \boldsymbol{x}\rangle \,|\, \boldsymbol{u}, t_1, \boldsymbol{V}\boldsymbol{x})] = 1 - \frac{1}{d}\sum_{i=1}^{n}\left\|\boldsymbol{P}_{t_1}^\top \boldsymbol{v}^{(i)}\right\|^2,$$

and

$$\begin{aligned}\mathbb{E}_{\boldsymbol{u},t_1}[\mathrm{Var}(\langle \boldsymbol{P}_{t_1}\boldsymbol{u}, \boldsymbol{x}\rangle \,|\, \boldsymbol{u}, t_1, \boldsymbol{V}\boldsymbol{x})] &= 1 - \frac{1}{Nd}\sum_{t_1=1}^{N}\sum_{i=1}^{n}\left\|\boldsymbol{P}_{t_1}^\top \boldsymbol{v}^{(i)}\right\|^2 \\ &= 1 - \frac{1}{Nd}\sum_{i=1}^{n}\left\|\boldsymbol{v}^{(i)}\right\|^2 = 1 - \frac{n}{Nd}.\end{aligned}$$

■

Appendix E. Auxiliary Lemmas

Lemma 42 *Suppose $\boldsymbol{A} \in \mathbb{R}^{d_1\times d_2}$ and $\boldsymbol{B} \in \mathbb{R}^{d_2\times d_3}$. Then, for all $r, s \ge 1$ and $p, q \ge 1$ such that $1/p + 1/q = 1$, we have*

$$\|\boldsymbol{A}\boldsymbol{B}\|_{r,s} \le \|\boldsymbol{A}\|_{r,p}\|\boldsymbol{B}\|_{q,s}.$$

Proof First, we note that for any vector $\boldsymbol{b} \in \mathbb{R}^{d_2}$ we have

$$\|\boldsymbol{A}\boldsymbol{b}\|_r = \left\|\sum_{j=1}^{d_2} b_j \boldsymbol{A}_{:,j}\right\|_r \le \sum_{j=1}^{d_2}|b_j|\|\boldsymbol{A}_{:,j}\|_r \le \|\boldsymbol{A}\|_{r,p}\|\boldsymbol{b}\|_q,$$

where the last inequality holds for all conjugate indices p, q and follows from Hölder's inequality. We now have

$$\|\boldsymbol{A}\boldsymbol{B}\|_{r,s}^s = \sum_{j=1}^{d_3} \|\boldsymbol{A}\boldsymbol{B}_{:,j}\|_r^s \leq \sum_{j=1}^{d_3} \|\boldsymbol{A}\|_{r,p}^s \|\boldsymbol{B}_{:,j}\|_q^s = \|\boldsymbol{A}\|_{r,p} \|\boldsymbol{B}\|_{q,s}.$$

■

The next lemma follows from standard Gaussian integration.

Lemma 43 *Suppose $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then $\mathrm{Var}(\|\boldsymbol{x}\|^2) = 2\,\mathrm{tr}(\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma}) + 4\boldsymbol{\mu}^\top \boldsymbol{\Sigma} \boldsymbol{\mu}$.*

The following lemma combines two different techniques for establishing a packing number over the unit ball, the first construction uses volume comparison, whereas the second construction uses Maurey's sparsification lemma, both of which are well-established in the literature.

Lemma 44 *Let $\mathcal{P}$ denote the ϵ-packing number of the unit ball in $\mathbb{R}^d$. We have*

$$\log \mathcal{P} \leq \left\{ d \log\left(1 + \frac{2}{\epsilon}\right) \right\} \wedge \left\{ \frac{1}{\epsilon^2}(1 + \log(1 + 2d\epsilon^2)) \right\}.$$

Finally, the lemma below allows us to approximate arbitrary Lipschitz functions with two-layer feedforward networks.

Lemma 45 ([7, Propositions 1 and 6]) *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ satisfies $|f(\boldsymbol{x})| \leq LR$ and $|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|_2$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ with $\|\boldsymbol{x}\|_2 \leq R$ and $\|\boldsymbol{x}'\|_2 \leq R$ and some constants $L, R > 0$. Then, for every $\varepsilon > 0$, there exists a positive integer m and $\boldsymbol{W} \in \mathbb{R}^{m \times d}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{a} \in \mathbb{R}^m$, such that*

$$\sup_{\|\boldsymbol{x}\|_2 \leq R} \left| f(\boldsymbol{x}) - \boldsymbol{a}^\top \sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \right| \leq \varepsilon.$$

Additionally, we have

$$m \leq C_d \Big(\frac{LR(1 + \log(LR/\varepsilon))}{\varepsilon} \Big)^d, \quad \left\| \boldsymbol{W}^\top \right\|_{2,\infty} \leq \frac{1}{R}, \quad \|\boldsymbol{b}\|_\infty \leq 1, \quad \|\boldsymbol{a}\|_2 \leq \frac{C_d LR}{\sqrt{m}} \cdot \left(\frac{LR(1 + \log(LR/\varepsilon))}{\varepsilon} \right)^{\frac{d+1}{2}}.$$