The Cost of Robustness: Tighter Bounds on Parameter Complexity for Robust Memorization in ReLU Nets

Yujun Kim* KYUJUN02@KAIST.AC.KR

Chaewon Moon* CHAEWON.MOON@KAIST.AC.KR

Chulhee Yun CHULHEE.YUN@KAIST.AC.KR

KAIST, Seoul, Republic of Korea

Abstract

We study the parameter complexity of *robust memorization* for ReLU networks: the number of parameters required to interpolate any given dataset with ϵ-separation between differently labeled points, while ensuring predictions remain consistent within a μ-ball around each training sample. We establish upper and lower bounds on the parameter count as a function of the robustness ratio $\rho = \mu/\epsilon$. Unlike prior work, we provide a fine-grained analysis across the entire range $\rho \in (0, 1)$ and obtain tighter upper and lower bounds that improve upon existing results. Our findings reveal that the parameter complexity of robust memorization *matches* that of *non-robust* memorization when ρ is small, but grows with increasing ρ.

1. Introduction

The topic of memorization investigates the expressive power of neural networks required to fit any given dataset exactly. This line of inquiry seeks to determine the minimal network size—measured in the number of parameters, or equivalently, parameter complexity—needed to interpolate any finite collection of N labeled examples. A number of works study both upper and lower bounds on the parameter complexity [3, 4, 13, 18]. The VC-dimension implies a lower bound of $\Omega(\sqrt{N})$ [1, 8], while Vardi et al. [16] show that $\tilde{\Theta}(\sqrt{N})$ parameters suffice for ReLU networks. Together, these results establish that memorizing any N distinct samples with ReLU networks can be done with $\tilde{\Theta}(\sqrt{N})$ parameters, tight up to logarithmic factors.

We now turn to a more challenging task beyond mere interpolation of data: **robust memorization**. We aim to quantify the additional parameter complexity required for a network to remain *robust* against adversarial attacks, going beyond standard non-robust memorization. To address the sensitivity of neural networks to small adversarial perturbations [2, 5, 9, 10, 14, 19], we consider the setting in which not only the data points but all points within a distance μ—referred to as the *robustness radius*—from each data point must be mapped to the corresponding label. More concretely, for any dataset with ϵ-separation between differently labeled data points, the network must memorize the dataset and the prediction must remain consistent within a μ-ball centered at each training sample. The parameter complexity for robust memorization is governed by the *robustness ratio* $\rho = \mu/\epsilon \in (0, 1)$ rather than the individual values of μ and ϵ (Appendix B illustrates the details). However, a precise understanding of how this complexity scales with ρ remains limited.

* Authors contributed equally to this paper.

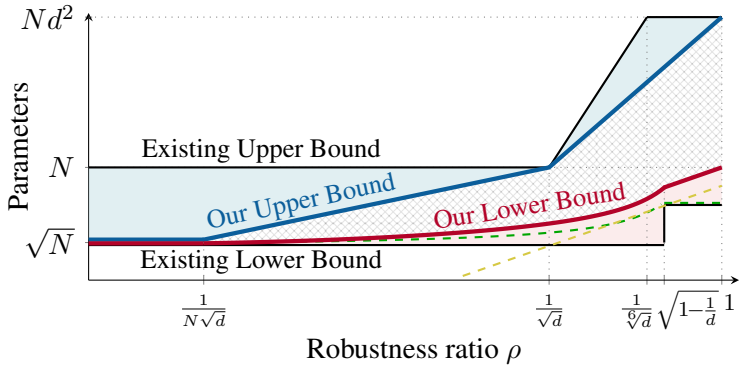


Figure 1: Summary of parameter bounds on a log-log scale when input dimension $d = \Theta(\sqrt{N})$. We omit constant factors in both axes. Solid blue and red curves show the sufficient (Theorem 7) and necessary (Theorem 3) numbers of parameters, respectively; the solid black lines are the best prior bound. Light-blue shading highlights our improvement in the upper bound, and light-red shading highlights our improvement in the lower bound. The cross-hatched area marks the remaining gap. Notably, this gap disappears in the smallest ρ regime. The yellow and green dashed line denotes the first term (Theorem 4) and the second term (Theorem 5) in Theorem 3, respectively.

1.1. Summary of Contribution

We study how the number of ReLU network parameters required for robust memorization varies with the robustness ratio ρ. We present improved upper and lower bounds for all $\rho \in (0, 1)$, which are tight in some regimes and substantially reduce the gap elsewhere. Figure 1 illustrates the improvement, and Appendix A discusses the prior bounds in detail. While results are mainly discussed in ℓ_2-norm, we also present the extension to general ℓ_p-norm in Appendix F.

- **Necessary Conditions for Robust Memorization.** We show that the first hidden layer must have a width of at least $\rho^2 \min\{N, d\}$, where d is the input dimension and N is the dataset size, by constructing a dataset that cannot be robustly memorized using a smaller width. Consequently, the network must have at least $\Omega(\rho^2 \min\{N, d\}d)$ parameters. Moreover, we prove that at least $\Omega(\sqrt{N/(1-\rho^2)})$ parameters are necessary by analyzing the VC-dimension. Combining these two results, we obtain a tighter lower bound on the parameter complexity of robust memorization of the form
$$P = \Omega\Big((\rho^2 \min\{N, d\} + 1)d + \min\Big\{\frac{1}{\sqrt{1-\rho^2}}, \sqrt{d}\Big\}\sqrt{N}\Big).$$

- **Sufficiency Conditions for Robust Memorization.** We establish improved upper bounds on the parameter count by analyzing three distinct regimes of ρ, tightening the bound in each case. For $\rho \in \left(0, \frac{1}{5N\sqrt{d}}\right]$, we achieve robust memorization using $\tilde{O}(\sqrt{N})$ parameters, matching the existing lower bound. For $\rho \in \left(\frac{1}{5N\sqrt{d}}, \frac{1}{5\sqrt{d}}\right]$, we obtain robust memorization with $\tilde{O}(Nd^{1/4}\rho^{1/2})$ parameters up to an arbitrarily small error, which interpolates between the existing lower bound $\sqrt{N}$ and the existing upper bound N. Finally, for larger values of ρ, where $\rho \in \left(\frac{1}{5\sqrt{d}}, 1\right]$, robust memorization is achieved with $\tilde{O}(Nd^2\rho^4)$ parameters, which interpolates between the existing upper bound N and Nd^2.

All together, we provide, to the best of our knowledge, the first theoretical characterization showing that the number of parameters required for robust memorization increases with the robustness radius ρ. Notably, when $\rho < \frac{1}{5N\sqrt{d}}$, the same number of parameters as in classical (non-robust) memorization suffices for robust memorization. These results suggest that, in terms of parameter count, achieving robustness against adversarial attacks is relatively inexpensive when the robustness radius is small. As the radius grows, however, the number of required parameters increases, reflecting the rising cost of achieving stronger robustness.

2. Preliminaries

2.1. Notation and the Network Architecture

Throughout the paper, we use d to denote the input dimension of the data, N to denote the number of data points in a dataset, and C to denote the number of classes for classification. For a natural number $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$. We use $\mathcal{B}_2(\boldsymbol{x}, \mu)$ to denote an open ℓ_2-ball centered at $\boldsymbol{x}$ with a radius μ. We use $\tilde{O}(\cdot)$ to hide the poly-logarithmic dependencies in problem parameters such as N, d, and p.

We define the neural network f recursively over the L layers. With $\boldsymbol{a}_0 = \boldsymbol{x}$, we let

$$\begin{aligned} \boldsymbol{a}_l &= \sigma(\boldsymbol{W}_l \boldsymbol{a}_{l-1}(\boldsymbol{x}) + \boldsymbol{b}_l) \text{ for } l = 1, 2, \ldots, L-1 \\ f(\boldsymbol{x}) &= \boldsymbol{W}_L(\boldsymbol{a}_L) + \boldsymbol{b}_L, \end{aligned}$$

where the activation $\sigma(\boldsymbol{u}) := \max\{\boldsymbol{0}, \boldsymbol{u}\}$ is the element-wise ReLU. We use $d_1, \ldots, d_{L-1}$ to denote the width of $L-1$ hidden layers. For $l \in [L]$, the symbols $\boldsymbol{W}_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and $\boldsymbol{b}_l \in \mathbb{R}^{d_l}$ denote the weight matrix and the bias vector for the l-th layer, respectively; here, we use the convention $d_0 = d$ and $d_L = 1$.

We count the number of parameters P of neural network f as the count of all parameters including zeros in the weight matrices and biases. Thus, the parameter count given the widths d_l is

$$P = \sum_{l=1}^{L} (d_{l-1} + 1) \cdot d_l. \tag{1}$$

We denote the set of neural networks with input dimension d and at most P parameters by

$$\mathcal{F}_{d,P} = \left\{ f : \mathbb{R}^d \to \mathbb{R} \mid f \text{ is a neural network with at most } P \text{ parameters} \right\}.$$

2.2. Dataset and Robust Memorization

For $d \geq 1$, $N \geq 2$, and $C \geq 2$, let $\mathcal{D}_{d,N,C}$ be the collection of all datasets $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times [C]$, such that $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ for all $i \neq j$. Hence, any $\mathcal{D} \in \mathcal{D}_{d,N,C}$ is a pairwise distinct d-dimensional dataset of size N with labels in $[C]$.

Definition 1 *For $\mathcal{D} \in \mathcal{D}_{d,N,C}$, the separation constant $\epsilon_{\mathcal{D}}$ is defined as*

$$\epsilon_{\mathcal{D}} := \frac{1}{2} \min \left\{ \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \mid (\boldsymbol{x}_i, y_i), (\boldsymbol{x}_j, y_j) \in \mathcal{D},\ y_i \neq y_j \right\}.$$

As we consider $\mathcal{D}$ with $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ for all $i \neq j$, we have $\epsilon_{\mathcal{D}} > 0$. Next, we define robust memorization of the given dataset.

Definition 2 *For $\mathcal{D} \in \mathcal{D}_{d,N,C}$ and a given robustness ratio $\rho \in (0,1)$, define the robustness radius as $\mu := \rho\epsilon_{\mathcal{D}}$. We say that a function $f : \mathbb{R}^d \to \mathbb{R}$ ρ-robustly memorizes $\mathcal{D}$ if*

$$f(\boldsymbol{x}') = y_i, \quad \text{for all } (\boldsymbol{x}_i, y_i) \in \mathcal{D} \text{ and } \boldsymbol{x}' \in \mathcal{B}_2(\boldsymbol{x}_i, \mu),$$

and $\mathcal{B}_2(\boldsymbol{x}_i, \mu)$ is referred to as the robustness ball of $\boldsymbol{x}_i$.

We emphasize that the range $\rho \in (0,1)$ covers the entire regime in which robust memorization is possible. Specifically, for $\rho > 1$, requiring memorization of $\rho\epsilon_{\mathcal{D}}$-radius neighbor of each data point leads to a contradiction as $\mathcal{B}_2(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D}}) \cap \mathcal{B}_2(\boldsymbol{x}_j, \rho\epsilon_{\mathcal{D}}) \neq \emptyset$ for some $y_i \neq y_j$. Moreover, if $\rho = 1$, any continuous function f cannot ρ-robustly memorize $\mathcal{D}$. If f is continuous and 1-robustly memorizes $\mathcal{D}$, it leads to a contradiction that $\overline{\mathcal{B}_2(\boldsymbol{x}_i, \epsilon_{\mathcal{D}})} \cap \overline{\mathcal{B}_2(\boldsymbol{x}_j, \epsilon_{\mathcal{D}})} \neq \emptyset$ for some $y_i \neq y_j$.

We provide a clarification on why considering only the robustness ratio ρ is both necessary and sufficient for robust memorization in Appendix B.

3. Necessary Number of Parameters for Robust Memorization

In this section, we present lower bounds on the number of parameters and width required for robust memorization over $\rho \in (0,1)$. Theorem 3 presents our main lower bound result.

Theorem 3 *Let $\rho \in (0,1)$. Suppose for any $\mathcal{D} \in \mathcal{D}_{d,N,2}$, there exists a neural network $f \in \mathcal{F}_{d,P}$ that can ρ-robustly memorize $\mathcal{D}$. Then, the number of parameters P must satisfy*

$$P = \Omega\Big((\rho^2 \min\{N, d\} + 1)d + \min\Big\{\frac{1}{\sqrt{1-\rho^2}}, \sqrt{d}\Big\}\sqrt{N}\Big).$$

The theorem states a necessary number of parameters for binary classification ($C = 2$). It can be trivially extended to the case $C > 2$, since multiclass classification requires at least as many parameters as binary classification. Moreover, while Theorem 3 focuses on ℓ_2-norm, we extend the necessity results to the general ℓ_p-norm in Appendix F. The lower bound on the number of parameters consists of two parts: one derived from the requirement on the first hidden layer width and the other from the VC dimension.

First Term: Necessity Condition by the First Hidden Layer Width. The first term $\Omega((\rho^2 \min\{N, d\} + 1)d)$ comes from the following proposition on the first hidden layer width.

Proposition 4 *There exists a dataset $\mathcal{D} \in \mathcal{D}_{d,N,2}$ such that, for any $\rho \in (0,1)$, any neural network $f : \mathbb{R}^d \to \mathbb{R}$ that ρ-robustly memorizes $\mathcal{D}$ must have the first hidden layer width at least $\rho^2 \min\{N-1, d\}$.*

The detailed proof of Theorem 4 is in Appendix D.1, and we refer readers to Appendix C.1 for a sketch of ideas.

Let us compare our results with some prior works. First, Egosi et al. [6] show that logarithmic width in N is both necessary and sufficient for robust memorization. However, their result is only sufficient under logarithmic width when $\rho \leq \frac{1}{\sqrt{d}}$, and therefore does not contradict ours in the regime where ρ necessitates width $\Omega(1)$. Regarding the necessity condition, Theorem 4 provides a tighter lower bound for all $\rho \geq 1/\sqrt{\min\{N-1, d\}}$ where our lower bound dominates the trivial lower bound 1.

Furthermore, we compare the results with the prior work in terms of ℓ_∞-norm, demonstrating that we recover a stronger version of an existing bound. Our Theorem 4 (under ℓ_2-norm) is carefully

translated to general ℓ_p-norms in Appendix F. In particular, for every $\rho \in (0,1)$, the analysis reveals that the same lower bound on width $\Omega(\rho^2 \min\{N,d\})$ holds whenever $p \geq 2$ and thus for $p=\infty$. Yu et al. [17] prove that the first hidden layer width d is necessary for ρ-robustly memorizing a certain dataset under ℓ_∞-norm, provided that $N > d$ and $\rho = 0.8$. As shown in Theorem 33, their result can in fact be refined to a width requirement $\min\{N-1,d\}$ for any $\rho \in (1/2,1)$, *without* the condition $N > d$. When $\rho \in (1/2,1)$, Theorem 4 translated to ℓ_∞-norm has necessity on width $\Omega(\rho^2 \min\{N,d\}) = \Omega(\min\{N,d\})$, recovering Theorem 33 and hence the result by Yu et al. [17].

Second Term: Necessity Condition by the VC-Dimension. Now, let us look at the necessary number of parameters given by the VC-dimension of the function class.

Proposition 5 *Let $\rho \in \left(0, \sqrt{1-\frac{1}{d}}\right]$. Suppose for any $\mathcal{D} \in \mathcal{D}_{d,N,2}$, there exists $f \in \mathcal{F}_{d,P}$ that ρ-robustly memorizes $\mathcal{D}$. Then, the number of parameters P must satisfy $P = \Omega\left(\sqrt{N/(1-\rho^2)}\right)$.*

The detailed proof of Theorem 5 is in Appendix D.2. Before the derivation of our result, we briefly review how the existing bound is obtained using VC-dimension arguments. Gao et al. [7], Li et al. [11] prove that for sufficiently large ρ, whenever $\mathcal{F}_{d,P}$ contains ρ-robust memorizer of any $\mathcal{D} \in \mathcal{D}_{d,N,2}$, then $\text{VC-dim}(\mathcal{F}_{d,P}) = \Omega(Nd)$. Combining this with a known upper bound $\text{VC-dim}(\mathcal{F}_{d,P}) = O(P^2)$ [8], they obtain $P = \Omega(\sqrt{Nd})$.

However, the prior lower bound $\Omega(\sqrt{Nd})$ is only known to apply for sufficiently large ρ, without specifying the precise range. Before our result, the only available lower bound for $\rho \in (0,1)$ was the one that trivially comes from non-robust memorization: $\Omega(\sqrt{N})$. No VC-dimension-based lower bound tailored to robust memorization is currently known.

In Theorem 5, we carefully characterize how the VC-dimension scales over the range $\rho \in (0, \sqrt{1-1/d}]$. In this range of ρ, we show whenever $\mathcal{F}_{d,P}$ contains ρ-robust memorizer of any $\mathcal{D} \in \mathcal{D}_{d,N,2}$, then $\text{VC-dim}(\mathcal{F}_{d,P}) = \Omega(N/(1-\rho^2))$; this thus gives the tighter bound $P = \Omega(\sqrt{N/(1-\rho^2)})$. At the endpoint $\rho = \sqrt{1-1/d}$, Theorem 5 implies that $\Omega(\sqrt{Nd})$ parameters are required. Therefore, the same lower bound applies for all $\rho \geq \sqrt{1-1/d}$, characterizing the regime in which the existing bound of $\sqrt{Nd}$ holds.

Combining Theorem 5 over $\rho \in (0, \sqrt{1-1/d}]$ and the $\Omega(\sqrt{Nd})$ bound over $\rho \in [\sqrt{1-1/d}, 1)$, we obtain the second term $\Omega(\min\{1/\sqrt{1-\rho^2}, \sqrt{d}\}\sqrt{N})$ in Theorem 3.

4. Sufficient Number of Parameters for Robust Memorization

In this section, we present sufficiency conditions on the number of parameters for robust memorization. One of our upper bounds is based on a relaxed notion of robust memorization, for which we define the ρ-robust memorization error of a neural network.

Definition 6 *For any $\mathcal{D} \in \mathcal{D}_{d,N,C}$, we define the ρ-robust memorization error of a network $f : \mathbb{R}^d \to \mathbb{R}$ on $\mathcal{D}$ as*

$$\mathcal{L}_\rho(f,\mathcal{D}) := \max_{(\boldsymbol{x}_i,y_i)\in\mathcal{D}} \mathbb{P}_{\boldsymbol{x}'\sim \mathit{Unif}(\mathcal{B}(\boldsymbol{x}_i,\mu))}[f(\boldsymbol{x}') \neq y_i],$$

where $\mu = \rho\epsilon_{\mathcal{D}}$. When $\mathcal{L}_\rho(f,\mathcal{D}) < \eta$, we say f can ρ-robustly memorize $\mathcal{D}$ with error at most η.

Note that if a network f ρ-robustly memorizes $\mathcal{D}$, then the error is zero; that is, by definition $\mathcal{L}_\rho(f,\mathcal{D}) = 0$. We now state our main upper bounds, showing that any given dataset in $\mathcal{D}_{d,N,C}$ can be ρ-robustly memorized by a network with ρ-dependent number of parameters.

Theorem 7 *For any dataset $\mathcal{D} \in \mathcal{D}_{d,N,C}$ and $\eta \in (0,1)$, the following statements hold:*

(i) *If $\rho \in \left(0, \frac{1}{5N\sqrt{d}}\right]$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(\sqrt{N})$ that ρ-robustly memorizes $\mathcal{D}$.*

(ii) *If $\rho \in \left(\frac{1}{5N\sqrt{d}}, \frac{1}{5\sqrt{d}}\right]$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(Nd^{\frac{1}{4}}\rho^{\frac{1}{2}})$ that ρ-robustly memorizes $\mathcal{D}$ with error at most η.*

(iii) *If $\rho \in \left(\frac{1}{5\sqrt{d}}, 1\right)$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(Nd^2\rho^4)$ that ρ-robustly memorizes $\mathcal{D}$.*

We note that we omitted the trivial additive factor d that accounts for parameters connected to input neurons. The three regimes in the theorem collectively cover all values of $\rho \in (0,1)$ and provide upper bounds. We present a proof sketch in Appendix C.2. The proof of Theorem 7 is provided in Appendix E, and its extension to the ℓ_p-norm setting is discussed in Appendix F.2.

In contrast to prior results, Theorems 7(i) and 7(ii) provide the first upper bounds for robust memorization that are sublinear in N. Notably, our construction reveals a continuous interpolation—driven by the robustness ratio ρ—from the classical memorization complexity of $\Theta(\sqrt{N})$ to the existing upper bound of $\tilde{O}(N)$ in Theorem 7(ii), and further from $\tilde{O}(N)$ to $\tilde{O}(Nd^2)$ as shown in Theorem 7(iii). This demonstrates how the sufficient parameter complexity increases gradually with ρ, capturing the full spectrum of the robustness ratio.

Tight Bounds for Robust Memorization with Small ρ. Theorem 7(i) establishes a tight upper bound $\tilde{O}(\sqrt{N})$ on the number of parameters required for robust memorization when the robustness ratio satisfies $\rho < \frac{1}{5N\sqrt{d}}$. This shows that, for sufficiently small ρ, robust memorization requires the same parameter complexity $\tilde{\Theta}(\sqrt{N})$ as classical (non-robust) memorization.

Perfect Robust Memorization with Threshold Activation Function. Theorem 7(ii) requires the allowance of an arbitrarily small robust memorization error, which arises from the fact that ReLU-only networks can represent only continuous functions. In contrast, if we are allowed to use discontinuous threshold activation in combination with ReLU network, we can achieve ρ-robust memorization—and therefore zero robust memorization error— in the same rate as Theorem 7(ii).

Tight Bounds of Width. In the large ρ regime, the network construction in Theorem 7(iii) for ρ-robust memorization has width $\tilde{O}(\rho^2 \min\{N-1, d\})$. This shows that width $\tilde{O}(\rho^2 \min\{N-1, d\})$ is sufficient for ρ-robust memorization. The complementary lower bound by Theorem 4 states that a width of at least $\rho^2 \min\{N-1, d\}$ is also necessary, tightly characterizing the minimum scale of width for robust memorization up to logarithmic factors for this case. Details in Theorem 22.

Moreover, Egosi et al. [6] show that logarithmic width is necessary and sufficient for $\rho = \tilde{O}(1/\sqrt{d})$. Our constructions for $\rho = \tilde{O}(1/\sqrt{d})$ (in Theorem 7(i), 7(ii)) also has logarithmic width. Therefore, our result tightly characterizes the required width along the entire $\rho \in (0,1)$.

5. Conclusion

We presented a tighter characterization of the parameter complexity necessary and sufficient for robust memorization across the full range of robustness ratio $\rho \in (0,1)$. Our results established matching upper and lower bounds for small ρ, and showed that robustness demands significantly more parameters than classical memorization as ρ grows. These findings highlight how robustness fundamentally increases memorization difficulty under adversarial attacks.

We establish tight complexity bounds in the regime where $\rho < \frac{1}{5N\sqrt{d}}$. However, in the remaining cases, a gap between the upper and lower bounds persists. A precise characterization of the parameter complexity for some ρ remains open and is essential for a complete understanding of the trade-off between robustness and network complexity.

Acknowledgement

This work was supported by three Institute of Information & communications Technology Planning & Evaluation (IITP) grants (No. RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST); No. RS-2022-II220184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics; No. RS-2024-00457882, National AI Research Lab Project) funded by the Korean government (MSIT).

References

[1] Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

[2] Alexander Bastounis, Anders C Hansen, and Verner Vlačić. The mathematics of adversarial attacks in ai – why deep learning is unstable despite the existence of stable neural networks, 2025. URL https://arxiv.org/abs/2109.06098.

[3] Eric B Baum. On the capabilities of multilayer perceptrons. *Journal of complexity*, 4(3): 193–215, 1988.

[4] Sébastien Bubeck, Ronen Eldan, Yin Tat Lee, and Dan Mikulincer. Network size and weights size for memorization with two-layers neural networks, 2020. URL https://arxiv.org/abs/2006.02855.

[5] Gavin Weiguang Ding, Kry Yik Chau Lui, Xiaomeng Jin, Luyu Wang, and Ruitong Huang. On the sensitivity of adversarial robustness to input data distributions, 2019. URL https://arxiv.org/abs/1902.08336.

[6] Amitsour Egosi, Gilad Yehudai, and Ohad Shamir. Logarithmic width suffices for robust memorization, 2025. URL https://arxiv.org/abs/2502.11162.

[7] Ruiqi Gao, Tianle Cai, Haochuan Li, Cho-Jui Hsieh, Liwei Wang, and Jason D Lee. Convergence of adversarial training in overparametrized neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[8] Paul W Goldberg and Mark R Jerrum. Bounding the vapnik-chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18(2-3):131–148, 1995.

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

[10] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples, 2021. URL https://arxiv.org/abs/2010.03593.

[11] Binghui Li, Jikai Jin, Han Zhong, John Hopcroft, and Liwei Wang. Why robust generalization in deep learning is difficult: Perspective of expressive power. *Advances in Neural Information Processing Systems*, 35:4370–4384, 2022.

[12] Jiri Matousek. *Lectures on discrete geometry*, volume 212. Springer Science & Business Media, 2013.

[13] Sejun Park, Jaeho Lee, Chulhee Yun, and Jinwoo Shin. Provable memorization via deep neural networks using sub-linear parameters, 2021. URL https://arxiv.org/abs/2010.13363.

[14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. URL https://arxiv.org/abs/1312.6199.

[15] Matus Telgarsky. Benefits of depth in neural networks. *CoRR*, abs/1602.04485, 2016. URL http://arxiv.org/abs/1602.04485.

[16] Gal Vardi, Gilad Yehudai, and Ohad Shamir. On the optimal memorization power of ReLU neural networks, 2021. URL https://arxiv.org/abs/2110.03187.

[17] Lijia Yu, Xiao-Shan Gao, and Lijun Zhang. OPTIMAL ROBUST MEMORIZATION WITH RELU NEURAL NETWORKS. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=47hDbAMLbc.

[18] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small relu networks are powerful memorizers: a tight analysis of memorization capacity, 2019. URL https://arxiv.org/abs/1810.07770.

[19] Chongzhi Zhang, Aishan Liu, Xianglong Liu, Yitao Xu, Hang Yu, Yuqing Ma, and Tianlin Li. Interpreting and improving adversarial robustness of deep neural networks with neuron sensitivity. *IEEE Transactions on Image Processing*, 30:1291–1304, 2021. ISSN 1941-0042. doi: 10.1109/tip.2020.3042083. URL http://dx.doi.org/10.1109/TIP.2020.3042083.

Contents

Appendix A. What is Known So Far?

Existing Lower Bounds. Since classical memorization requires $\Omega(\sqrt{N})$ parameters, it follows that robust memorization must also satisfy a lower bound of at least $\Omega(\sqrt{N})$ parameters for any $\rho \in (0,1)$. A lower bound specific to robust memorization is established by the work of Li et al. [11], which shows that for input dimension d, $\Omega(\sqrt{Nd})$ parameters are necessary for robust memorization under ℓ_2-norm for sufficiently large ρ. However, the authors do not characterize the range of ρ over which this lower bound remains valid. Our Theorem 5 presented later shows that the $\Omega(\sqrt{Nd})$ lower bound can be extended to the range $\rho \in \left(\sqrt{1-1/d}, 1\right)$. Combining these observations, we obtain the following unified lower bound: suppose that for any dataset $\mathcal{D}$ with input dimension d and size N, there exists a neural network with P parameters that robustly memorizes $\mathcal{D}$ with robustness ratio ρ under ℓ_2-norm. Then, the number of parameters P must satisfy

$$P = \Omega\left(\left(1 + \sqrt{d} \cdot \mathbf{1}_{\rho \geq \sqrt{1-\frac{1}{d}}}\right)\sqrt{N} + d\right), \tag{2}$$

where the d term accounts for the parameters connected to the input neurons. In the setting $d = O(\sqrt{N})$, the lower bounds increase discontinuously from $\sqrt{N}$ to $\sqrt{Nd}$. This phenomenon is visualized in the case $d = \Theta(\sqrt{N})$ in Figure 1.

While our main analysis focuses on the ℓ_2-norm, there also exist results under the ℓ_∞-norm. In particular, Yu et al. [17] show that under the ℓ_∞-norm and certain assumptions, ρ-robust memorization requires the first hidden layer to have width at least d. Notably, this result does not extend to the ℓ_2-norm setting. Our analysis not only provides improvements in the ℓ_∞-norm regime but also removes the assumption on dataset—made in prior work—that the number of data points must be greater than d.

Existing Upper Bounds. From the work of Yu et al. [17], it is proven that $O(Nd^2)$ parameters suffice for any $\rho \in (0,1)$. See Appendix G.1 for an analysis of the parameter complexity of their construction. Furthermore, Egosi et al. [6] show that for $\rho \in \left(0, \frac{1}{\sqrt{d}}\right)$, a network of width $\log N$ suffices for ρ-robust memorization. Although they did not explicitly quantify the total number of parameters, their result, combined with the $O(Nd^2)$ upper bound, implies a total parameter count of $O(N(\log N)^2) = \tilde{O}(N)$. Additionally, their construction implicitly yields a smooth interpolation between $O(N)$ and $O(Nd^2)$ as ρ varies within the intermediate range $(1/\sqrt{d}, 1/\sqrt[6]{d})$, as shown in Appendix G.2.

To sum up, the existing upper bound states that for any dataset $\mathcal{D}$ with input dimension d and size N, there exist a neural network that achieves robust memorization on $\mathcal{D}$ with the robustness ratio ρ under ℓ_2-norm, with the number of parameters P bounded as follows:

$$P = \begin{cases} \tilde{O}(N+d) & \text{if } \rho \in (0, 1/\sqrt{d}]. \\ \tilde{O}(Nd^3\rho^6 + d) & \text{if } \rho \in (1/\sqrt{d}, 1/\sqrt[6]{d}]. \\ \tilde{O}(Nd^2) & \text{if } \rho \in (1/\sqrt[6]{d}, 1). \end{cases} \tag{3}$$

When $d = O(N)$, the upper bound transitions continuously from $\tilde{O}(N)$ to $\tilde{O}(Nd^2)$; Figure 1 shows an example when $d = \Theta(\sqrt{N})$.

Appendix B. Why Only $\rho = \mu/\epsilon_{\mathcal{D}}$ Matters

We describe both necessity and sufficiency conditions for robust memorization in terms of the ratio $\rho = \mu/\epsilon_{\mathcal{D}}$, rather than describing it in terms of individual values μ and $\epsilon_{\mathcal{D}}$. This is because the results remain invariant under scaling of the dataset.

Specifically regarding the sufficiency condition, suppose f ρ-robustly memorizes $\mathcal{D}$, and we have robustness radius $\mu = \rho\epsilon_{\mathcal{D}}$. Then for any $c > 0$, the scaled dataset $c\mathcal{D} := \{(c\boldsymbol{x}_i, y_i)\}_{i=1}^N$ with separation $\epsilon_{c\mathcal{D}} = c\epsilon_{\mathcal{D}}$ can be ρ-robustly memorized by the scaled function $\boldsymbol{x} \mapsto f(\frac{1}{c}\boldsymbol{x})$ where the robustness radius is $c\mu$ for this case. Moreover, the scaled function can be implemented through a network with the same number of parameters as the neural network f via scaling the first hidden layer weight matrix by $1/c$.

On the other hand, this implies that the necessity condition can also be characterized in terms of ρ. Suppose we have a dataset $\mathcal{D}$ with a fixed $\epsilon_{\mathcal{D}}$ for which ρ-robustly memorizing it requires a certain number of parameters P. Then, the scaled dataset $c\mathcal{D}$ with a separation $\epsilon_{c\mathcal{D}} = c\epsilon_{\mathcal{D}}$ also requires the same number of parameters for ρ-robust memorization. If $c\mathcal{D}$ can be memorized with less than P parameters, then by parameter rescaling from the previous paragraph, $\mathcal{D}$ can also be memorized with less than P parameters, leading to a contradiction.

Hence, the robustness ratio $\rho = \mu/\epsilon_{\mathcal{D}}$ captures the essential difficulty of robust memorization, independent of scaling. We henceforth state our upper and lower bounds in terms of ρ.

Appendix C. Key Proof Ideas

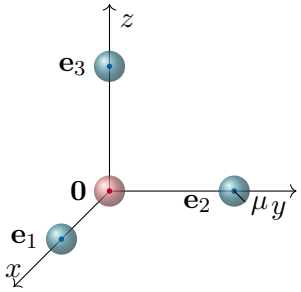
In this section, we outline the sketch of proof for some of the results from Sections 3 and 4.

C.1. Proof Sketch for Theorem 4

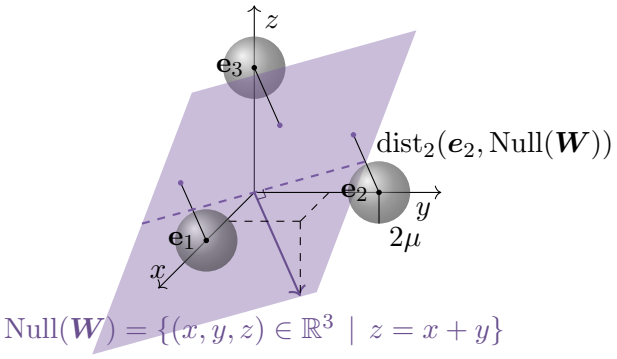
For simplicity, we sketch the case $N = d + 1$, where the proposition reduces to showing that the first hidden layer must have width at least $\rho^2 d$. To this end, we construct the dataset $\mathcal{D} = \{(\boldsymbol{e}_j, 1)\}_{j\in[d]} \cup \{(\mathbf{0}, 2)\}$, assigning label 1 to the standard basis points and label 2 to the origin, as shown in Appendix C.1.

Let f be an ρ-robust memorizer of $\mathcal{D}$ with the first hidden layer width m, and let $\boldsymbol{W} \in \mathbb{R}^{m\times d}$ denote the weight matrix of the first hidden layer. Since $\epsilon_{\mathcal{D}} = 1/2$, the robustness radius is $\mu = \rho\epsilon_{\mathcal{D}} = \rho/2$. For any $j \in [d]$, take any $\boldsymbol{x} \in \mathcal{B}_2(\boldsymbol{e}_j, \mu)$ and $\boldsymbol{x}' \in \mathcal{B}_2(\mathbf{0}, \mu)$. Then, $f(\boldsymbol{x}) = 1$ and $f(\boldsymbol{x}') = 2$ must hold, implying $\boldsymbol{W}\boldsymbol{x} \neq \boldsymbol{W}\boldsymbol{x}'$. Therefore, $\boldsymbol{x} - \boldsymbol{x}'$ should not lie in the null space of $\boldsymbol{W}$. All such possible differences $\boldsymbol{x} - \boldsymbol{x}'$ form a ball of radius 2μ around each standard basis point, illustrated as the gray ball in Appendix C.1. Thus, the distance between each standard basis point and the null space of $\boldsymbol{W}$ must be at least 2μ; otherwise, some gray balls intersect with the null space.

The null space of $\boldsymbol{W}$ is a $d - m$ dimensional space, assuming that $\boldsymbol{W}$ has full row rank. (The full proof generalizes even without this assumption.) By Theorem 10, the distance between the set of standard basis points and any subspace of dimension $d - m$ is at most $\sqrt{m/d}$. Therefore, $\rho = 2\mu \leq \mathrm{dist}_2(\{\boldsymbol{e}_j\}_{j\in[d]}, \mathrm{Null}(\boldsymbol{W})) \leq \sqrt{m/d}$ and thus the first hidden layer width satisfies $m \geq \rho^2 d$.



(a) Dataset for Theorem 4.



(b) $\mathrm{Null}(\boldsymbol{W}) \subset \mathbb{R}^3$ and the standard basis

Figure 2: In (a), blue balls have label 1; the red ball has label 2. (b) illustrates the distance between $\mathrm{Null}(\boldsymbol{W}) \subset \mathbb{R}^3$ and the standard basis for $\boldsymbol{W} = \begin{bmatrix}1 & 1 & -1\end{bmatrix}$ with the first hidden layer width 1.

C.2. Proof Sketch for Theorem 7

We now highlight the key construction techniques used to prove Theorem 7.

Separation-Preserving Dimensionality Reduction All three results in Theorem 7 leverage a strengthened version of the Johnson-Lindenstrauss (JL) lemma (Theorem 27) to project data from a high-dimensional space $\mathbb{R}^d$ (left in Figure 3) to a lower-dimensional space $\mathbb{R}^m$ (right), while preserving pairwise distances up to a multiplicative factor. Specifically, any pair of points that are $2\epsilon_{\mathcal{D}}$-separated in $\mathbb{R}^d$ can remain at least $\frac{4}{5}\sqrt{\frac{m}{d}}\epsilon_{\mathcal{D}}$-separated after the projection. Meanwhile, each

robustness ball of radius μ is preserved under the projection, given that we utilize a strengthened version of the JL lemma which use randomized orthonormal projections [12]. Since the geometry is preserved—specifically, the separation is maintained up to a factor of $\Omega(\sqrt{m/d})$ and the robustness radius remains unchanged under projection—we can ρ-robustly memorize data points in $\mathbb{R}^d$ by projecting them to $\mathbb{R}^m$ and memorizing the projected points.

In Theorems 7(i) and 7(ii), the data is projected to $\mathbb{R}^m$ with $m = O(\log N)$ in the first hidden layer. Since the construction of the remaining layers requires only $O(m)$ width, the network width is also logarithmic in N. The projection to logarithmic dimension while preserving the separation can only be applied to $\rho = O(1/\sqrt{d})$. This is because the separation through the projection is preserved by factor $\Omega(\sqrt{m/d})$, and hence if ρ is larger than the scale $O(\sqrt{m/d}) = \tilde{O}(1/\sqrt{d})$, the projected robustness balls may overlap one another. On the other hand, Theorem 7(iii), which deals with the largest ρ range, projects the data to a larger dimension. This ensures a greater amount of separation after the projection, allowing larger ρ at the cost of using more parameters. The dimension to which the points are projected is proportional to ρ^2, and thus increases with ρ.

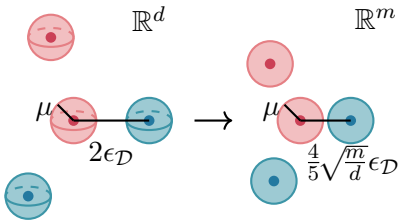


Figure 3: Separation-Preserving Projection

The idea of separation-preserving dimension reduction and deriving conditions under which robustness balls remain disjoint after projection is concurrently proposed by Egosi et al. [6]. However, their approach to ensuring the separability of robustness balls is substantially different from ours. Since the classical JL lemma does not inherently guarantee the preservation of ball separability, the authors do not rely on the JL lemma directly. Instead, they establish a probabilistic analogue through a technically involved analysis that bounds the probability that a random projection satisfies the required separation property. In contrast, we utilize a strengthened version of the JL lemma, and show that there exists a projection that preserves separability in a more straightforward manner, as shown in Appendix E.5.

Mapping to Lattices from Grid For Theorem 7(i) and 7(ii), we utilize the $\tilde{O}(\sqrt{N})$-parameter memorization developed by Vardi et al. [16]. In order to adopt the technique, it is necessary to assign a scalar value in $\mathbb{R}$ to each data point. This is because the construction memorizes the data after projecting them onto $\mathbb{R}$. Furthermore, this scalar assignment must meaningfully reflect the spatial structure of the data—preserving relative distances and neighborhood relationships of robustness ball.

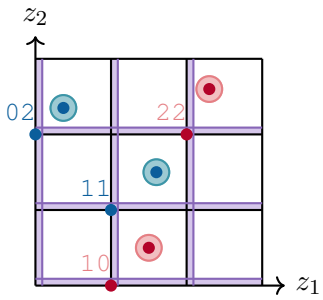
We achieve this using a coordinate map induced by spatial discretizations. Specifically, we reduce the dimension to $m = O(\log N)$ and partition the space $\mathbb{R}^m$ into a regular grid, assign an integer index to each grid cell—through the grid indexing, mapping each unit interval $\prod_{j\in[m]}[z_j, z_j + 1)$ to $z_1 R^{m-1} + z_2 R^{m-2} + \cdots + z_m$ for each $\boldsymbol{z} = (z_1, \cdots, z_m) \in \mathbb{Z}^m$ and some sufficiently large integer R—and associate each index with the label of the projected robustness ball contained in that cell. The network then memorizes the mapping from each grid index to its corresponding label.

Under the condition on ρ in Theorem 7(i), we show that after an appropriate translation, each projected robustness ball lies entirely within a single grid cell, and no two balls with different labels occupy the same cell, as shown in Appendix C.2.

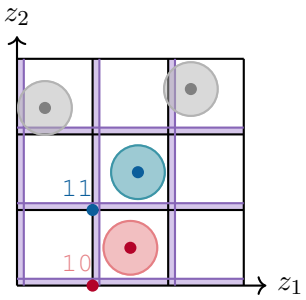
The main challenge in implementing the grid indexing is its discontinuity, which cannot be exactly represented by continuous ReLU networks. As a result, approximating it with ReLU introduces an error region where the ReLU approximation fails to implement the indexing correctly, as indicated by the purple shaded bands in Appendix C.2.

To address this, we ensure that all (projected) robustness balls remain un-intersected with the purple error region through a translation. This allows each point within the robustness ball to be safely mapped to a unique grid index, enabling ρ-robust memorization using only $\tilde{O}(\sqrt{N})$ parameters.

In Theorem 7(ii), we allow an arbitrarily small error in order to cover a larger range of ρ. Here, we no longer require every robustness ball to lie entirely within a single grid cell. We adopt a sequential memorization strategy. At each step, only a subset of data points is robustly memorized using the method from Theorem 7(i). We translate the data so that the subset of interest at the current step avoids the error region. Since only a few points are of interest per step, a larger ρ is allowed at the step. Other balls may cross cell boundaries as long as they do not interfere with the currently active cells, as shown in Appendix C.2. When a robustness ball that is not of interest intersects the error region, it may incur an error proportional to the volume of the intersection. However, we can make the error region—and thus the error at each step—arbitrarily small. Repeating this process yields robust memorization of all N data points with arbitrarily small error.



(a) The setting for Theorem 7(i), where each robust ball is entirely contained within a single grid cell, and no two balls with different labels occupy the same cell. This guarantees well-defined indexing without ambiguity.



(b) The relaxed setting in Theorem 7(ii) allows some balls to extend across adjacent grid cell boundaries, as long as they do not interfere with the specific cells being memorized at that step.

Figure 4: Grid-based Lattice Mapping.

Appendix D. Proofs for Section 3

D.1. Necessary Condition on Width for Robust Memorization

Proposition 8 *There exists a dataset $\mathcal{D} \in \mathcal{D}_{d,N,2}$ such that, for any $\rho \in (0,1)$, any neural network $f : \mathbb{R}^d \to \mathbb{R}$ that ρ-robustly memorizes $\mathcal{D}$ must have the first hidden layer width at least $\rho^2 \min\{N-1, d\}$.*

Proof To prove Theorem 4, we consider two cases based on the relationship between $N-1$ and d. In the first case, where $N-1 \leq d$, establishing the proposition requires that the first hidden layer has width at least $\rho^2(N-1)$. In the second case, where $N-1 > d$, the required width is at least $\rho^2 d$. For each case, we construct a dataset $\mathcal{D} \in \mathcal{D}_{d,N,2}$ such that any network that ρ-robustly memorizes $\mathcal{D}$ must have a first hidden layer of width no smaller than the corresponding bound.

Case I : $N-1 \leq d$**.** Let $\mathcal{D} = \{(\boldsymbol{e}_j, 2)\}_{j\in[N-1]} \cup \{(\mathbf{0}, 1)\}$. Then, $\mathcal{D}$ has separation constant $\epsilon_{\mathcal{D}} = 1/2$. Let f be a neural network that ρ-robust memorizes $\mathcal{D}$, and denote the width of its first hidden layer as m. Denote by $\boldsymbol{W} \in \mathbb{R}^{d\times m}$ the weight matrix of the first hidden layer of f. Assume for contradiction that $m < \rho^2(N-1)$.

Let $\mu = \rho\epsilon_{\mathcal{D}}$ denote the robustness radius. Then, the network f must distinguish every point in $B_2(\boldsymbol{e}_j, \mu)$ from every point in $B_\mu(\mathbf{0})$, for all $j \in [N-1]$. Therefore, for any $\boldsymbol{x} \in B_2(\boldsymbol{e}_j, \mu)$ and $\boldsymbol{x}' \in B_2(\mathbf{0}, \mu)$, we must have

$$\boldsymbol{W}\boldsymbol{x} \neq \boldsymbol{W}\boldsymbol{x}',$$

or equivalently, $\boldsymbol{x} - \boldsymbol{x}' \notin \mathrm{Null}(\boldsymbol{W})$, where $\mathrm{Null}(\cdot)$ denotes the null space of a given matrix. Note that

$$B_2(\boldsymbol{e}_j, \mu) - B_2(\mathbf{0}, \mu) := \{\boldsymbol{x} - \boldsymbol{x}' \mid \boldsymbol{x} \in B_2(\boldsymbol{e}_j, \mu) \text{ and } \boldsymbol{x}' \in B_2(\mathbf{0}, \mu)\} = B_2(\boldsymbol{e}_j, 2\mu).$$

Hence, it is necessary that $B_2(\boldsymbol{e}_j, 2\mu) \cap \mathrm{Null}(\boldsymbol{W}) = \emptyset$ for all $j \in [N-1]$, or equivalently,

$$\mathrm{dist}_2(\boldsymbol{e}_j, \mathrm{Null}(\boldsymbol{W})) \geq 2\mu \quad \text{for all } j \in [N-1]. \tag{4}$$

Since $\dim(\mathrm{Col}(\boldsymbol{W}^\top)) \leq m$, where $\mathrm{Col}(\cdot)$ denotes the column space of the given matrix, it follows that $\dim(\mathrm{Null}(\boldsymbol{W})) \geq d - m$. Using Theorem 11, we can upper bound the distance between the set $\{\boldsymbol{e}_j\}_{j\in[N-1]} \subseteq \mathbb{R}^d$ and any subspace of dimension $d-m$.

Let $Z \subseteq \mathrm{Null}(\boldsymbol{W})$ be a subspace such that $\dim(Z) = d - m$, and apply Theorem 11 with substitutions $d = d$, $t = N-1$, $k = d-m$ and $Z = Z$. The conditions of lemma, namely $t \leq d$ and $k \geq d - t$, are satisfied since $N - 1 \leq d$ and $m < \rho^2(N-1) \leq N-1$. Therefore, we obtain the bound

$$\min_{j\in[N-1]} \mathrm{dist}_2(\boldsymbol{e}_j, Z) \leq \sqrt{\frac{m}{N-1}}.$$

By combining the above inequality with Equation (4), we obtain

$$2\mu \leq \min_{j\in[N-1]} \mathrm{dist}_2(\boldsymbol{e}_j, \mathrm{Null}(\boldsymbol{W})) \overset{(a)}{\leq} \min_{j\in[N-1]} \mathrm{dist}_2(\boldsymbol{e}_j, Z) \leq \sqrt{\frac{m}{N-1}}, \tag{5}$$

where (a) follows from that $Z \subseteq \text{Null}(\boldsymbol{W})$. Since $\epsilon_{\mathcal{D}} = 1/2$, we have $2\mu = 2\rho\epsilon_{\mathcal{D}} = \rho$, so Equation (5) becomes

$$\rho \leq \sqrt{\frac{m}{N-1}}\,.$$

This implies that $m \geq \rho^2(N-1)$, contradicting the assumption $m < \rho^2(N-1)$. Therefore, the width requirement $m \geq \rho^2(N-1)$ is necessary. This concludes the statement for the case $N-1 \leq d$.

Case II : $N-1 > d$**.** We construct the first $d+1$ data points in the same manner as in Case I, using the construction for $N = d+1$. For the remaining $N-d-1$ data points, we set them sufficiently distant from the first $d+1$ data points to ensure that the separation constant remains $\epsilon_{\mathcal{D}} = 1/2$.

In particular, we set $\boldsymbol{x}_{d+2} = 2\boldsymbol{e}_1, \boldsymbol{x}_{d+3} = 3\boldsymbol{e}_1, \cdots, \boldsymbol{x}_N = (N-d)\boldsymbol{e}_1$ and assign $y_{d+2} = y_{d+3} = \cdots = y_N = 2$. Compared to the case $N = d+1$, this construction preserves $\epsilon_{\mathcal{D}}$ while adding more data points to memorize. Since the first $d+1$ data points are constructed as in the case $N = d+1$, the same lower bound applies. Specifically, by the result of Case I, any network that ρ-robustly memorizes this dataset mus have a first hidden layer of width at least $\rho^2(d+1-1) = \rho^2 d$. This concludes the arguement for the case $N-1 > d$.

Combining the result from the two cases $N-1 \leq d$ and $N-1 > d$ completes the proof of the theorem. ■

D.2. Necessary Condition on Parameters for Robust Memorization

For sufficiently large ρ, Gao et al. [7] and Li et al. [11] prove that, for any $\mathcal{D} \in \mathcal{D}_{d,N,C}$, if there exists $f \in \mathcal{F}_{d,P}$ that ρ-robustly memorizes $\mathcal{D}$, the number of parameters P should satisfy $P = \Omega(\sqrt{Nd})$. However, the authors do not characterize the range of ρ over which this lower bound remains valid.

In our work, we establish a lower bound that depends on ρ in the regime $\rho \leq \sqrt{1-1/d}$, which becomes $\sqrt{Nd}$ when $\rho = \sqrt{1-1/d}$. This implies that the existing lower bound $\sqrt{Nd}$ remains valid for $\rho \in [\sqrt{1-1/d}, 1)$. As a result, we obtain a lower bound that holds continuously from $\rho \approx 0$ up to $\rho \approx 1$, and thus interpolate the existing lower bound $\sqrt{Nd}$.

Proposition 9 *Let* $\rho \in \left(0, \sqrt{1-\frac{1}{d}}\right]$*. Suppose for any* $\mathcal{D} \in \mathcal{D}_{d,N,2}$*, there exists* $f \in \mathcal{F}_{d,P}$ *that* ρ*-robustly memorizes* $\mathcal{D}$*. Then, the number of parameters* P *must satisfy* $P = \Omega\left(\sqrt{N/(1-\rho^2)}\right)$*.*

Proof To prove the statement, we show that for any $\mathcal{D} \in \mathcal{D}_{d,N,2}$, if there exists a network $f \in \mathcal{F}_{d,P}$ that ρ-robustly memorize $\mathcal{D}$, then

$$\text{VC-dim}(\mathcal{F}_{d,P}) = \Omega\left(\frac{N}{1-\rho^2}\right).^{1} \tag{6}$$

Since $\text{VC-dim}(\mathcal{F}_{d,P}) = O(P^2)$, it follows that $P = \Omega(\sqrt{N/(1-\rho^2)})$.

1. We follow the definition of VC-dimension by Bartlett et al. [1]. Note that the VC-dimension of a real-valued function class is defined as the VC-dimension of $\text{sign}(\mathcal{F}) := \{\text{sign} \circ f \mid f \in \mathcal{F}\}$. Since we consider the label set $[2] = \{1, 2\}$ for robust memorization while the VC-dimension requires the label set $\{+1, -1\}$, we take an additional step of an affine transformation in the last step of the proof.

Let $k := \lfloor \frac{1}{1-\rho^2} \rfloor$. To establish the desired VC-dimension lower bound, it suffices to show that

$$\text{VC-dim}(\mathcal{F}_{d,P}) \geq k \cdot \lfloor \frac{N}{2} \rfloor.$$

This implies Equation (6), as desired. To this end, it suffices to construct $k \cdot \lfloor \frac{N}{2} \rfloor$ points in $\mathbb{R}^d$ that can be shattered by $\mathcal{F}_{d,P}$. These points are organized as $\lfloor \frac{N}{2} \rfloor$ groups, each consisting of k elements.

We begin by constructing the first group. Since $\rho \in \left(0, \sqrt{\frac{d-1}{d}}\right]$, we have $k = \lfloor \frac{1}{1-\rho^2} \rfloor \in (1, d]$. Define the first group $\mathcal{X}_1 := \{\boldsymbol{e}_j\}_{j=1}^k \subseteq \mathbb{R}^d$, consisting of the first k standard basis vectors in $\mathbb{R}^d$. The remaining $\lfloor \frac{N}{2} \rfloor - 1$ groups are constructed by translating $\mathcal{X}_1$. For each $l = 1, \cdots \lfloor \frac{N}{2} \rfloor$, define

$$\mathcal{X}_l := \boldsymbol{c}_l + \mathcal{X}_1 = \{\boldsymbol{c}_l + \boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{X}_1\},$$

where $\boldsymbol{c}_l := 2d^2(l-1) \cdot \boldsymbol{e}_1$ ensures that the groups are sufficiently distant from one another. Note that $\boldsymbol{c}_1 = \boldsymbol{0}$, so that $\mathcal{X}_1$ is consistent with the definition above. Now, define $\mathcal{X} := \cup_{l \in [\lfloor N/2 \rfloor]} \mathcal{X}_l$ as the union of all groups, comprising $k \times \lfloor \frac{N}{2} \rfloor$ points in total.

We claim that for any $\mathcal{D} \in \mathcal{D}_{d,N,2}$, if there exists a network $f \in \mathcal{F}_{d,P}$ that ρ-robustly memorizes $\mathcal{D}$, then the point set $\mathcal{X}$ is shattered by $\mathcal{F}_{d,P}$. To prove the claim, consider an arbitrary labeling $\mathcal{Y} = \{y_{l,j}\}_{l \in [\lfloor N/2 \rfloor], j \in [k]} \subset \{\pm 1\}$ of the points in $\mathcal{X}$, where each label $y_{l,j}$ corresponds to the point $\boldsymbol{x}_{l,j} := \boldsymbol{c}_l + \boldsymbol{e}_j \in \mathcal{X}$.

Given the labeling $\mathcal{Y}$, we construct $\mathcal{D} \in \mathcal{D}_{d,N,2}$ with labels in $\{1, 2\}$ such that any function $f \in \mathcal{F}_{d,P}$ that ρ-robustly memorizes $\mathcal{D}$ can be affinely transformed to $f' = 2f - 3 \in \mathcal{F}_{d,P}$, which satisfies $f'(\boldsymbol{x}_{l,j}) = y_{l,j} \in \{\pm 1\}$ for all $\boldsymbol{x}_{l,j} \in \mathcal{X}$. In other words, f' exactly memorizes the given labeling $\mathcal{Y}$ over $\mathcal{X}$, thereby showing that $\mathcal{X}$ is shatterd by $\mathcal{F}_{d,P}$. The affine transformation is necessary to match the $\{1, 2\}$-valued outputs of f with the $\{\pm 1\}$ labeling required for the shattering argument.

For each $l \in [\lfloor N/2 \rfloor]$, define the index sets

$$J_l^+ = \{j \in [k] \mid y_{l,j} = +1\}, \quad J_l^- = \{j \in [k] \mid y_{l,j} = -1\},$$

which partition the group-wise labeling $\{y_{l,j}\}_{j \in [k]} \subset \mathcal{Y}$ into positive and negative indices. We then define

$$\boldsymbol{x}_{2l-1} = \boldsymbol{c}_l + \sum_{j \in J_l^+} \boldsymbol{e}_j - \sum_{j \in J_l^-} \boldsymbol{e}_j,$$
$$\boldsymbol{x}_{2l} = \boldsymbol{c}_l + \sum_{j \in J_l^-} \boldsymbol{e}_j - \sum_{j \in J_l^+} \boldsymbol{e}_j.$$

Let $y_{2l-1} = 2, y_{2l} = 1$, and define the dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i \in [N]} \in \mathcal{D}_{d,N,2}$. Figure 5 illustrates the case $l = 1$ with $J_1^+ = \{1, 3\}$ and $J_1^- = \{2\}$, where the blue and red dots denote the points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively.

To analyze the separation constant $\epsilon_{\mathcal{D}}$, we consider the distance between pairs of points with different labels. Specifically, for each l, the two points $\boldsymbol{x}_{2l-1}$ and $\boldsymbol{x}_{2l}$ have opposite labels by construction. Consider their distance:

$$\|\boldsymbol{x}_{2l-1} - \boldsymbol{x}_{2l}\|_2 = \left\| 2 \left(\sum_{j \in J_l^+} \boldsymbol{e}_j - \sum_{j \in J_l^-} \boldsymbol{e}_j \right) \right\|_2 \overset{(a)}{=} 2\sqrt{k},$$

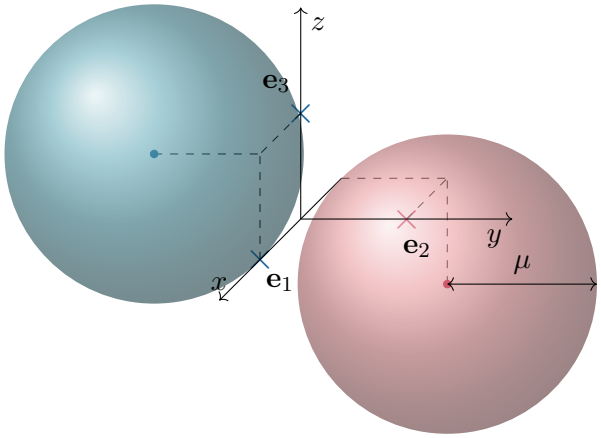


Figure 5: Reduction of Shattering to Robust Memorization. The cross marks refer to the points to be shattered, and the circular dots refer to the points for robust memorization. The centers of robustness balls change with respect to the labels of the points to be shattered.

where (a) holds since $J_l^+ \cap J_l^- = \emptyset$ and $J_l^+ \cup J_l^- = [k]$. Now, for $l \neq l'$, consider the distance between $\boldsymbol{x}_{2l-1}$ and $\boldsymbol{x}_{2l'}$, which again correspond to different labels. We have:

$$\begin{aligned}
\mathrm{dist}_2(\boldsymbol{x}_{2l-1}, \boldsymbol{x}_{2l'}) &\overset{(a)}{\geq} \mathrm{dist}_2(\boldsymbol{c}_l, \boldsymbol{c}_{l'}) - \mathrm{dist}_2(\boldsymbol{c}_l, \boldsymbol{x}_{2l-1}) - \mathrm{dist}_2(\boldsymbol{c}_{l'}, \boldsymbol{x}_{2l'}) \\
&\overset{(b)}{\geq} 2d^2 - \sqrt{k} - \sqrt{k} \\
&\overset{(c)}{\geq} 2d^2 - 2\sqrt{d} \\
&\overset{(d)}{\geq} 2\sqrt{d} \\
&\overset{(e)}{\geq} 2\sqrt{k},
\end{aligned}$$

where (a) follows from the triangle inequality, (b) uses $\mathrm{dist}_2(\boldsymbol{c}_l, \boldsymbol{x}_{2l-1}) = \mathrm{dist}_2(\boldsymbol{c}_{l'}, \boldsymbol{x}_{2l'}) = \sqrt{k}$, (c) and (e) use $k \leq d$, and (d) holds for all $d \geq 2$. Thus, we conclude that $\epsilon_{\mathcal{D}} \geq \sqrt{k}$.

Let $f \in \mathcal{F}_{d,P}$ be a function that ρ-robustly memorizes $\mathcal{D}$. We begin by deriving a lower bound on the robustness radius μ in order to verify that $f' = 2f - 3$ correctly memorizes the given labeling $\mathcal{Y}$ over $\mathcal{X}$. Define $\phi(t) := \sqrt{\frac{t-1}{t}}$. The function ϕ is strictly increasing for $t \geq 1$, and maps $[1, \infty)$ onto $[0, 1)$. Hence, it admits an inverse $\phi^{-1} : [0, 1) \to [1, \infty)$, defined as $\phi^{-1}(\rho) = \frac{1}{1-\rho^2}$. Therefore, we have

$$\rho = \phi(\phi^{-1}(\rho)) = \phi\left(\frac{1}{1-\rho^2}\right) \geq \phi\left(\lfloor \frac{1}{1-\rho^2} \rfloor\right) = \phi(k) = \sqrt{\frac{k-1}{k}}.$$

Given $\epsilon_{\mathcal{D}} \geq \sqrt{k}$ and $\rho \geq \sqrt{\frac{k-1}{k}}$, it follows that $\mu = \rho\epsilon_{\mathcal{D}} \geq \sqrt{k-1}$. Thus, any function f that ρ-robustly memorizes $\mathcal{D}$ must also memorize all points within an ℓ_2-ball of radius $\sqrt{k-1}$ centered at each point in $\mathcal{D}$.

Next, for $\boldsymbol{x}_{l,j} \in \mathcal{X}$ with positive label $y_{l,j} = +1$, we have

$$\begin{aligned}\|\boldsymbol{x}_{l,j} - \boldsymbol{x}_{2l-1}\|_2 &= \left\| (\boldsymbol{c}_l + \boldsymbol{e}_j) - (\boldsymbol{c}_l + \sum_{j' \in J_l^+} \boldsymbol{e}_{j'} - \sum_{j' \in J_l^-} \boldsymbol{e}_{j'}) \right\|_2 \\ &= \left\| \sum_{\substack{j' \in J_l^+ \\ j' \neq j}} \boldsymbol{e}_{j'} - \sum_{j' \in J_l^-} \boldsymbol{e}_{j'} \right\|_2 \\ &= \sqrt{k-1}.\end{aligned}$$

Now consider a sequence $\{\boldsymbol{z}_n\}_{n \in \mathbb{N}}$ such that $\boldsymbol{z}_n \to \boldsymbol{x}_{l,j}$ as $n \to \infty$ and

$$\|\boldsymbol{z}_n - \boldsymbol{x}_{2l-1}\|_2 < \sqrt{k-1} \quad \text{for all } n \in \mathbb{N}.$$

In particular, we take

$$\boldsymbol{z}_n := \frac{n-1}{n} \boldsymbol{x}_{l,j} + \frac{1}{n} \boldsymbol{x}_{2l-1},$$

which satisfies such properties. Then, $\boldsymbol{z}_n \in \mathcal{B}(\boldsymbol{x}_{2l-1}, \mu)$ for all n, and by robustness of f, $f(\boldsymbol{z}_n) = f(\boldsymbol{x}_{2l-1}) = 2$. By continuity of f, we have

$$f(\boldsymbol{x}_{l,j}) = f(\lim_{n \to \infty} \boldsymbol{z}_n) = \lim_{n \to \infty} f(\boldsymbol{z}_n) = \lim_{n \to \infty} 2 = 2.$$

Similarly, for $\boldsymbol{x}_{l,j} \in \mathcal{X}$ with negative label $y_{l,j} = -1$, we have $\|\boldsymbol{x}_{l,j} - \boldsymbol{x}_{2l}\|_2 = \sqrt{k-1}$, so that $f(\boldsymbol{x}_{l,j}) = 1$.

Since we can adjust the weight and the bias of the last hidden layer, $\mathcal{F}_{d,P}$ is closed under affine transformation; that is, $af + b \in \mathcal{F}_{d,P}$ whenever $f \in \mathcal{F}_{d,P}$. In particular, $f' := 2f - 3 \in \mathcal{F}_{d,P}$. This f' satisfies $f'(\boldsymbol{x}_{l,j}) = 2f(\boldsymbol{x}_{l,j}) - 3 = 2 \cdot 2 - 3 = +1$ whenever $y_{l,j} = +1$ and $f'(\boldsymbol{x}_{l,j}) = 2f(\boldsymbol{x}_{l,j}) - 3 = 2 \cdot 1 - 3 = -1$ whenever $y_{l,j} = -1$. Thus, $\operatorname{sign} \circ f'$ perfectly classifies $\mathcal{X}$ according to the given labeling $\mathcal{Y}$. Since the labeling $f' \in \mathcal{F}_{d,P}$ was arbitrary, it follows that $\mathcal{F}_{d,P}$ shatters $\mathcal{X}$, completing the proof of the theorem.

■

D.3. Lemmas for Appendix D

The following lemma upper bounds the ℓ_2-distance between the standard basis and any subspace of a given dimension.

Lemma 10 *Let $\{\boldsymbol{e}_j\}_{j \in [d]} \subseteq \mathbb{R}^d$ denote the standard basis of $\mathbb{R}^d$. Then, for any k-dimensional subspace $Z \subseteq \mathbb{R}^d$,*

$$\max_{j \in [d]} \|\mathrm{Proj}_Z(\boldsymbol{e}_j)\|_2 \geq \sqrt{\frac{k}{d}}.$$

In particular,

$$\min_{j \in [d]} \mathrm{dist}_2(\boldsymbol{e}_j, Z) \leq \sqrt{\frac{d-k}{d}}$$

Proof Let $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_k\} \subseteq \mathbb{R}^d$ be an orthonormal basis of Z, and denote each $\boldsymbol{u}_j = (u_{j1}, u_{j2}, \cdots, u_{jd})^\top$. Let $U \in \mathbb{R}^{d\times k}$ be the matrix whose colums are $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k$, so that

$$U = \begin{bmatrix} | & | & & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_k \\ | & | & & | \end{bmatrix}.$$

Then the projection matrix P onto Z is given by

$$P = U(U^\top U)^{-1}U^\top = UU^\top = \sum_{l=1}^{k} \boldsymbol{u}_l \boldsymbol{u}_l^\top \in \mathbb{R}^{d\times d}.$$

Now, for each standard basis vector $\boldsymbol{e}_j$, the squared norm of its projection onto Z is:

$$\|P\boldsymbol{e}_j\|_2^2 = \left\|\sum_{l=1}^{k} \boldsymbol{u}_l \boldsymbol{u}_l^\top \boldsymbol{e}_j\right\|_2^2 = \left\|\sum_{l=1}^{k} u_{lj}\boldsymbol{u}_l\right\|_2^2 = \sum_{l=1}^{k}(u_{lj})^2,$$

where the last equality holds as $\boldsymbol{u}_l$ are orthonormal. Moreover,

$$\max_{j\in[d]} \|P\boldsymbol{e}_j\|_2^2 \geq \frac{1}{d}\sum_{j\in[d]} \|P\boldsymbol{e}_j\|_2^2 = \frac{1}{d}\sum_{j\in[d]}\sum_{l=1}^{k}(u_{lj})^2 = \frac{1}{d}\sum_{l=1}^{k}\sum_{j\in[d]}(u_{lj})^2 = \frac{1}{d}\sum_{l=1}^{k} 1 = \frac{k}{d}.$$

This proves the first statement of the lemma. To prove the second statement, observe that for any $\boldsymbol{v} \in \mathbb{R}^d$, we can write

$$\boldsymbol{v} = \mathrm{Proj}_Z(\boldsymbol{v}) + \mathrm{Proj}_{Z^\perp}(\boldsymbol{v}),$$

so that $\|\boldsymbol{v}\|_2^2 = \|\mathrm{Proj}_Z(\boldsymbol{v})\|_2^2 + \|\mathrm{Proj}_{Z^\perp}(\boldsymbol{v})\|_2^2$. Noticing $\mathrm{dist}_2(\boldsymbol{v}, Z) = \|\mathrm{Proj}_{Z^\perp}(\boldsymbol{v})\|_2$ together with the first statement,

$$\min_{j\in[d]} \mathrm{dist}_2(\boldsymbol{e}_j, Z) = \min_{j\in[d]} \|\mathrm{Proj}_{Z^\perp}(\boldsymbol{e}_j)\|_2 = \max_{j\in[d]} \sqrt{1 - \|\mathrm{Proj}_Z(\boldsymbol{e}_j)\|_2^2} \leq \sqrt{1 - \frac{k}{d}} = \sqrt{\frac{d-k}{d}},$$

concludes the second statement. ■

The next lemma generalizes Theorem 10 to the case where we consider only the distance to a subset of the standard basis, instead of the whole standard basis.

Lemma 11 *Let $1 \leq t \leq d$, and let $\{\boldsymbol{e}_j\}_{j\in[t]} \subseteq \mathbb{R}^d$ denote the first t standard basis vectors. Then, for any k-dimensional subspace $Z \subseteq \mathbb{R}^d$ with $k \geq d - t$, we have*

$$\max_{j\in[t]} \|\mathrm{Proj}_Z(\boldsymbol{e}_j)\|_2 \geq \sqrt{\frac{k-(d-t)}{t}}.$$

In particular,

$$\min_{j\in[t]} \mathrm{dist}_2(\boldsymbol{e}_j, Z) \leq \sqrt{\frac{d-k}{t}}.$$

Proof Let $Q = [\boldsymbol{e}_1 \boldsymbol{e}_2 \cdots \boldsymbol{e}_t]^\top \in \mathbb{R}^{t \times d}$. Then, we have the orthogonal decomposition:

$$\mathbb{R}^d = \mathrm{Col}(Q^\top) \oplus \mathrm{Null}(Q) = (Z \cap \mathrm{Col}(Q^\top)) \oplus (Z^\perp \cap \mathrm{Col}(Q^\top)) \oplus \mathrm{Null}(Q)$$

By taking dimensions,

$$\begin{aligned}
\dim(Z \cap \mathrm{Col}(Q^\top)) &= \dim(\mathbb{R}^d) - \dim(Z^\perp \cap \mathrm{Col}(Q^\top)) - \dim(\mathrm{Null}(Q)) \\
&\geq \dim(\mathbb{R}^d) - \dim(Z^\perp) - \dim(\mathrm{Null}(Q)) \\
&= d - (d-k) - (d-t) \\
&= k - (d-t).
\end{aligned}$$

Now, consider the restriction of $\mathbb{R}^d$ to $\mathbb{R}^t$ by the linear map

$$\phi : \mathrm{span}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_t\} \subset \mathbb{R}^d \to \mathbb{R}^t, \quad \phi\left(\sum_{i=1}^t a_i \boldsymbol{e}_j\right) = \begin{bmatrix} a_1 \\ \vdots \\ a_t \end{bmatrix}.$$

Since $\mathrm{Col}(Q^\top) = \mathrm{span}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_t\}$, the projection satisfies:

$$\max_{j \in [t]} \left\| \mathrm{Proj}_{Z \cap \mathrm{Col}(Q^\top)}(\boldsymbol{e}_j) \right\|_2 = \max_{j \in [t]} \left\| \mathrm{Proj}_{\phi(Z \cap \mathrm{Col}(Q^\top))}(\phi(\boldsymbol{e}_j)) \right\|_2.$$

By applying Theorem 10 with the restricted space $\mathbb{R}^t$, we obtain

$$\max_{j \in [t]} \left\| \mathrm{Proj}_{Z \cap \mathrm{Col}(Q^\top)}(\boldsymbol{e}_j) \right\|_2 \geq \sqrt{\frac{k-(d-t)}{t}}.$$

Since $Z \supseteq Z \cap \mathrm{Col}(Q^\top)$, it follows that

$$\max_{j \in [t]} \left\| \mathrm{Proj}_{Z}(\boldsymbol{e}_j) \right\|_2 \geq \max_{j \in [t]} \left\| \mathrm{Proj}_{Z \cap \mathrm{Col}(Q^\top)}(\boldsymbol{e}_j) \right\|_2 \geq \sqrt{\frac{k-(d-t)}{t}}.$$

This proves the first statement. To prove the second statement, for any $\boldsymbol{v} \in \mathbb{R}^d$, decompose $\boldsymbol{v}$ as

$$\boldsymbol{v} = \mathrm{Proj}_Z(\boldsymbol{v}) + \mathrm{Proj}_{Z^\perp}(\boldsymbol{v}),$$

and note that $\|\boldsymbol{v}\|_2^2 = \|\mathrm{Proj}_Z(\boldsymbol{v})\|_2^2 + \|\mathrm{Proj}_{Z^\perp}(\boldsymbol{v})\|_2^2$. Using $\mathrm{dist}_2(\boldsymbol{v}, Z) = \|\mathrm{Proj}_{Z^\perp}(\boldsymbol{v})\|_2$ together with the first statement, we have

$$\begin{aligned}
\min_{j \in [t]} \mathrm{dist}_2(\boldsymbol{e}_j, Z) &= \min_{j \in [t]} \|\mathrm{Proj}_{Z^\perp}(\boldsymbol{e}_j)\|_2 \\
&= \max_{j \in [t]} \sqrt{1 - \|\mathrm{Proj}_Z(\boldsymbol{e}_j)\|_2^2} \\
&\leq \sqrt{1 - \frac{k-(d-t)}{t}} \\
&= \sqrt{\frac{d-k}{t}},
\end{aligned}$$

concludes the second statement. ■

D.4. Explicit Proof of Theorem 3

While we have mentioned how Theorems 4 and 5 can imply Theorem 3 in Section 3, we elaborate on a more detailed proof here.

Theorem 3 *Let $\rho \in (0,1)$. Suppose for any $\mathcal{D} \in \mathcal{D}_{d,N,2}$, there exists a neural network $f \in \mathcal{F}_{d,P}$ that can ρ-robustly memorize $\mathcal{D}$. Then, the number of parameters P must satisfy*

$$P = \Omega\Big((\rho^2 \min\{N,d\}+1)d + \min\Big\{\frac{1}{\sqrt{1-\rho^2}}, \sqrt{d}\Big\}\sqrt{N}\Big).$$

Proof Let $f \in \mathcal{F}_{d,P}$ be a neural network that ρ-robustly memorizes $\mathcal{D}$ and let m denote the width of first hidden layer. We first note the trivial lower bound that $m \geq 1$. From Theorem 4, we also have the lower bound $m \geq \rho^2 \min\{N-1,d\}$. Thus,

$$m \geq \max\{\rho^2 \min\{N-1,d\}, 1\} \geq \frac{1}{2}(\rho^2 \min\{N-1,d\}+1).$$

Since we count all parameters as Equation (1), the number of parameters in the first layer is $(d+1)m$. Therefore,

$$P \geq (d+1)\cdot m \geq (d+1)\cdot\frac{1}{2}(\rho^2 \min\{N-1,d\}+1) = \Omega(d(\rho^2 \min\{N,d\}+1)).$$

In addition, For $\rho \in \Big(0, \sqrt{1-\frac{1}{d}}\Big]$, by Theorem 5, we get the lower bound of parameters

$$P = \Omega\left(\sqrt{\frac{N}{1-\rho^2}}\right).$$

Note that the inequality $\frac{1}{\sqrt{1-\rho^2}} \leq \sqrt{d}$ holds, so we may write

$$\min\{\frac{1}{\sqrt{1-\rho^2}}, \sqrt{d}\} = \frac{1}{\sqrt{1-\rho^2}}.$$

For $\rho \in \Big(\sqrt{1-\frac{1}{d}}, 1\Big)$, observe that $\frac{1}{\sqrt{1-\rho^2}} > \sqrt{d}$, and since $\rho = \sqrt{1-\frac{1}{d}}$ yields $\sqrt{\frac{N}{1-\rho^2}} = \sqrt{Nd}$, we also require $\sqrt{Nd}$ parameters in this regime. Thus, by combining both regimes, we obtain:

$$P = \Omega\left(\min\{\frac{1}{\sqrt{1-\rho^2}}, \sqrt{d}\}\sqrt{N}\right).$$

By combining the bounds from Theorem 4 and Theorem 5, we conclude:

$$\begin{aligned} P =& \Omega\left(\max\left\{(\rho^2 \min\{N,d\}+1)d, \min\{\frac{1}{\sqrt{1-\rho^2}, \sqrt{d}}\}\sqrt{N}\right\}\right) \\ =& \Omega\left((\rho^2 \min\{N,d\}+1)d + \min\{\frac{1}{\sqrt{1-\rho^2}, \sqrt{d}}\}\sqrt{N}\right). \end{aligned}$$

■

Appendix E. Proofs for Section 4

In this section, we prove Theorem 7. An extension of Theorem 7 to ℓ_p-norm is provided in Appendix F.2.

Theorem 7 *For any dataset $\mathcal{D} \in \mathcal{D}_{d,N,C}$ and $\eta \in (0,1)$, the following statements hold:*

(i) *If $\rho \in \left(0, \frac{1}{5N\sqrt{d}}\right]$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(\sqrt{N})$ that ρ-robustly memorizes $\mathcal{D}$.*

(ii) *If $\rho \in \left(\frac{1}{5N\sqrt{d}}, \frac{1}{5\sqrt{d}}\right]$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(Nd^{\frac{1}{4}}\rho^{\frac{1}{2}})$ that ρ-robustly memorizes $\mathcal{D}$ with error at most η.*

(iii) *If $\rho \in \left(\frac{1}{5\sqrt{d}}, 1\right)$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(Nd^2\rho^4)$ that ρ-robustly memorizes $\mathcal{D}$.*

To prove Theorem 7, we decompose it into three theorems (Theorems 12, 14 and 22), each corresponding to one of the cases in the statement. Their proofs are provided in Appendices E.1 to E.3, respectively.

E.1. Sufficient Condition for Robust Memorization with Small Robustness Radius

Theorem 12 *Let $\rho \in \left(0, \frac{1}{5N\sqrt{d}}\right]$. For any dataset $\mathcal{D} \in \mathcal{D}_{d,N,C}$, there exists a neural network $f \in \mathcal{F}_{d,P}$ that ρ-robustly memorizes $\mathcal{D}$, where the number of parameters satisfies $P = \tilde{O}(\sqrt{N})$.*

Proof For given ρ and $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i\in[N]} \in \mathcal{D}_{d,N,C}$, we construct $f \in \mathcal{F}_{d,P}$ that satisfies the stated condition. The construction consists of four steps. In each step, we construct a function that can be implemented via a neural network so that when all four functions are composed together, it forms a ρ-robust memorizer of $\mathcal{D}$.

Stage I (Projection onto log-scale Dimension and Scaling via the First Layer Hidden Weight Matrix). In the first step, we map the data into $m := \max\{\lceil 600 \log N \rceil, \lceil 10 \log d \rceil\}$ dimension via linear transformation. We construct the linear mapping by dividing the cases into $d < 600 \log N$ or $d \geq 600 \log N$.

For the case $d < 600 \log N$, we have $d < m$. We consider the natural (linear) embedding from $\mathbb{R}^d$ to $\mathbb{R}^m$ defined by $(x_1, \cdots, x_d) \overset{\phi}{\mapsto} (x_1, \cdots, x_d, 0, \cdots, 0)$. The ϕ is 1-Lipchitz, and $\mathcal{D}' := \{(\phi(\boldsymbol{x}_i), y_i)\}_{i\in[N]} \in \mathcal{D}_{m,N,C}$ satisfies $\epsilon_{\mathcal{D}'} \geq \epsilon_{\mathcal{D}}$.

Otherwise, for the case $d \geq 600 \log N$, we first confirm that $m \leq d$. First, $600 \log N \leq d$ implies

$$\lceil 600 \log N \rceil \leq d. \tag{7}$$

Additionally, as $N \geq 2$, we have $d \geq 600 \log N \geq 600 \log 2 \geq 400$. By Theorem 23, this implies $10 \log d \leq d$ and therefore

$$\lceil 10 \log d \rceil \leq d. \tag{8}$$

By Equations (7) and (8), we have $m \leq d$.

By Theorem 28, there exists 1-Lipschitz linear mapping $\phi : \mathbb{R}^d \to \mathbb{R}^m$ and $\beta > 0$ such that $\mathcal{D}' := \{(\phi(\boldsymbol{x}_i), y_i)\}_{i\in[N]} \in \mathcal{D}_{m,N,C}$ satisfies $\epsilon_{\mathcal{D}'} \geq \frac{4}{5}\beta\epsilon_{\mathcal{D}}$. As $m \geq 10 \log d$, the inequality $\beta \geq \frac{1}{2}\sqrt{\frac{m}{d}}$ is satisfied by Theorem 28. Therefore, $\epsilon_{\mathcal{D}'} \geq \frac{4}{5}\beta\epsilon_{\mathcal{D}} \geq \frac{2}{5}\sqrt{\frac{m}{d}}\epsilon_{\mathcal{D}}$.

In both cases, we have 1-Lipschitz linear map ϕ such that $\mathcal{D}' = \{(\phi(\boldsymbol{x}_i), y_i)\}_{i\in[N]}$ has separation

$$\epsilon_{\mathcal{D}'} \geq \frac{2}{5}\sqrt{\frac{m}{d}}\epsilon_{\mathcal{D}}. \tag{9}$$

. We use this ϕ to construct the first hidden layer. Set the first hidden layer matrix as the matrix $\boldsymbol{W}$ corresponding to $\frac{5}{4}\cdot\frac{\sqrt{d}}{\epsilon_{\mathcal{D}}}\phi$ under the standard basis of $\mathbb{R}^d$ and $\mathbb{R}^m$. Set the first hidden layer bias $\boldsymbol{b}$ so that

$$\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \geq \boldsymbol{0} \text{ for all } \boldsymbol{x} \in \mathcal{B}_2(\boldsymbol{x}_i, \mu) \text{ and for all } i \in [N], \tag{10}$$

where the comparison between two vectors is element-wise. Define $f_1 : \mathbb{R}^d \to \mathbb{R}^m$ as $f_1(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$.

We claim that for $\mathcal{D}'' = \{(\mathrm{ReLU}(f_1(\boldsymbol{x}_i)), y_i)\}_{i\in[N]}$, we have (i) $\epsilon_{\mathcal{D}''} \geq \sqrt{m}/2$ and (ii) for $\rho'' = \frac{1}{2N\sqrt{m}}$, whenever $g \in \mathcal{F}_{m,P}$ is a ρ''-robust memorizer of $\mathcal{D}''$, then $g \circ \mathrm{ReLU} \circ f_1$ is a ρ-robust memorizer of $\mathcal{D}$. For any $y_i \neq y_j$, we have

$$\begin{aligned}
\|\mathrm{ReLU}(f_1(\boldsymbol{x}_i)) - \mathrm{ReLU}(f_1(\boldsymbol{x}_j))\| &\overset{(a)}{=} \|f_1(\boldsymbol{x}_i) - f_1(\boldsymbol{x}_j)\|_2 \\
&= \|(\boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{b}) - (\boldsymbol{W}\boldsymbol{x}_j + \boldsymbol{b})\|_2 \\
&= \|\boldsymbol{W}(\boldsymbol{x}_i) - \boldsymbol{W}(\boldsymbol{x}_j)\|_2 \\
&\overset{(b)}{=} \frac{5}{4}\cdot\frac{\sqrt{d}}{\epsilon_{\mathcal{D}}}\|\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)\|_2 \\
&\overset{(c)}{\geq} \frac{5}{4}\cdot\frac{\sqrt{d}}{\epsilon_{\mathcal{D}}}\cdot 2\epsilon_{\mathcal{D}'} \\
&\overset{(d)}{\geq} \frac{5}{4}\cdot\frac{\sqrt{d}}{\epsilon_{\mathcal{D}}}\times 2\cdot\frac{2}{5}\sqrt{\frac{m}{d}}\epsilon_{\mathcal{D}} \\
&= \sqrt{m},
\end{aligned}$$

where (a) is because $f_1(\boldsymbol{x}_i) \geq \boldsymbol{0}$ for all $i \in [N]$, (b) is by the definition of $\boldsymbol{W}$, (c) is by the definition of $\mathcal{D}'$, and (d) is by Equation (9). This proves the first claim $\epsilon_{\mathcal{D}''} \geq \sqrt{m}/2$. To prove the second claim, let $\mu := \rho\epsilon_{\mathcal{D}}$ and $\mu'' := \rho''\epsilon_{\mathcal{D}''}$. Then,

$$\begin{aligned}
\mathrm{ReLU}(f_1(\mathcal{B}_2(\boldsymbol{x}_i, \mu))) &\overset{(a)}{=} f_1(\mathcal{B}_2(\boldsymbol{x}_i, \mu)) \\
&\overset{(b)}{=} \mathcal{B}_2(f_1(\boldsymbol{x}_i), \frac{5}{4}\cdot\frac{\sqrt{d}}{\epsilon_{\mathcal{D}}}\times\mu) \\
&\overset{(c)}{=} \mathcal{B}_2(f_1(\boldsymbol{x}_i), \frac{5}{4}\cdot\sqrt{d}\rho) \\
&\overset{(d)}{\subseteq} \mathcal{B}_2(f_1(\boldsymbol{x}_i), \frac{1}{4N}) \\
&\overset{(e)}{\subseteq} \mathcal{B}_2(f_1(\boldsymbol{x}_i), \rho''\epsilon_{\mathcal{D}''}) \\
&\overset{(f)}{\subseteq} \mathcal{B}_2(\mathrm{ReLU}(f_1(\boldsymbol{x}_i)), \rho''\epsilon_{\mathcal{D}''}),
\end{aligned}$$

where (a),(f) are by Equation (10), (b) is because f_1 is $\frac{5}{4} \cdot \frac{\sqrt{d}}{\epsilon}$-Lipschitz, (c) use $\mu = \rho\epsilon_{\mathcal{D}}$, (d) use $\rho \leq \frac{1}{5N\sqrt{d}}$, and (e) is because $\rho''\epsilon_{\mathcal{D}''} = \frac{\epsilon_{\mathcal{D}''}}{2N\sqrt{m}} \geq \frac{1}{4N}$, as $\epsilon_{\mathcal{D}''} \geq \sqrt{m}/2$.

Hence, g memorizing the robustness ball $\mathcal{B}_2(\mathrm{ReLU}(f_1(\boldsymbol{x}_i)), \rho''\epsilon_{\mathcal{D}''})$ on projected space leads to $g \circ \mathrm{ReLU} \circ f_1$ memorizing the robustness ball for $\mathcal{D}$. In other words, whenever g is a ρ''-robust memorizer of $\mathcal{D}''$, then $g \circ \mathrm{ReLU} \circ f_1$ is a ρ-robust memorizer of $\mathcal{D}$. With $\rho'' = \frac{1}{N\sqrt{m}}$, Stage II to IV aims to find a ρ''-robust memorizer g of $\mathcal{D}''$.

Stage II (Translation for Distancing from Lattice via the Bias) For simplicity of the notation, let us denote $\boldsymbol{z}_i = \mathrm{ReLU}(f_1(\boldsymbol{x}_i))$ so that $\mathcal{D}'' = \{(\boldsymbol{z}_i, y_i)\}_{i\in[N]}$. Recall that $\epsilon_{\mathcal{D}''} = \frac{\sqrt{m}}{2}$ and $\rho'' = \frac{1}{2N\sqrt{m}}$, the robustness radius is $\mu'' = \rho''\epsilon_{\mathcal{D}''} = \frac{1}{4N}$.

By applying Theorem 24 to $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N$, there exist a translation vector $\boldsymbol{b}_2 = (b_{21}, \cdots, b_{2m}) \in \mathbb{R}^m$ such that

$$\mathrm{dist}(z_{i,j} - b_j, \mathbb{Z}) \geq \frac{1}{2N}, \quad \forall i \in [N], j \in [d], \tag{11}$$

i.e., the translated points $\{\boldsymbol{z}_i - \boldsymbol{b}_2\}_{i\in[N]}$ are coordinate-wise far from the integer lattice. Moreover, by additional translation $\boldsymbol{z}_i$ (by some natural number, coordinate-wise), we can ensure all coordinates are positive while keeping the property Equation (11). Hence, we may assume without loss of generality $\boldsymbol{b}_2$ also has the property

$$\boldsymbol{z}_i - \boldsymbol{b}_2 \geq \boldsymbol{0} \text{ for all } i \in [N] \tag{12}$$

Let us denote $\mathcal{D}''' = \{(\boldsymbol{z}'_i, y_i)\}_{i\in[N]}$, where $\boldsymbol{z}'_i := \boldsymbol{z}_i - \boldsymbol{b}$. We have $\epsilon_{\mathcal{D}'''} = \epsilon_{\mathcal{D}''}$. For $\rho''' := \rho'' = \frac{1}{2N\sqrt{m}}$, we have the robustness radius $\mu''' := \mu'' = \frac{1}{4N}$.

Upon the two layers constructed from stages I, II, it suffices to construct a network that ρ'''-robustly memorizes $\mathcal{D}'''$. Note that the robustness balls after stage II are not affected when passing the ReLU, by Equations (11) and (12).

Stage III (Grid Indexing) By Equation (11), each $\boldsymbol{z}'_i \in \mathbb{R}^m$ is at least $2\mu'''$ distant away from any lattice hyperplane $H_{z,j} := \{\boldsymbol{z} \in \mathbb{R}^m \mid z_j = z\}$ for with any $j \in [m]$ and $z \in \mathbb{Z}$. Thus, each robustness ball of $\mathcal{D}'''$ lies completely within a single integer lattice (or unit grid) of the form $\prod_{j=1}^m [n_j, n_j + 1)$, where $(n_1, \cdots, n_m) \in \mathbb{Z}^m$.

Since $\epsilon_{\mathcal{D}'''} \geq \sqrt{m}/2$, we have $\left\|\boldsymbol{z}'_i - \boldsymbol{z}'_{i'}\right\|_2 \geq \sqrt{m}$ for all i, i' with $y_i \neq y_{i'}$. As $\sup\{\|\boldsymbol{z} - \boldsymbol{z}'\|_2 \mid \boldsymbol{z}, \boldsymbol{z}' \in \prod_{j=1}^m [n_j, n_j + 1)\} = \sqrt{m}$, no two data-points with data points lie within the same integer lattice. Since each μ'''-ball lies within a single grid, we conclude that no two μ'''-ball with different labels lie within the same grid.

We define $R = \max_{i\in[N]} \|\boldsymbol{z}'_i\|_\infty$ $(= \max_{i\in[N], j\in[m]}(z'_{i,j}))$. Our goal in this step is to construct $\mathrm{Flatten}$ mapping defined as

$$\mathrm{Flatten}(x) := R^{m-1}\lfloor x_1 \rfloor + R^{m-2}\lfloor x_2 \rfloor + \cdots + \lfloor x_m \rfloor.$$

This maps a grid $\prod_{j=1}^m [n_j, n_{j+1})$ onto $\sum_{j=1}^m R^{j-1} n_j$.

Since $\mathrm{Flatten}$ is discontinuous in nature, we construct $\overline{\mathrm{Flatten}}$, which is continuous and matches $\mathrm{Flatten}$ in the region of our interest. By applying Theorem 25 to $\gamma = \frac{1}{4N}$ and $n = \lceil \log_2 R \rceil$, we obtain the network $\overline{\mathrm{Floor}} := \overline{\mathrm{Floor}}_{\lceil \log_2 R \rceil}$ with $O(\log_2 R)$ parameters such that

$$\overline{\mathrm{Floor}}(x) = \lfloor x \rfloor \quad \forall \boldsymbol{x} \in \mathcal{B}_2(\boldsymbol{x}_i, \mu').$$

We define our network $\overline{\text{Flatten}}$ as

$$\overline{\text{Flatten}}(x) = R^{m-1}\overline{\text{Floor}}(x_1) + \cdots + \overline{\text{Floor}}(x_d).$$

We can construct $\overline{\text{Flatten}}$ with $O(m \log_2 R)$ parameters. $\overline{\text{Flatten}}$ maps each robustness ball $\mathcal{B}_2(\boldsymbol{z}_i', \mu''')$ to $\overline{\text{Flatten}} = \text{Flatten}(\boldsymbol{z}_i) \in \mathbb{Z}$. Let us denote $m_i := \text{Flatten}(\boldsymbol{z}_i)$. Then, $m_i \in \mathbb{Z} \in [0, R^{m+1}]$ for all $i \in [N]$.

Stage IV (Memorization) Finally, it remains to memorize N points $\{(m_i, y_i)\}_{i=1}^N \subset \mathbb{N} \times [C]$. Since multiple balls with the same label may correspond to the same grid index, it is possible that for some $i \neq j$ with $y_i = y_j$, we have $m_i = m_j$. Let $N' \leq N$ denote the number of distinct pairs (m_i, y_i). It suffices to memorize only these N' distinct data points in $\mathbb{R}^m$.

We apply Theorem 13 by Vardi et al. [16], using $r = R^m$ since $m_i = \text{Flatten}(\boldsymbol{x}_i) \leq R^{m+1}$ to construct f_{mem} with $\tilde{O}(\sqrt{M} \cdot \log(5R^m N^2 \epsilon^{-1}\sqrt{\pi m})) = \tilde{O}(m\sqrt{M}) = \tilde{O}(\log N \sqrt{M}) = \tilde{O}(\sqrt{M}) = \tilde{O}(\sqrt{N})$ parameters such that $f_{mem}(m_i) = y_i$.

The final network is $f : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$f(\boldsymbol{x}) = f_{mem} \circ \text{ReLU} \circ \overline{\text{Flatten}} \circ \text{ReLU} \circ (\text{ReLU}(f_1(\boldsymbol{x})) - \boldsymbol{b}_2)$$

The total construction requires $\tilde{O}(md + m + m \log_2 R + \sqrt{N}) = \tilde{O}(d + \sqrt{N})$ parameters. ■

The following is the classical memorization upper bound of parameters used in the proof of Theorem 12

Theorem 13 (Classical Memorization, Theorem 3.1 from Vardi et al. [16]) *Let $N, d, C \in \mathbb{N}$, and $r, \epsilon > 0$, and let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \in \mathbb{R}^d \times [C]$ be a set of N labeled samples with $\|\boldsymbol{x}_i\| \leq r$ for every i and $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \geq 2\epsilon$ for every $i \neq j$. Denote $R := 5rN^2\epsilon^{-1}\sqrt{\pi d}$. Then, there exists a neural network $F : \mathbb{R}^d \to \mathbb{R}$ with width 12 and depth*

$$O\left(\sqrt{N \log N} + \sqrt{\frac{N}{\log N}} \cdot \max\{\log(R), \log(C)\}\right),$$

such that $F(\boldsymbol{x}_i) = y_i$ for every $i \in [N]$.

E.2. Sufficient Condition for Near-Perfect Robust Memorization with Moderate Robustness Radius

Theorem 14 *Let $\rho \in \left(0, \frac{1}{5\sqrt{d}}\right]$, and $\eta \in (0, 1)$. For any dataset $\mathcal{D} \in \mathcal{D}_{d,N,C}$, there exists a neural network $f \in \mathcal{F}_{d,P}$ that ρ-robustly memorizes $\mathcal{D}$ with error at most η, where the number of parameters satisfies $P = \tilde{O}(Nd^{\frac{1}{4}}\rho^{\frac{1}{2}})$.*

Proof Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i \in [N]} \in \mathcal{D}_{d,N,C}$, and suppose the robustness ratio satisfies $\rho < \frac{1}{5\sqrt{d}}$. We aim to construct a network f that ρ-robustly memorizes $\mathcal{D}$ with $\tilde{O}(Nd^{\frac{1}{4}}\rho^{\frac{1}{2}})$ parameters.

Stage I (Projecting the Balls onto $\log N$-scale Dimensional Space): If $d > \lceil 600 \log N \rceil$, we begin by projecting the data points and their balls from $\mathbb{R}^d$ to $\mathbb{R}^m$ where $m = \lceil 600 \log N \rceil$. By applying Theorem 28, we can get 1-Lipschitz linear mapping $\phi : \mathbb{R}^d \to \mathbb{R}^m$ such that

$$\|\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)\|_2 \geq \frac{4}{5}\sqrt{\frac{m}{d}}\epsilon_{\mathcal{D}} \quad \forall y_i \neq y_j.$$

By the 1-Lipschitzness of ϕ, it holds that

$$\phi\left(\mathcal{B}_2(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D}})\right) \subset \mathcal{B}_2\left(\phi(\boldsymbol{x}_i), \rho\epsilon_{\mathcal{D}}\right).$$

We define the projection network $f_{\text{proj}} := \phi(\boldsymbol{x})$, which can be implemented as a linear layer with $dm = O(d \log N)$ parameters. Let $\boldsymbol{x}' := f_{\text{proj}}(\boldsymbol{x})$. We now work with the dataset $(\boldsymbol{x}'_i, y_i) \subset \mathbb{R}^m \times [C]$ with new separation $\frac{2}{5}\sqrt{\frac{m}{d}}\epsilon_{\mathcal{D}}$. We denote the new robustness ratio as

$$\rho' := \frac{5}{2}\sqrt{\frac{d}{m}}\rho < \frac{1}{2\sqrt{m}}.$$

Stage II (Memorizing N^α Points at Each Layer): We group N data points to $\lceil N^{1-\alpha} \rceil$ groups with index $\{I_j\}_{j=1}^{N^{1-\alpha}}$, each with $|I_j| \le \lfloor N^\alpha \rfloor + 1$.

For each $j \in [N^{1-\alpha}]$, we apply Theorem 21 to the group $\{(\boldsymbol{x}'_i, y_i)\}_{i \in I_j}$ using failure probability $\frac{\eta}{N^{1-\alpha}}$ and α satisfying $\lceil N^\alpha \rceil = \lfloor \frac{1}{2\rho'\sqrt{m}} \rfloor \le \frac{1}{2\rho'\sqrt{m}}$ where $\frac{1}{2\rho'\sqrt{m}} > 1$. Then, ρ' satisfies $\rho' \le \frac{1}{2\lceil N^\alpha \rceil \sqrt{m}} < \frac{1}{2N^\alpha\sqrt{m}}$. We obtain a neural network $\tilde{f}_j$ with $\tilde{O}\left(N^{\frac{\alpha}{2}}\right)$ parameters such that:

$$\begin{aligned} \tilde{f}_j(\boldsymbol{x}) &= y_i \quad \forall \boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}), i \in I_j, \\ \mathbb{P}_{\boldsymbol{x} \in \text{Unif}(\mathcal{B}(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))}\left[\tilde{f}_j(\boldsymbol{x}) \in \{0, y_i\}\right] &\ge 1 - \frac{\eta}{N^{1-\alpha}} \quad \forall i \in [N] \backslash I_j. \end{aligned}$$

Thus, we have

$$\mathbb{P}_{\boldsymbol{x} \in \text{Unif}(\mathcal{B}(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))}\left[\tilde{f}_j(\boldsymbol{x}) \in \{0, y_i\}\right] \ge 1 - \frac{\eta}{N^{1-\alpha}} \quad \forall i \in [N], j \in [N^{1-\alpha}]. \tag{13}$$

We define for each j:

$$f_j\left(\begin{pmatrix}\boldsymbol{x} \\ y\end{pmatrix}\right) = \begin{pmatrix}\boldsymbol{x} \\ y + \sigma\left(\tilde{f}_j(\boldsymbol{x}) - y\right)\end{pmatrix},$$

so that the last coordinate $y + \sigma\left(\tilde{f}_j(\boldsymbol{x}) - y\right) = \max\{\tilde{f}_j(\boldsymbol{x}), y\}$. Finally, we define the network

$$f(\boldsymbol{x}) := \begin{pmatrix}\mathbf{0} \\ 1\end{pmatrix}^\top f_{N^{1-\alpha}} \circ \cdots \circ f_2 \circ f_1 \begin{pmatrix} f_{\text{proj}}(\boldsymbol{x}) \\ 0 \end{pmatrix}.$$

We now verify the correctness of the construction. For any $x \in \mathcal{B}(x_i, \rho\epsilon_{\mathcal{D},p})$, since we partition $[N]$ into disjoint groups $\{I_j\}_{j \in [N^{1-\alpha}]}$, there exists a unique index j such that $i \in I_j$ and thus $\tilde{f}_j(x) = y_i$ holds. For all $j' \ne j$, the networks satisfy $\tilde{f}_{j'}(x) \in \{0, y_i\}$ with high probability, so none of them can exceed y_i. Since the final network outputs the maximum among y and all $\tilde{f}_j(x)$, we have $f(x) = y_i$ as long as each $\tilde{f}_j(x) \in \{0, y_i\}$. Therefore,

$$\left[\tilde{f}_j(\boldsymbol{x}) \in \{0, y_i\} \quad \forall j \in [N^{1-\alpha}]\right] \Rightarrow f(\boldsymbol{x}) = y_i.$$

Since each $\tilde{f}_j$ satisfies $\mathbb{P}_{x \sim \text{Unif}(\mathcal{B}(x_i, \rho\epsilon_{\mathcal{D},p}))}[\tilde{f}_j(x) \in \{0, y_i\}] \ge 1 - \frac{\eta}{N^{1-\alpha}}$ for all $j \in [N^{1-\alpha}]$, we lower bound the success probability using the product:

Thus, we have:

$$\begin{aligned}\mathbb{P}_{\boldsymbol{x}\in\mathrm{Unif}(\mathcal{B}(\boldsymbol{x}_i,\rho\epsilon_{\mathcal{D},p}))}\left[f(\boldsymbol{x})=y_i\right] \geq & \mathbb{P}_{\boldsymbol{x}\in\mathrm{Unif}(\mathcal{B}(\boldsymbol{x}_i,\rho\epsilon_{\mathcal{D},p}))}\left[\tilde{f}_j(\boldsymbol{x})\in\{0,y_i\} \quad \forall j\in[N^{1-\alpha}]\right] \\ \geq & \prod_{j\in[N^{1-\alpha}]} \mathbb{P}_{\boldsymbol{x}\in\mathrm{Unif}(\mathcal{B}(\boldsymbol{x}_i,\rho\epsilon_{\mathcal{D},p}))}\left[\tilde{f}_j(\boldsymbol{x})\in\{0,y_i\}\right] \\ \overset{(a)}{\geq} & \left(1-\frac{\eta}{N^{1-\alpha}}\right)^{N^{1-\alpha}} \\ \geq & 1-\eta,\end{aligned}$$

where (a) holds by Equation (13).

We verify the number of parameters of f. The network f_{proj} needs $dm = O(d\log N)$ parameters. When we obtain $\tilde{f}_j$ by Theorem 21, data points need to be translated. It needs $m^2 = O((\log N)^2)$ parameters. Each $\tilde{f}_j$ has $\tilde{O}\left(N^{\frac{\alpha}{2}}\right)$ parameters so the number of parameters is

$$\tilde{O}\left(N^{1-\alpha}\times N^{\frac{\alpha}{2}}\right)=\tilde{O}\left(N^{1-\frac{\alpha}{2}}\right)\overset{(a)}{=}\tilde{O}(N\rho'^{\frac{1}{2}}m^{\frac{1}{4}})\overset{(b)}{=}\tilde{O}\left(Nd^{\frac{1}{4}}\rho^{\frac{1}{2}}\right),$$

where (a) holds by $\lceil N^\alpha\rceil = \lfloor\frac{1}{2\rho'\sqrt{m}}\rfloor$ and (b) holds by the definition of $\rho' = \frac{5}{2}\sqrt{\frac{d}{m}}\rho$. ■

This construction is motivated by the need to handle overlapped robustness balls with same label. We transform the construction of classical memorization in Vardi et al. [16] in two key directions: first, from memorizing isolated data points $\boldsymbol{x}_i$ to memorizing entire robustness neighborhoods $\mathcal{B}_p(\boldsymbol{x}_i,\mu)$; and second, to ensuring correct classification even within regions where multiple robustness balls with same label overlap. To accomplish this, we introduce disjoint, integer-aligned interval encodings and carefully control the error propagation caused by dimension reduction, as addressed in Theorem 20.

E.2.1. MEMORIZATION OF INTEGERS WITH SUBLINEAR PARAMETERS IN N

Lemmas in this section are slight extension of those in Vardi et al. [16], adapted to our integer-based encoding scheme.

From here, $\mathrm{BIN}_{i:j}(n)$ denotes the bit string from position i to j (inclusive) in the binary representation of n. For example, $\mathrm{BIN}_{1:3}(37) = 4$, since $(37)_{10} = (100101)_2$ so that $\mathrm{BIN}_{1:3}(37) = (100)_2 = (4)_{10}$.

Lemma 15 *Let $\eta > 0$ and $m, n \in \mathbb{N}$ with $m < n$. Then, there exists a neural network $F : \mathbb{R} \to \mathbb{R}$ with width 2 and depth 2 such that $F(x) = 1$ for $x \in [m, n-\eta]$ and $F(x) = 0$ for $x \leq m-\eta$ or $x \geq n$.*

Proof We construct a network F:

$$F(x)=\sigma\left(1-\sigma\left(-\frac{1}{\eta}(x-m)\right)\right)+\sigma\left(1-\sigma\left(\frac{1}{\eta}(x-(n-\eta))\right)\right)+1.$$

It satisfies the requirements with depth 2 and width 2. ■

Lemma 16 *Let $\eta \in (0,1)$, and let $m_1 < \cdots < m_N$ be natural numbers. Let $N_1, N_2 \in \mathbb{N}$ satisfy $N_1 \cdot N_2 \geq N$, and let $w_1, \ldots, w_{N_1} \in \mathbb{N}$. Then, there exists a neural network $F : \mathbb{R} \to \mathbb{R}$ with width 4 and depth $3N_1 + 2$ such that, for all $i \in [N]$ and all $x \in [m_i, m_i + 1 - \eta]$,*

$$F(x) = w_{\lceil \frac{i}{N_2} \rceil},$$

and $F(x) = 0$ for $x \in \mathbb{R} \setminus \bigcup_{j \in [N_1]} (m_{(j-1)N_2+1} - \eta,\ m_{jN_2} + 1)$.

Proof Let $j \in [N_1]$. We define network blocks $\tilde{F}_j : \mathbb{R} \to \mathbb{R}$ and $F_j : \mathbb{R}^2 \to \mathbb{R}^2$ as follows. By applying Theorem 15, we construct $\tilde{F}_j$ such that:

$$\tilde{F}_j(x) = \begin{cases} 1 & \text{if } x \in \left[m_{(j-1)N_2+1},\ m_{jN_2} + 1 - \eta\right], \\ 0 & \text{if } x \leq m_{(j-1)N_2+1} - \eta \text{ or } x \geq m_{jN_2} + 1. \end{cases}$$

In other words, for any $x \in [m_i, m_i + 1 - \eta]$, $\tilde{F}_i(x) = 1$ if $i \in [(j-1) \cdot N_2 + 1, j \cdot N_2]$, and $\tilde{F}_j(x) = 0$ otherwise.

Next, we define:

$$F_j\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = \begin{pmatrix} x \\ y + w_j \cdot \tilde{F}_j(x) \end{pmatrix}.$$

Finally, we define the network $F(x) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}^\top F_{N_1} \circ \cdots \circ F_1 \left(\begin{pmatrix} x \\ 0 \end{pmatrix}\right)$.

We now verify the correctness of the construction. For $i \in [N]$, and let $x \in [m_i, m_i + 1 - \eta]$. For $j = \lceil \frac{i}{N_2} \rceil$, we have $\tilde{F}_j(\tilde{x}_i) = 1$, and for all $j' \neq j$, $\tilde{F}_{j'}(\tilde{x}_i) = 0$. Therefore, the output of F satisfies $F(\tilde{x}_i) = w_j = w_{\lceil \frac{i}{N_2} \rceil}$.

The width of each F_j is at most the width required to implement $\tilde{F}_j$, plus two additional units to carry the values of x and y. Since the width of $\tilde{F}_j$ is 2, the width of F is at most 4. Each block F_j has depth 3, and F is a composition of N_1 blocks. Additionally, one layer is used for the input to get $x \mapsto \begin{pmatrix} x \\ 0 \end{pmatrix}$, and another to extract the last coordinate of the final input. Thus, the total depth of F is $3N_1 + 2$. ■

Lemma 17 (Lemma A.7, Vardi et al. [16]) *Let $n \in \mathbb{N}$ and let $i, j \in \mathbb{N}$ with $i < j \leq n$. Denote Telgarsky's triangle function by $\varphi(z) := \sigma(\sigma(2z) - \sigma(4z - 2))$. Then, there exists a neural network $F : \mathbb{R}^2 \to \mathbb{R}^3$ with width 5 and depth $3(j - i + 1)$, such that for any $x \in \mathbb{N}$ with $len(x) \leq n$, if the input of F is $\begin{pmatrix} \varphi^{(i-1)}\left(\frac{x}{2^n} + \frac{1}{2^{n+1}}\right) \\ \varphi^{(i-1)}\left(\frac{x}{2^n} + \frac{1}{2^{n+2}}\right) \end{pmatrix}$, then it outputs: $\begin{pmatrix} \varphi^{(j)}\left(\frac{x}{2^n} + \frac{1}{2^{n+1}}\right) \\ \varphi^{(j)}\left(\frac{x}{2^n} + \frac{1}{2^{n+2}}\right) \\ \mathrm{BIN}_{i:j}(x) \end{pmatrix}$.*

Lemma 18 (Extension of Lemma A.5, Vardi et al. [16]) *Let $\eta > 0$, and let $n, \rho, c \in \mathbb{N}$ and $u, w \in \mathbb{N}$. Assume that for all $\ell, k \in \{0, 1, \ldots, n-1\}$ with $\ell \neq k$, the bit segments of u satisfy*

$$\mathrm{BIN}_{\rho \cdot \ell + 1 : \rho \cdot (\ell+1)}(u) \neq \mathrm{BIN}_{\rho \cdot k + 1 : \rho \cdot (k+1)}(u).$$

Then, there exists a neural network $F : \mathbb{R}^3 \to \mathbb{R}$ with width 12 and depth $3n \cdot \max\{\rho, c\} + 2n + 2$, such that the following holds:

For every $x > 0$, if there exist $j \in \{0, 1, \ldots, n-1\}$ such that

$$x \in [\mathrm{BIN}_{\rho\cdot j+1:\rho\cdot(j+1)}(u), \mathrm{BIN}_{\rho\cdot j+1:\rho\cdot(j+1)}(u) + 1 - \eta],$$

then the network satisfies

$$F\left(\begin{pmatrix} x \\ w \\ u \end{pmatrix}\right) = \mathrm{BIN}_{c\cdot j+1:c\cdot(j+1)}(w)\,.$$

Moreover, $F\left(\begin{pmatrix} x \\ w \\ u \end{pmatrix}\right) = 0$ *for*

$$x \in \mathbb{R} \setminus \bigcup_{j\in\{0,\cdots,n-1\}} (\mathrm{BIN}_{\rho\cdot j+1:\rho\cdot(j+1)}(u) - \eta, \mathrm{BIN}_{\rho\cdot j+1:\rho\cdot(j+1)}(u) + 1).$$

Proof We define the triangle function $\varphi(z) := \sigma(\sigma(2z) - \sigma(4z-2))$ as introduced by Telgarsky [15]. For $i \in \{0, 1, \ldots, n-1\}$, we construct a network block F_i:

$$F_i : \begin{pmatrix} x \\ \varphi^{(i\cdot\rho)}\left(\frac{u}{2^{n\cdot\rho}} + \frac{1}{2^{n\cdot\rho+1}}\right) \\ \varphi^{(i\cdot\rho)}\left(\frac{u}{2^{n\cdot\rho}} + \frac{1}{2^{n\cdot\rho+2}}\right) \\ \varphi^{(i\cdot c)}\left(\frac{w}{2^{n\cdot c}} + \frac{1}{2^{n\cdot c+1}}\right) \\ \varphi^{(i\cdot c)}\left(\frac{w}{2^{n\cdot c}} + \frac{1}{2^{n\cdot c+2}}\right) \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ \varphi^{((i+1)\cdot\rho)}\left(\frac{u}{2^{n\cdot\rho}} + \frac{1}{2^{n\cdot\rho+1}}\right) \\ \varphi^{((i+1)\cdot\rho)}\left(\frac{u}{2^{n\cdot\rho}} + \frac{1}{2^{n\cdot\rho+2}}\right) \\ \varphi^{((i+1)\cdot c)}\left(\frac{w}{2^{n\cdot c}} + \frac{1}{2^{n\cdot c+1}}\right) \\ \varphi^{((i+1)\cdot c)}\left(\frac{w}{2^{n\cdot c}} + \frac{1}{2^{n\cdot c+2}}\right) \\ y + y_i \end{pmatrix}$$

where $y_i = \mathrm{BIN}_{i\cdot c+1:(i+1)\cdot c}(w)$ if $x \in [\mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u),\ \mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u)+1-\eta]$, and $y_i = 0$ if $x \le \mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u) - \eta$ or $x \ge \mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u) + 1$.

To compute y_i, we first extract the relevant bit segments from u and w using Theorem 17. We define two subnetworks F_i^w, F_i^u:

$$F_i^u : \begin{pmatrix} \varphi^{(i\cdot\rho)}\left(\frac{u}{2^{n\cdot\rho}} + \frac{1}{2^{n\cdot\rho+1}}\right) \\ \varphi^{(i\cdot\rho)}\left(\frac{u}{2^{n\cdot\rho}} + \frac{1}{2^{n\cdot\rho+2}}\right) \end{pmatrix} \mapsto \begin{pmatrix} \varphi^{((i+1)\cdot\rho)}\left(\frac{u}{2^{n\cdot\rho}} + \frac{1}{2^{n\cdot\rho+1}}\right) \\ \varphi^{((i+1)\cdot\rho)}\left(\frac{u}{2^{n\cdot\rho}} + \frac{1}{2^{n\cdot\rho+2}}\right) \\ \mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u) \end{pmatrix}$$

$$F_i^w : \begin{pmatrix} \varphi^{(i\cdot c)}\left(\frac{w}{2^{n\cdot c}} + \frac{1}{2^{n\cdot c+1}}\right) \\ \varphi^{(i\cdot c)}\left(\frac{w}{2^{n\cdot c}} + \frac{1}{2^{n\cdot c+2}}\right) \end{pmatrix} \mapsto \begin{pmatrix} \varphi^{((i+1)\cdot c)}\left(\frac{w}{2^{n\cdot c}} + \frac{1}{2^{n\cdot c+1}}\right) \\ \varphi^{((i+1)\cdot c)}\left(\frac{w}{2^{n\cdot c}} + \frac{1}{2^{n\cdot c+2}}\right) \\ \mathrm{BIN}_{i\cdot c+1:(i+1)\cdot c}(w) \end{pmatrix}.$$

A subnetwork F_i^u maps the pair of triangle encodings of u to the updated encodings for $i+1$, along with the extracted bits $\mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u)$. A subnetwork F_i^w does the same for w, yielding $\mathrm{BIN}_{i\cdot c+1:(i+1)\cdot c}(w)$.

We then construct a network with width 2 and depth 2 to obtain y_i from inputs $\mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u)$ and x. Firstly, we use Theorem 15 to construct a network that output $\tilde{y}_i$:

$$\tilde{y}_i = \begin{cases} 1 & \text{if } x \in [\mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u),\ \mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u) + 1 - \eta], \\ 0 & \text{if } x \le \mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u) - \eta \text{ or } x \ge \mathrm{BIN}_{i\cdot\rho+1:(i+1)\cdot\rho}(u) + 1. \end{cases}$$

Secondly, we construct the following 1-layer network that use $\tilde{y}_i$ as input:

$$\begin{pmatrix} \tilde{y}_i \\ \mathrm{BIN}_{i\cdot c+1:(i+1)\cdot c}(w) \end{pmatrix} \mapsto \sigma\left(\tilde{y}_i \cdot 2^{c+1} - 2^{c+1} + \mathrm{BIN}_{i\cdot c+1:(i+1)\cdot c}(w)\right)$$

This ensures that the output is $\mathrm{BIN}_{i\cdot c+1:(i+1)\cdot c}(w)$ if $\tilde{y}_i = 1$, and the output is 0 if $\tilde{y}_i = 0$ since $\mathrm{BIN}_{i\cdot c+1:(i+1)\cdot c}(w) \leq 2^c$.

Finally, the full network F is constructed as a composition:

$$F := G \circ F_{n-1} \circ \cdots \circ F_0 \circ H \ ,$$

where for $x, w, u > 0$: (1) $H : \mathbb{R}^3 \to \mathbb{R}^6$ is a 1-layer network that maps (x, w, u) to the required initial encoding inputs, namely:

$$H : \begin{pmatrix} x \\ w \\ u \end{pmatrix} \mapsto \begin{pmatrix} x \\ \frac{u}{2^{n\cdot\rho}} + \frac{1}{2^{n\cdot\rho+1}} \\ \frac{u}{2^{n\cdot\rho}} + \frac{1}{2^{n\cdot\rho+2}} \\ \frac{w}{2^{n\cdot c}} + \frac{1}{2^{n\cdot c+1}} \\ \frac{w}{2^{n\cdot c}} + \frac{1}{2^{n\cdot c+2}} \\ 0 \end{pmatrix},$$

(2) $G : \mathbb{R}^5 \to \mathbb{R}$ is a 1-layer network that outputs the last coordinate.

We verify the correctness of the construction. The output of the full network is given by:

$$F\left(\begin{pmatrix} x \\ w \\ u \end{pmatrix}\right) = \sum_{i=0}^{n-1} y_i.$$

If there exists $j \in \{0, 1 \ldots, n-1\}$ such that $x \in [\mathrm{BIN}_{\rho\cdot j+1:\rho\cdot(j+1)}(u), \mathrm{BIN}_{\rho\cdot j+1:\rho\cdot(j+1)}(u) + 1 - \eta]$, then by the construction we obtain $y_j = \mathrm{BIN}_{c\cdot j+1:c\cdot(j+1)}(w)$, while $y_\ell = 0$ for all $\ell \neq j$. This is because the bit-encoded intervals are disjoint as $\mathrm{BIN}_{\rho\cdot\ell+1:\rho\cdot(\ell+1)}(u) \neq \mathrm{BIN}_{\rho\cdot k+1:\rho\cdot(k+1)}(u)$. Hence, the final output of F is:

$$\sum_{i=0}^{n-1} y_i = y_j = \mathrm{BIN}_{c\cdot j+1:c\cdot(j+1)}(w).$$

We now analyze the width and depth of the constructed network F. Each block F_i comprises F_i^w and F_i^u, each of width 5. In addition, two neurons are used to process x and y, resulting in a total width of 12. The outputs $\tilde{y}_i$ and y_i are produced by additional layers with width 2 and 1, respectively, both of which are smaller than 12. We also compose the networks H and G, with width 6 and 1, respectively, again remaining within 12.

Each of the networks F_i^u and F_i^w has depth at most $3\max\{\rho, c\}$. The layers obtaining $\tilde{y}_i$ and y_i contribute an additional 2 layers, resulting in a total depth of $3\max\{\rho, c\} + 2$ for each block F_i. Composing all n such blocks, and including one additional layer each for H and G, the total depth of network F is $3n \cdot \max\{\rho, c\} + 2n + 2$.

■

E.2.2. PRECISE CONTROL OF ROBUST MEMORIZATION ERROR

Theorem 21 constructs the network for Stage II in Theorem 14, while the robust memorization error is controlled in Theorem 20.

Lemma 19 *Let $N, d, C \in \mathbb{N}$, and let $(m_1, y_1), \ldots, (m_N, y_N) \in \mathcal{D}_{1,N,C} \subset \mathbb{N} \times [C]$ be a set of N labeled samples with $m_i \neq m_j$ for every $i \neq j$. Then, there exists a neural network $F : \mathbb{R}^d \to \mathbb{R}$ with $\tilde{O}(\sqrt{N})$ parameters such that*

$$F(m) = \begin{cases} y_i & \text{for every } m \in \{m_i\}_{i \in N}, \\ 0 & \text{for every } m \in \mathbb{N} \backslash \{m_i\}_{i \in N}. \end{cases}$$

Proof Let $\mathcal{M} = \{m_i\}_{i \in N}$. We group the elements in $\mathcal{M}$ to $\lceil \sqrt{N} \rceil$ groups, each containing at most $\lfloor \sqrt{N} \rfloor + 1$ natural numbers inside. For each interval indexed by $j \in \{1, \ldots, \lceil \sqrt{N} \rceil\}$, we define two integers $w_j, u_j \in \mathbb{N}$ to encode the integer $m_i \in \mathcal{M}$ and the corresponding labels y_i as follows.

For each $i \in [N]$, letting $j := \left\lceil \frac{i}{\lfloor \sqrt{N} \rfloor + 1} \right\rceil$, $k := i \mod (\lfloor \sqrt{N} \rfloor + 1)$ and $R := max_{i \in [N]}\, m_i$, we define:

$$\begin{aligned} \mathrm{BIN}_{k \cdot \log_2 R + 1 : (k+1) \cdot \log_2 R}(u_j) &= m_i \\ \mathrm{BIN}_{k \cdot \log_2 C + 1 : (k+1) \cdot \log_2 C}(w_j) &= y_i \, . \end{aligned}$$

Thus, in each group j, the integer u_j contains $\log_2 R$ bits per integer, which represent the k-th integer in this group. In the same manner, w_j contains $\log_2 C$ bits per integer, which represent the label of the k-th integer in this group.

By applying Theorem 16 to $\eta = \frac{1}{2}$, we construct a neural network F_1 that maps $m \in \mathcal{M}$ to their corresponding groups, and maps $m \in \mathbb{N} \backslash \bigcup_{j \in [\lceil \sqrt{N} \rceil]} [m_{(j-1)(\lfloor \sqrt{N} \rfloor + 1) + 1},\ m_{j(\lfloor \sqrt{N} \rfloor + 1)} + 1)$ to 0. Thus, all natural numbers are assigned to their corresponding group or 0.

For each $i \in [N]$, we define the group index

$$j_i := \left\lceil \frac{i}{\lfloor \sqrt{N} \rfloor + 1} \right\rceil .$$

Then, the network F_1 maps any input $m \in \mathcal{M}$ to the representation

$$F_1(m) = \begin{pmatrix} m \\ w_{j_i} \\ u_{j_i} \end{pmatrix},$$

and $F_1(m) = \begin{pmatrix} m \\ 0 \\ 0 \end{pmatrix}$ for $m \in \mathbb{N} \backslash \bigcup_{j \in [\lceil \sqrt{N} \rceil]} [m_{(j-1)(\lfloor \sqrt{N} \rfloor + 1) + 1},\ m_{j(\lfloor \sqrt{N} \rfloor + 1)} + 1)$. The network F_2 has width 9 and depth $O(\sqrt{N})$.

Now, we apply Theorem 18 to construct a network $F_2 : \mathbb{R}^3 \to \mathbb{R}$ with the following property. For each $i \in [N], j \in \left[\lceil \sqrt{N} \rceil\right]$, and $k \in \left\{0, \ldots, \lfloor \sqrt{N} \rfloor\right\}$, suppose that m_i is the k-th integer in the j-th group. Then, the network satisfies :

$$F_2 \left(\begin{pmatrix} m_i \\ w_j \\ u_j \end{pmatrix} \right) = \mathrm{BIN}_{k \cdot \log_2 C + 1 : (k+1) \cdot \log_2 C}(w_j) = y_i.$$

Moreover, for $m \in \mathbb{N} \setminus \mathcal{M}$, $F_2\left(\begin{pmatrix} m \\ w_j \\ u_j \end{pmatrix}\right) = 0$ or $F_2\left(\begin{pmatrix} m \\ 0 \\ 0 \end{pmatrix}\right) = 0$. Thus, the network F_3 extracts the label corresponding to each data point from the encoded label set of the group to which the interval belongs or outputs 0. The network F_3 has width 12, depth $O(\sqrt{N})$.

Finally, we define the classifier network $F : \mathbb{R}^d \to \mathbb{R}$ as

$$F(\boldsymbol{x}) = F_2 \circ F_1(\boldsymbol{x}).$$

The overall network F has width 12 and depth $\tilde{O}(\sqrt{N})$, which correspond to the maximum width and total depth of its component networks. ■

Lemma 20 *Let $\mathcal{B}_2(x_0, \mu)$ be a Euclidean ball with center $x_0 \in \mathbb{R}^d$ and radius $\mu > 0$. Let $u \in \mathbb{R}^d$ be a unit vector, and define the affine function $f(x) := \frac{1}{2\mu}(u^\top x + b)$ for some $b \in \mathbb{R}$. Then for any interval $I \subset \mathbb{R}$ of length η, the volume fraction of the ball mapped into I satisfies:*

$$\frac{\mathrm{Vol}\left(\{x \in \mathcal{B}_2(x_0, \mu) \mid f(x) \in I\}\right)}{\mathrm{Vol}\left(\mathcal{B}_2(x_0, \mu)\right)} \leq \frac{2\eta}{B\left(\frac{1}{2}, \frac{d+1}{2}\right)},$$

where $B(\cdot, \cdot)$ denotes the Beta function.

Proof Let $x = x_0 + \mu y$, so that $y \in \mathcal{B}_d(0, 1)$. Under this change of variables,

$$f(x) = \frac{1}{2\mu}(u^\top(x_0 + \mu y) + b) = \frac{1}{2}(u^\top y) + \frac{1}{2\mu}(u^\top x_0 + b).$$

Thus, $f(x) \in I$ if and only if $u^\top y \in J$, where

$$J := 2I - \frac{1}{\mu}(u^\top x_0 + b) \subset \mathbb{R}$$

is an interval of length 2η. We define the preimage set

$$A := \{x \in \mathcal{B}_2(x_0, \mu) \mid f(x) \in I\}.$$

Then,

$$\mathrm{Vol}(A) = \mu^d \cdot \mathrm{Vol}\left(\left\{y \in \mathcal{B}_d(0, 1) \middle| u^\top y \in J\right\}\right).$$

The distribution of $u^\top y$, where $y \sim \mathrm{Unif}(\mathcal{B}_2(0, 1))$, has density

$$p(t) = \frac{1}{Z_d}(1 - t^2)^{\frac{d-1}{2}} \text{ for } t \in [-1, 1], \quad Z_d = B\left(\frac{1}{2}, \frac{d+1}{2}\right).$$

Thus,

$$\mathrm{Vol}(A) = \mu^d \int_J p(t)dt \leq \mu^d \cdot \int_J 1\, dt = \mu^d \cdot 2\eta,$$
$$\mathrm{Vol}(B_2(x_0, \mu)) = \mu^d \cdot Z_d.$$

Hence,

$$\frac{\mathrm{Vol}\left(x \in \mathcal{B}_2(x_0, \mu) : f(x) \in I\right)}{\mathrm{Vol}\left(\mathcal{B}_2(x_0, \mu)\right)} = \frac{\mathrm{Vol}(A)}{\mathrm{Vol}(B_2(x_0, \mu))} \leq \frac{2\eta}{Z_d} = \frac{2\eta}{B\left(\frac{1}{2}, \frac{d+1}{2}\right)}.$$

■

Lemma 21 *Let $\eta \in (0,1)$, $\alpha \in [0,1]$ and let $\{(\boldsymbol{x}_i, y_i)\}_{i\in[N]} = \mathcal{D} \in \mathcal{D}_{d,N,C}$ be a class-separated dataset. Suppose the robustness ratio ρ satisfies $\rho < \frac{1}{2N^\alpha\sqrt{d}}$. Then, for any $I \subset [N]$ with $|I| = \lfloor N^\alpha \rfloor$, there exist a neural network f with $\tilde{O}\left(N^{\frac{\alpha}{2}}\right)$ parameters such that:*

$$f(\boldsymbol{x}) = y_i \quad \forall \boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}), i \in I,$$
$$\mathbb{P}_{\boldsymbol{x}\in\mathrm{Unif}(\mathcal{B}(\boldsymbol{x}_i,\rho\epsilon_{\mathcal{D},p}))}\left[f(\boldsymbol{x}) \in \{0, y_i\}\right] \geq 1-\eta \quad \forall i \in [N]\backslash I.$$

Proof We begin with the same assumptions on the dataset $\{(\boldsymbol{x}_i, y_i)\}_{i\in I}$ as used in the proof of Theorem 12 without loss of generability. Given the robustness ratio ρ, we scale the dataset $\mathcal{D}$ by a factor of $\frac{1}{4\lfloor N^\alpha\rfloor\rho\epsilon_{\mathcal{D},p}}$. Then, applying Theorem 24, we translate the data points so that:

$$\mathrm{dist}(x_{i,j} - b_j, \mathbb{Z}) \geq \frac{1}{2\lfloor N^\alpha\rfloor}, \quad \forall i \in I, j \in [d]. \tag{14}$$

We now consider the scaled and translated dataset $\mathcal{D}'$ such that (i) is point-separated with $\epsilon^\dagger_{\mathcal{D}',p} = \frac{1}{4\lfloor N^\alpha\rfloor\rho}$ separation (ii) holds Equation (14), and (iii) all coordinates of data points are positive. Moreover, we consider $\mu' = \frac{1}{4\lfloor N^\alpha\rfloor}$-ball.

The construction closely follows that of Theorem 12. Define a neural network $\overline{\mathrm{Flatten}} : \mathbb{R}^d \to \mathbb{R}$ as

$$\overline{\mathrm{Flatten}}(\boldsymbol{x}) = R^{d-1}\overline{\mathrm{Floor}}(x_1) + \cdots + \overline{\mathrm{Floor}}(x_d),$$

where $\overline{\mathrm{Floor}} := \overline{\mathrm{Floor}_{\lceil\log_2 R\rceil}}$ is the approximate floor function with $O(\log_2 R)$ parameters obtained from Theorem 25 with $\gamma = \eta' \leq \frac{\mu' B\left(\frac{1}{2}, \frac{d+1}{2}\right)}{2d}\eta$.

We also define:

$$\mathrm{Flatten}(\boldsymbol{x}) := R^{d-1}\lfloor x_1\rfloor + R^{d-2}\lfloor x_2\rfloor + \cdots + \lfloor x_d\rfloor.$$

By construction of $\overline{\mathrm{Floor}}$, for any $\boldsymbol{x}$ satisfying:

$$x_j - \lfloor x_j\rfloor > \eta' \quad \forall j \in [d], \tag{15}$$

we have:

$$\overline{\mathrm{Flatten}}(\boldsymbol{x}) = \mathrm{Flatten}(\boldsymbol{x}).$$

From Equation (14), each coordinate of the data points $\boldsymbol{x}_i$ for $i \in I$ has a distance at least $2\mu'$ from the integer lattice. Thus, for $\boldsymbol{x}$ in μ'-ball centered with at $\boldsymbol{x}_i$, namely, $\boldsymbol{x} \in \mathcal{B}_2(\boldsymbol{x}_i, \mu')$, it has a distance at least μ' from the integer lattice. Thus, the ball lies within a grid, and we have:

$$x_j - \lfloor x_j\rfloor \geq \mu' \overset{(a)}{>} \frac{\mu' B\left(\frac{1}{2}, \frac{d+1}{2}\right)}{2d}\eta \overset{(b)}{\geq} \eta',$$

where (a) holds since $B\left(\frac{1}{2}, \frac{d+1}{2}\right) \leq \frac{\pi}{2}$ and $\eta < 1$, and (b) holds from the definition of η'. Thus, for any $i \in I$ and any $\boldsymbol{x} \in \mathcal{B}_2(\boldsymbol{x}_i, \mu')$, the point $\boldsymbol{x}$ is mapped to the same integer value $m_i := \overline{\mathrm{Flatten}}(\boldsymbol{x})$, namely,

$$\overline{\mathrm{Flatten}}(\boldsymbol{x}) = \mathrm{Flatten}(\boldsymbol{x}) = m_i \in \mathbb{N} \quad \forall \boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}_i, \mu'),\ i \in I.$$

Note that the lattice distance condition in Equation (14) applies only to the subset $\{(\boldsymbol{x}_i, y_i)\}_{i\in I}$, rather than the entire dataset. As a result, for indices $i \in [N] \setminus I$, the distance from the lattice is not

guaranteed. Thus, it can lie across the lattice. However, it guarantees that data points with different labels are assigned to different grid cells since:

$$\epsilon_{\mathcal{D}',p} = \frac{1}{4\lfloor N^\alpha \rfloor \rho} \overset{(a)}{>} \frac{N^\alpha \sqrt{d}}{2\lfloor N^\alpha \rfloor} \geq \frac{\sqrt{d}}{2},$$

where (a) holds form the ρ condition. So for any $i \neq j$ such that $y_i \neq y_j$, two balls $\mathcal{B}(\boldsymbol{x}_i, \mu')$ and $\mathcal{B}(\boldsymbol{x}_j, \mu')$ never intersect the same grid cells. For $c \in [C]$, let

$$G_c := \bigcup_{\substack{i \in [I] \\ \text{s.t. } y_i = c}} \{m_i\},$$

then, we define $G := \bigcup_{c \in [C]} G_c = \{m_i\}_{i \in [N]}$.

For $\boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}_i, \mu')$,

$$\mathrm{Flatten}(\boldsymbol{x}) \notin \bigcup_{\substack{c \in [C] \\ \text{s.t.} y_i \neq c}} G_c.$$

Since $\mathrm{Flatten}(\boldsymbol{x})$ is integer, it is equivalent with the following:

$$\mathrm{Flatten}(\boldsymbol{x}) \notin \bigcup_{\substack{c \in [C] \\ \text{s.t.} y_i \neq c}} G_c \iff [\mathrm{Flatten}(\boldsymbol{x}) \notin G \quad \text{or} \quad \mathrm{Flatten}(\boldsymbol{x}) \in G_{y_i}] \tag{16}$$

Applying Theorem 19 to the dataset $\{(m_i, y_i)\}_{i \in I}$, we obtain a neural network f_{mem} with $\tilde{O}(N^{\frac{\alpha}{2}})$ parameters satisfying:

$$f_{\text{mem}}(m) = \begin{cases} y_i & \text{for every } m \in \{m_i\}_{i \in N} = G, \\ 0 & \text{for every } m \in \mathbb{N} \backslash \{m_i\}_{i \in N} = \mathbb{N} \backslash G. \end{cases}$$

We define the final network as

$$f := f_{\text{mem}} \circ \overline{\mathrm{Flatten}}.$$

It has $\tilde{O}(N^{\frac{\alpha}{2}} + d \log_2 R)$ parameters. Neglecting the trivial linear term in d, the total parameter count is $\tilde{O}(N^{\frac{\alpha}{2}})$.

For $i \in I$ and any $\boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}_i, \mu')$, we have

$$f(\boldsymbol{x}) = f_{\text{mem}} \circ \overline{\mathrm{Flatten}}(\boldsymbol{x}) = f_{\text{mem}}(m_i) = y_i.$$

Next, consider $i \in [N] \setminus I$. For any $\boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}_i, \mu')$, it satisfies Equation (16). Let's consider each case. First, if $\mathrm{Flatten}(\boldsymbol{x}) \notin G$, it holds $f_{\text{mem}} \circ \mathrm{Flatten}(\boldsymbol{x}) = 0$ by the construction of f_{mem}. Second, if $\mathrm{Flatten}(\boldsymbol{x}) \in G_{y_i}$, it holds $f_{\text{mem}} \circ \mathrm{Flatten}(\boldsymbol{x}) = y_i$.

From the construction of $\overline{\mathrm{Flatten}}$, if $\boldsymbol{x}$ holds Equation (15), we have

$$f_{\text{mem}} \circ \overline{\mathrm{Flatten}}(\boldsymbol{x}) = f_{\text{mem}} \circ \mathrm{Flatten}(\boldsymbol{x}) \in \{0, y_i\}.$$

What is left here is to consider the probability of when Equation (15) holds.

To bound the probability of when Equation (15) does not hold, observe that

$$\begin{aligned}
&\mathbb{P}_{\boldsymbol{x}\in\mathrm{Unif}(\mathcal{B}(\boldsymbol{x}_i,\rho\epsilon_{\mathcal{D},p}))}\left[f(\boldsymbol{x})\notin\{0,y_i\}\right]\\
=&\mathbb{P}_{\boldsymbol{x}\in\mathrm{Unif}(\mathcal{B}(\boldsymbol{x}_i,\mu'))}\left[x_j-\lfloor x_j\rfloor\le\eta'\quad\exists j\in[d]\right]\\
\le&\sum_{j\in[d]}\mathbb{P}_{\boldsymbol{x}\in\mathrm{Unif}(\mathcal{B}(\boldsymbol{x}_i,\mu'))}\left[x_j-\lfloor x_j\rfloor\le\eta'\right]\\
\le&\sum_{j\in[d]}\max_{\substack{\mathcal{I}_j\\ \text{s.t. }\mathrm{Len}(\mathcal{I}_j)=\eta'}}\mathbb{P}_{\boldsymbol{x}\in\mathrm{Unif}(\mathcal{B}(\boldsymbol{x}_i,\mu'))}\left[x_j\in\mathcal{I}_j\right]\\
\overset{(a)}{\le}&\sum_{j\in[d]}\frac{2\eta'}{\mu' B\left(\frac{1}{2},\frac{d+1}{2}\right)}\\
=&\frac{2\eta' d}{\mu' B\left(\frac{1}{2},\frac{d+1}{2}\right)}\\
\le&\eta
\end{aligned}$$

The inequality (a) follows from Theorem 20 applied to a unit vector $u=\mathbf{e_j}$, $b=0$, and an interval $\frac{\mathcal{I}_j}{\mu'}$. This concludes the proof. ■

E.3. Sufficient Condition for Robust Memorization with Large Robustness Radius

Theorem 22 *Let $\rho\in\left(\frac{1}{5\sqrt{d}},1\right)$. For any dataset $\mathcal{D}\in\mathcal{D}_{d,N,C}$, there exists $f\in\mathcal{F}_{d,P}$ that ρ-robustly memorizes $\mathcal{D}$, where the number of parameters satisfies $P=O(Nd^2\rho^4+d^2\rho^2)$.*

Proof Let $\mathcal{D}=\{(\boldsymbol{x}_i,y_i)\}_{i\in[N]}\in\mathcal{D}_{d,N,C}$ be given. We divide the proof into three cases, the first case under $\rho\in[1/3,1)$, the second case under $\rho\in(1/5\sqrt{d},1/3)$ and $d<600\log N$, and finally the third case under $\rho\in(1/5\sqrt{d},1/3)$ and $d\ge 600\log N$. The first two cases follow easily from the prior works, while the third case requires a careful analysis using the dimension reduction technique that follows from the Johnson-Lindenstrauss lemma. Let us deal with each case one by one.

Case I: $\rho\in[1/3,1)$. In the first case, where $\rho\in[1/3,1)$, the result directly follows from the prior result by Yu et al. [17]. In particular, we apply Theorem 43. Let us denote $R:=\max_{i\in[N]}\|\boldsymbol{x}_i\|_2$ and $\gamma:=(1-\rho)\epsilon_{\mathcal{D},p}$. Note that $R\ge\|\boldsymbol{x}_i\|_\infty$ for all $i\in[N]$ as $\|\boldsymbol{x}\|_2\ge\|\boldsymbol{x}\|_\infty$ for all $\boldsymbol{x}\in\mathbb{R}^d$. By applying Theorem 43, there exists $f\in\mathcal{F}_{d,P}$ with $P=O(Nd^2(\log(\frac{d}{\gamma^2}+\log R))$ parameters that ρ-robustly memorize $\mathcal{D}$. The number of parameters can be further bounded as follows:

$$O(Nd^2(\log(\frac{d}{\gamma^2}+\log R))\overset{(a)}{=}O(Nd^2\rho^4\cdot(\log(\frac{d}{\gamma^2}+\log R))\overset{(b)}{=}\tilde{O}(Nd^2\rho^4),$$

where (a) is due to $\rho=\Omega(1)$, (b) hides the logarithmic factors.

Case II: $\rho\in(1/5\sqrt{d},1/3)$ and $d<600\log N$. In the second case, where $d<600\log N$ and $(1/5\sqrt{d},1/3)$, the result also directly follows from the prior result by Yu et al. [17]. In particular, we apply Theorem 43. Let us denote $R:=\max_{i\in[N]}\|\boldsymbol{x}_i\|_2$ and $\gamma:=(1-\rho)\epsilon_{\mathcal{D},p}$. Note that $R\ge\|\boldsymbol{x}_i\|_\infty$ for all $i\in[N]$ as $\|\boldsymbol{x}\|_2\ge\|\boldsymbol{x}\|_\infty$ for all $\boldsymbol{x}\in\mathbb{R}^d$. By Theorem 43, there exists

$f \in \mathcal{F}_{d,P}$ with $P = O(Nd^2(\log(\frac{d}{\gamma^2} + \log R))$ parameters that ρ-robustly memorize $\mathcal{D}$. The number of parameters can be further bounded as follows:

$$O(Nd^2(\log(\frac{d}{\gamma^2} + \log R)) \overset{(a)}{=} O(N(\log N)^2 \cdot (\log(\frac{d}{\gamma^2} + \log R)) \overset{(b)}{=} \tilde{O}(N) \overset{(c)}{=} \tilde{O}(Nd^2\rho^4),$$

where (a) is due to $d \leq 600 \log N$, (b) hides the logarithmic factors, and (c) is because $N \leq 625Nd^2\rho^4$ for all $\rho \in \left(\frac{1}{5\sqrt{d}}, \frac{1}{3}\right)$.

Case III: $\rho \in (1/5\sqrt{d}, 1/3)$ and $d \geq 600 \log N$. In the third case, where $d \geq 600 \log N$, we utilize the dimension reduction technique by Theorem 28. We apply Theorem 28 with $m = \max\{\lceil 9d\rho^2 \rceil, \lceil 600 \log N \rceil, \lceil 10 \log d \rceil\}$ and $\alpha = 1/5$. Let us first check that the specified m satisfies the condition $24\alpha^{-2} \log N \leq m \leq d$ for the proposition to be applied. $\alpha = 1/5$ and $m \geq 600 \log N$ ensure the first inequality $24\alpha^{-2} \log N \leq m$. The second inequality $m \leq d$ is decomposed into three parts. Since $\rho \leq \frac{1}{3}$, we have $9d\rho^2 \leq d$ so that

$$\lceil 9d\rho^2 \rceil \leq d. \tag{17}$$

Moreover, $600 \log N \leq d$ implies

$$\lceil 600 \log N \rceil \leq d. \tag{18}$$

Additionally, as $N \geq 2$, we have $d \geq 600 \log N \geq 600 \log 2 \geq 400$. By Theorem 23, this implies $10 \log d \leq d$ and therefore

$$\lceil 10 \log d \rceil \leq d. \tag{19}$$

Gathering Equations (17) to (19) proves $m \leq d$.

By the Theorem 28, there exists 1-Lipchitz linear mapping $\phi : \mathbb{R}^d \to \mathbb{R}^m$ and $\beta > 0$ such that $\mathcal{D}' := \{(\phi(\boldsymbol{x}_i), y_i)\}_{i \in [N]} \in \mathcal{D}_{m,N,C}$ satisfies

$$\epsilon_{\mathcal{D}'} \geq \frac{4}{5}\beta\epsilon_{\mathcal{D}}. \tag{20}$$

As $m \geq 10 \log d$, the inequality $\beta \geq \frac{1}{2}\sqrt{\frac{m}{d}}$ is satisfied by Theorem 28. Therefore, we have

$$\beta \geq \frac{1}{2}\sqrt{\frac{m}{d}} \overset{(a)}{\geq} \frac{1}{2}\sqrt{\frac{\lceil 9d\rho^2 \rceil}{d}} \geq \frac{1}{2}\sqrt{\frac{9d\rho^2}{d}} = \frac{3}{2}\rho, \tag{21}$$

where (a) is by the definition of m. Moreover, since ϕ is 1-Lipchitz,

$$\|\phi(\boldsymbol{x}_i)\|_2 = \|\phi(\boldsymbol{x}_i - \boldsymbol{0})\|_2 = \|\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{0})\|_2 \leq \|\boldsymbol{x}_i - \boldsymbol{0}\|_2 = \|\boldsymbol{x}_i\|_2, \tag{22}$$

for all $i \in [N]$. Hence, by letting $R := \max_{i \in [N]}\{\|\boldsymbol{x}_i\|_2\}$, we have $\|\phi(\boldsymbol{x}_i)\|_2 \leq R$ for all $i \in [N]$.

Now, we set the first layer hidden matrix as the matrix $\boldsymbol{W}$ corresponding to ϕ under the standard basis of $\mathbb{R}^d$ and $\mathbb{R}^m$. Moreover, set the first hidden layer bias as $\boldsymbol{b} := 2R\boldsymbol{1} = 2R(1, 1, \cdots, 1) \in \mathbb{R}^m$. Then, we have

$$\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \geq \boldsymbol{0}, \tag{23}$$

for all $\boldsymbol{x} \in \mathcal{B}_2(\boldsymbol{x}_i, \epsilon_{\mathcal{D},2})$ for all $i \in [N]$, where the comparison between two vectors are element-wise. This is because for all $i \in [N], j \in [m]$ and $\boldsymbol{x} \in \mathcal{B}_2(\boldsymbol{x}, \epsilon_{\mathcal{D},2})$, we have

$$(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})_j = (\boldsymbol{W}\boldsymbol{x})_j + 2R \geq 2R - \|\boldsymbol{W}\boldsymbol{x}\|_2 \overset{(a)}{\geq} 2R - \|\boldsymbol{x}\|_2 \overset{(b)}{\geq} 2R - (R + \epsilon_{\mathcal{D},2}) \overset{(c)}{\geq} 0,$$

where (a) is by Equation (22), (b) is by the triangle inequality, and (c) is due to $R > \epsilon_{\mathcal{D},2}$.

We construct the first layer of the neural network as $f_1(\boldsymbol{x}) := \sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$ which includes the activation σ. Then, by above properties, $\mathcal{D}'' := \{(f_1(\boldsymbol{x}_i), y_i)\}_{i \in [N]}$ satisfies

$$\epsilon_{\mathcal{D}''} \geq \frac{6}{5}\rho\epsilon_{\mathcal{D}}. \tag{24}$$

This is because for $i \neq j$ with $y_i \neq y_j$ we have

$$\begin{aligned}
\|f_1(\boldsymbol{x}_i) - f_1(\boldsymbol{x}_j)\|_2 &= \|\sigma(\boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{b}) - \sigma(\boldsymbol{W}\boldsymbol{x}_j + \boldsymbol{b})\|_2 \\
&\overset{(a)}{=} \|(\boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{b}) - (\boldsymbol{W}\boldsymbol{x}_j + \boldsymbol{b})\|_2 \\
&= \|\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_k)\|_2 \\
&\overset{(b)}{\geq} 2\epsilon_{\mathcal{D}'} \\
&\overset{(c)}{\geq} 2 \times \frac{4}{5}\beta\epsilon_{\mathcal{D}} \\
&\overset{(d)}{\geq} 2 \times \frac{4}{5} \times \frac{3}{2}\rho\epsilon_{\mathcal{D}} \\
&= \frac{12}{5}\rho\epsilon_{\mathcal{D}},
\end{aligned}$$

where (a) is by Equation (23), (b) is by the definition of the $\epsilon_{\mathcal{D}'}$, (c) is by Equation (20), and (d) is by Equation (21). By Theorem 43 applied to $\mathcal{D}'' \in \mathcal{D}_{m,N,C}$, there exists $f_2 \in \mathcal{F}_{m,P}$ with $P = O(Nm^2(\log(\frac{d}{(\gamma'')^2} + \log R''))$ number of parameters that $\frac{5}{6}$-robustly memorize $\mathcal{D}''$, where

$$\begin{aligned}
\gamma'' &:= (1 - \frac{5}{6})\epsilon_{\mathcal{D}''} \overset{(a)}{\geq} \frac{1}{6} \times \frac{12}{5}\rho\epsilon_{\mathcal{D}} = \frac{2}{5}\rho\epsilon_{\mathcal{D}}, \\
R'' &:= \max_{i \in [N]} \|f_1(\boldsymbol{x}_i)\|_2 = \max_{i \in [N]} \|\sigma(\boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{b})\|_2 = \max_{i \in [N]} \|\boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{b}\|_2 \\
&\leq \max_{i \in [N]} \|\boldsymbol{W}\boldsymbol{x}_i\|_2 + \|\boldsymbol{b}\|_2 \leq 3R,
\end{aligned}$$

where (a) is by Equation (24).

Now, we claim that $f := f_2 \circ f_1$ ρ-robustly memorize $\mathcal{D}$. For any $i \in [N]$, take $\boldsymbol{x} \in \mathcal{B}_2(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},2})$. Then, by Equation (23), we have $f_1(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$ and $f_1(\boldsymbol{x}_i) = \boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{b}$ so that

$$\|f_1(\boldsymbol{x}) - f_1(\boldsymbol{x}_i)\|_2 = \|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{W}\boldsymbol{x}_i\|_2 \leq \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 \leq \rho\epsilon_{\mathcal{D}}. \tag{25}$$

Moreover, combining Equations (24) and (25) results $\|f_1(\boldsymbol{x}) - f_1(\boldsymbol{x}_i)\|_2 \leq \frac{5}{6}\rho\epsilon_{\mathcal{D}'',2}$. Since f_2 $\frac{5}{6}$-robustly memorize $\mathcal{D}''$, we have

$$f(\boldsymbol{x}) = f_2(f_1(\boldsymbol{x})) = f_2(f_1(\boldsymbol{x}_i)) = y_i.$$

In particular, $f(\boldsymbol{x}) = y_i$ for any $\boldsymbol{x} \in \mathcal{B}_2(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},2})$, concluding that f is a ρ-robust memorizer $\mathcal{D}$. Regarding the number of parameters to construct f, notice that f_1 consists of $(d+1)m = \tilde{O}(d^2\rho^2)$ parameters as $m = \tilde{O}(d\rho^2)$. f_2 consists of $\tilde{O}(Nm^2) = \tilde{O}(Nd^2\rho^4)$ parameters. Therefore, f in total consists of $\tilde{O}(Nd^2\rho^4 + d^2\rho^2)$ number of parameters. This proves the theorem for the third case. ■

Here is a lemma that is used to prove Theorem 22.

Lemma 23 *For $t \geq e^5$, we have $t \geq 10 \log t$.*

Proof Define $u(t) := t - 10 \log t$ on the domain $(0, \infty)$. Then, for all $t > 10$,

$$\frac{du}{dt} = 1 - \frac{10}{t} > 0,$$

so that u is an increasing function on $(10, \infty)$. In particular,

$$u(e^5) = e^5 - 10 \log(e^5) = e^5 - 50 \geq 0$$

This concludes that $u(t) \geq 0$ for all $t \geq e^5$, or equivalently, $t \geq 10 \log t$ for all $t \geq e^5$. ■

E.4. Lemmas for Lattice Mapping

Lemma 24 (Avoiding Being Near Grid) *Let $N, d \in \mathbb{N}$ and $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N \in \mathbb{R}^d$. Then, there exists a translation vector $\boldsymbol{b} \in \mathbb{R}^d$ such that:*

$$\operatorname{dist}([\boldsymbol{x}_i]_j - b_j, \mathbb{Z}) \geq \frac{1}{2N}, \quad \forall i \in [N], j \in [d],$$

i.e., the translated points $\{\boldsymbol{x}_i - \boldsymbol{b}\}_{i\in[N]}$ are coordinate-wise far from the integer lattice.

Proof

For each coordinate $j \in [d]$, consider the set $\{x_{i,j}\}_{i\in[N]}$ of all j-th coordinate values. Let $\{x\} := x = \lfloor x \rfloor$ denote the fractional part of x. We consider the collection of fractional parts $\{\{x_{i,j}\}\}_{i\in[N]}$, and without loss of generality, assume $\{x_{1,j}\} < \{x_{2,j}\} < \cdots < \{x_{N,j}\}$. Define the maximum fractional gap as

$$g_j := \max\left(\max_{i\in[N-1]} \left(\{x_{i+1,j}\} - \{x_{i,j}\}\right),\ 1 - \{x_{N,j}\} + \{x_{1,j}\}\right).$$

We claim:

$$g_j \geq \frac{1}{N}.$$

Otherwise, we have:

$$\{x_{N,j}\} - \{x_{1,j}\} = \sum_{i=1}^{N-1} (\{x_{i+1,j}\} - \{x_{i,j}\}) < \frac{N-1}{N},$$
$$1 - \{x_{N,j}\} + \{x_{1,j}\} < \frac{1}{N} \iff \{x_{N,j}\} - \{x_{1,j}\} > \frac{N-1}{N},$$

which leads to a contradiction.

Now, we define the translation coordinate $b_j \in \mathbb{R}$ based on the location where the maximum g_j is attained. If the maximum occurs at some consecutive pair $(\{x_{i,j}\}, \{x_{i+1,j}\})$ satisfying $\{x_{i+1,j}\} - \{x_{i,j}\} = g_j$, we set

$$b_j = \frac{x_{i,j} + x_{i+1,j}}{2}.$$

Otherwise, if the maximum is attained as $1 - x_{N,j} + x_{1,j} = g_j$, we define

$$b_j = \frac{1 + x_{1,j} + x_{N,j}}{2}.$$

We define the full translation vector $b = (b_1, \ldots, b_d) \in \mathbb{R}^d$. Then the translated points $\{x_i - b\}_{i\in[N]}$ satisfy:

$$\operatorname{dist}(x_{i,j} - b_j, \mathbb{Z}) = \min(\{x_{i,j} - b_j\}, 1 - \{x_{i,j} - b_j\}) \geq \frac{1}{2} g_j \geq \frac{1}{2N}, \quad \text{for all } i \in [N], j \in [d].$$

This holds because b_j is chosen as the midpoint of the widest gap between fractional values, ensuring that all fractional parts are at least $\frac{g_j}{2}$ away from the nearest integer. Therefore, the translated points are coordinate-wise far from lattice points. ■

The following lemma shows that we can approximate the floor function using a logarithmic number of ReLU units with respect to the length of the interval of interest.

Lemma 25 (Floor Function Approximation) *For any $n \in \mathbb{N}$ and any $\gamma \in (0, 1)$, there exists a n-layer network $\overline{\mathrm{Floor}_n}$ with $4n$ number of ReLU units such that*

$$\overline{\mathrm{Floor}_n}(x) = \lfloor x \rfloor \text{ for all } x \in [0, 2^n) \text{ such that } x - \lfloor x \rfloor > \gamma.$$

Proof

To reconcile the discontinuity of the floor function with the continuity of ReLU networks, we first define a discontinuous ideal building block that exactly replicates the floor function on the target interval $[0, 2^n)$. We then approximate this building block using a continuous neural network with ReLU activations.

The ideal building block Δ is defined as:

$$\Delta(x) := \begin{cases} 2x & \text{if } x \in (0, \frac{1}{2}] \\ 2x - 1 & \text{if } x \in (\frac{1}{2}, 1] \\ 0 & \text{otherwise} \end{cases}.$$

For $n \in \mathbb{N}$, define the function Floor_n by:

$$\mathrm{Floor}_n(x) = \Delta^n(-\frac{x}{2^n} + 1) + x - 1.$$

We will show by induction that $\mathrm{Floor}_n = \lfloor x \rfloor$ for all $x \in [0, 2^n)$.

For the base case $n = 1$,

$$\mathrm{Floor}_1(x) = \Delta(-\frac{x}{2} + 1) + x - 1 = \begin{cases} 2(-\frac{x}{2} + 1) - 1 + x - 1 = 0 & \text{if } x \in [0, 1) \\ 2(-\frac{x}{2} + 1) + x - 1 = 1 & \text{if } x \in [1, 2) \\ 0 + x - 1 = x - 1 & \text{otherwise} \end{cases}.$$

This proves the base case: for all $x \in [0, 2)$, we have $\mathrm{Floor}_1(x) = \lfloor x \rfloor$.

For the inductive step, assume that $\mathrm{Floor}_n(x) = \lfloor x \rfloor$ holds for all $x \in [0,\ 2^n)$. We aim to prove that $\mathrm{Floor}_{n+1}(x) = \lfloor x \rfloor$ for all $x \in [0,\ 2^{n+1})$. Recall that:

$$\Delta(-\frac{x}{2^{n+1}}+1) = \begin{cases} -\frac{x}{2^n}+1 & \text{if } x \in [0, 2^n) \quad (\Leftrightarrow -\frac{x}{2^{n+1}}+1 \in (\frac{1}{2}, 1]) \\ -\frac{x}{2^n}+2 = -\frac{x-2^n}{2^n}+1 & \text{if } x \in [2^n, 2^{n+1}) \quad (\Leftrightarrow -\frac{x}{2^{n+1}}+1 \in (0, \frac{1}{2}]) \\ 0 & \text{otherwise} \end{cases}$$

Thus, we have

$$\begin{aligned} \mathrm{Floor}_{n+1}(x) &= \Delta^{n+1}(-\frac{x}{2^{n+1}}+1) + x - 1 \\ &= \Delta^n(\Delta(-\frac{x}{2^{n+1}}+1)) + x - 1 \\ &= \begin{cases} \Delta^n(-\frac{x}{2^n}+1) + x - 1 = \mathrm{Floor}_n(x) = \lfloor x \rfloor & \text{if } x \in [0, 2^n) \\ \Delta^n(-\frac{x-2^n}{2^n}+1) + x - 1 = \mathrm{Floor}_n(x-2^n) + 2^n = \lfloor x \rfloor & \text{if } x \in [2^n, 2^{n+1}) \\ \Delta^n(0) + x - 1 = x - 1 & \text{otherwise} \end{cases}. \end{aligned}$$

Therefore, by induction,

$$\mathrm{Floor}_n = \lfloor x \rfloor \text{ for all } x \in [0, 2^n).$$

Next, we define the ReLU approximation $\overline{\Delta_n}$ of the discontinuous block Δ as:

$$\overline{\Delta_n}(x) := 2\sigma(x) - \frac{1}{\gamma_n}\sigma\left(x - \frac{1}{2} + \gamma_n\right) + \left(\frac{1}{\gamma_n} + 2\right)\sigma\left(x - \frac{1}{2}\right) + \frac{1}{\gamma_n}\sigma\left(x - 1 + \gamma_n\right),$$

where $\gamma_n = \frac{\gamma}{2^n}$. It can be shown that:

$$\overline{\Delta_n}(x) = \Delta(x) \text{ for all } x \in [0, \frac{1}{2} - \gamma_n] \cup [\frac{1}{2}, 1 - \gamma_n].$$

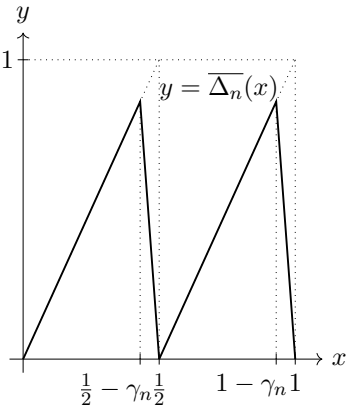


Figure 6: Plot of the ReLU-based approximation $\overline{\Delta_n}(x)$ of the ideal discontinuous building block $\Delta(x)$.

We now explain why this approximation remains valid under recursive composition up to depth n.

Let us define the variable $x' := -\frac{x}{2^n} + 1$, so that $x = 2^n(1 - x')$ and $x' \in (0, 1]$. Our target function is:

$$\mathrm{Floor}_n(x) = \Delta^n(-\frac{x}{2^n} + 1) + x - 1 = \Delta^n(x') + x - 1.$$

We are given the assumption $x - \lfloor x \rfloor > \gamma$, and we aim to express this in terms of x' to ensure $\overline{\Delta_n}^n(x') = \Delta^n(x')$. We proceed step-by-step:

$$\begin{aligned}
& x - \lfloor x \rfloor > \gamma \\
\iff & 2^n(1 - x') - \lfloor 2^n(1 - x') \rfloor > \gamma \\
\iff & -2^n x' - \lfloor -2^n x' \rfloor > \gamma \\
\iff & -2^n x' + \lceil 2^n x' \rceil > \gamma \\
\iff & 2^n x' < \lceil 2^n x' \rceil - \gamma \\
\iff & 2^n x' \in (\lceil 2^n x' \rceil - 1, \lceil 2^n x' \rceil - \gamma) \\
\iff & 2^n x' \in \bigcup_{k \in \mathbb{Z}} (k - 1, k - \gamma) \\
\iff & x' \in \bigcup_{k \in \mathbb{Z}} \left(\frac{k-1}{2^n}, \frac{k-\gamma}{2^n} \right).
\end{aligned}$$

Since $x' \in (0, 1]$, we only need to consider $k \in [2^n]$, i.e.,

$$x' \in \bigcup_{k \in [2^n]} \left(\frac{k-1}{2^n}, \frac{k-\gamma}{2^n} \right).$$

We will now prove by induction on n the following statement:

$$\overline{\Delta_n}^n(x) = \Delta^n(x) \text{ for } x \in \bigcup_{k \in [2^n]} \left(\frac{k-1}{2^n}, \frac{k-\gamma}{2^n} \right).$$

For the base case $n = 1$, by construction of $\overline{\Delta_1}(x)$, we know $\overline{\Delta_1}(x) = \Delta(x)$ for all $x \in [0, \frac{1}{2} - \frac{\gamma}{2}] \cup [\frac{1}{2}, 1 - \frac{\gamma}{2}]$, which contains the union $\bigcup_{k \in [2]} \left(\frac{k-1}{2}, \frac{k-\gamma}{2} \right)$. Hence the base case holds.

For the inductive step, assume the claim holds for n. We show it holds for $n + 1$. Let $x \in \bigcup_{k \in [2^{n+1}]} \left(\frac{k-1}{2^{n+1}}, \frac{k-\gamma}{2^{n+1}} \right)$. We analyze two cases based on $x \in [0, \frac{1}{2})$ or $x \in [\frac{1}{2}, 1)$.

First, let $x \in \bigcup_{k \in [2^n]} \left(\frac{k-1}{2^{n+1}}, \frac{k-\gamma}{2^{n+1}} \right) \subset [0, \frac{1}{2})$. Then $x < \frac{k-\gamma}{2^{n+1}} \leq \frac{1}{2} - \gamma_{n+1}$, so $\overline{\Delta_{n+1}}(x) = 2x$.

Let $y := 2x$. Then:

$$y \in \bigcup_{k \in [2^n]} \left(\frac{k-1}{2^n}, \frac{k-\gamma}{2^n} \right).$$

Therefore:

$$\begin{aligned}
\overline{\Delta_{n+1}}^{n+1}(x) = \overline{\Delta_{n+1}}^n(\overline{\Delta_{n+1}}(x)) = \overline{\Delta_{n+1}}^n(2x) &= \overline{\Delta_{n+1}}^n(y) \\
&\overset{(a)}{=} \overline{\Delta_n}^n(y)
\end{aligned}$$

$$\overset{(b)}{=}\Delta^n(y)$$
$$=\Delta^n(2x) = \Delta^{n+1}(x).$$

The equation (a) follows from the fact that $\overline{\Delta_{n+1}}(x) = \overline{\Delta_n}(x)$ for $x \in [0, \frac{1}{2} - \gamma_n] \cup [\frac{1}{2}, 1 - \gamma_n]$. The equation (b) follows directly from the induction hypothesis.

Second, let $x \in \bigcup_{k\in[2^{n+1}]\setminus[2^n]} \left(\frac{k-1}{2^{n+1}}, \frac{k-\gamma}{2^{n+1}}\right) \subset [\frac{1}{2}, 1)$. Then $\frac{1}{2} \le x < \frac{k-\gamma}{2^{n+1}} \le 1 - \gamma_{n+1}$, so $\overline{\Delta_{n+1}}(x) = 2x - 1$.

Let $y := 2x - 1$. Then:

$$y \in \bigcup_{k\in[2^{n+1}]\setminus[2^n]} \left(\frac{k - 2^n - 1}{2^n}, \frac{k - 2^n - \gamma}{2^n}\right).$$

Thus,

$$\overline{\Delta_{n+1}}^{n+1}(x) = \overline{\Delta_{n+1}}^{n}(\overline{\Delta_{n+1}}(x)) = \overline{\Delta_{n+1}}^{n}(2x - 1) = \overline{\Delta_{n+1}}^{n}(y)$$
$$\overset{(a)}{=}\overline{\Delta_n}^{n}(y)$$
$$\overset{(b)}{=}\Delta^n(y)$$
$$=\Delta^n(2x - 1) = \Delta^{n+1}(x).$$

The equation (a) follows from the fact that $\overline{\Delta_{n+1}}(x) = \overline{\Delta_n}(x)$ for $x \in [0, \frac{1}{2} - \gamma_n] \cup [\frac{1}{2}, 1 - \gamma_n]$. The equation (b) follows directly from the induction hypothesis.

Therefore, by induction, we have shown that

$$\overline{\Delta}_n^n(x') = \Delta^n(x') \quad \text{for all } x' \in \bigcup_{k\in[2^n]} \left(\frac{k-1}{2^n}, \frac{k-\gamma}{2^n}\right).$$

We now define the ReLU-based floor approximation by

$$\overline{\mathrm{Floor}_n}(x) := \overline{\Delta}_n^n\left(-\frac{x}{2^n} + 1\right) + x - 1.$$

Recall that the ideal target function is given by

$$\mathrm{Floor}_n(x) = \Delta^n\left(-\frac{x}{2^n} + 1\right) + x - 1,$$

and let $x' := -\frac{x}{2^n} + 1$. When $x - \lfloor x \rfloor > \gamma$, the value x' satisfies

$$x' \in \bigcup_{k\in[2^n]} \left(\frac{k-1}{2^n}, \frac{k-\gamma}{2^n}\right),$$

so that $\overline{\Delta}_n^n(x') = \Delta^n(x')$ by the result above.

Therefore, we conclude:

$$\overline{\mathrm{Floor}_n}(x) = \mathrm{Floor}_n(x) = \lfloor x \rfloor \quad \text{for all } x \in [0, 2^n) \text{ such that } x - \lfloor x \rfloor > \gamma.$$

■

E.5. Dimension Reduction via Careful Analysis of the Johnson-Lindenstrauss Lemma

We begin with a lemma that states a concentration of the length of the projection.

Lemma 26 (Lemma 15.2.2, Matousek [12]) *For a unit vector $\boldsymbol{x} \in S^{d-1}$, let*

$$\phi(\boldsymbol{x}) = (x_1, x_2, \cdots, x_m)$$

be the mapping of $\boldsymbol{x}$ onto the subspace spanned by the first k coordinates. Consider $\boldsymbol{x} \in S^{d-1}$ chosen uniformly at random. Then, there exists β such that $\|\phi(\boldsymbol{x})\|_2$ is sharply concentrated around β,

$$\mathbb{P}[\|\phi(\boldsymbol{x})\|_2 \geq \beta + t] \leq 2e^{-t^2 d/2} \text{ and } \mathbb{P}[\|\phi(\boldsymbol{x})\|_2 \leq \beta - t] \leq 2e^{-t^2 d/2},$$

where for $m \geq 10 \log d$, we have $\beta \geq \frac{1}{2}\sqrt{\frac{m}{d}}$.

Based on the above concentration inequality, we state the Johnson-Lindenstrauss lemma, in a version which reflects the benefit on the ratio of the norm preserved when the projecting dimension increases. The proof follows that of Theorem 15.2.1 in Matousek [12] with a slight modification.

Lemma 27 *For $N \geq 2$, let $X \subseteq \mathbb{R}^d$ be an N point set. Then, for any $\alpha \in (0,1)$ and $24\alpha^{-2}\log N \leq m \leq d$, there exists a 1-Lipschitz linear mapping $\phi : \mathbb{R}^d \to \mathbb{R}^m$ and $\beta > 0$ such that*

$$(1-\alpha)\beta \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_2 \leq \left\|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\right\|_2 \leq (1+\alpha)\beta \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_2, \tag{26}$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in X$. Moreover, $\beta \geq \frac{1}{2}\sqrt{\frac{m}{d}}$ whenever $m \geq 10 \log d$.

Proof If $\boldsymbol{x} = \boldsymbol{x}'$, the inequality trivially holds for any ϕ. Hence, it suffices to find ϕ that satisfies Equation (26) for all $\boldsymbol{x}, \boldsymbol{x}' \in X$ with $\boldsymbol{x} \neq \boldsymbol{x}'$. Consider a random k-dimensional subspace L, and ϕ be a projection onto L. For any fixed $\boldsymbol{x} \neq \boldsymbol{x}' \in X$, Theorem 26 implies that $\left\|\phi(\frac{\boldsymbol{x}-\boldsymbol{x}'}{\|\boldsymbol{x}-\boldsymbol{x}'\|_2})\right\|_2$ is concentrated around some constant β. i.e.

$$\mathbb{P}\left[\left\|\phi\left(\frac{\boldsymbol{x}-\boldsymbol{x}'}{\|\boldsymbol{x}-\boldsymbol{x}'\|_2}\right)\right\|_2 \geq (1+\alpha)\beta\right] \leq 2e^{-\alpha^2\beta^2 d/2} \overset{(a)}{\leq} 2e^{-\alpha^2 m/8} \overset{(b)}{\leq} 2e^{-3\log N} = \frac{2}{N^3} \overset{(c)}{\leq} \frac{1}{N^2},$$

where we use $\beta \geq \frac{1}{2}\sqrt{\frac{m}{d}}$ at (a), $m \geq 24\alpha^{-2}\log N$ at (b), and $N \geq 2$ at (c). Similarly,

$$\mathbb{P}\left[\left\|\phi\left(\frac{\boldsymbol{x}-\boldsymbol{x}'}{\|\boldsymbol{x}-\boldsymbol{x}'\|_2}\right)\right\|_2 \leq (1-\alpha)\beta\right] \leq \frac{1}{N^2}.$$

By linearity of ϕ, we have $\phi(\boldsymbol{x} - \boldsymbol{x}') = \phi(\boldsymbol{x}) - \phi(\boldsymbol{x})$. Taking the union bound over the two probability bounds above, the following event happens with probability at most $2/N^2$:

$$\left\|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\right\|_2 \geq (1+\alpha)\beta \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_2 \text{ or } \left\|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\right\|_2 \leq (1-\alpha)\beta \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_2. \tag{27}$$

Next, we take a union bound over all $\frac{N(N-1)}{2}$ pairs $\boldsymbol{x}, \boldsymbol{x}' \in X$ with $\boldsymbol{x} \neq \boldsymbol{x}'$. Then, the probability that Equation (27) happens for any $\boldsymbol{x}, \boldsymbol{x}' \in X$ with $\boldsymbol{x} \neq \boldsymbol{x}'$ is at most $\frac{2}{N^2} \times \frac{N(N-1)}{2} = 1 - \frac{1}{N} < 1$.

Hence, there exists a k-dimensional subspace L such that Equation (27) does not hold for any pair of $\boldsymbol{x}, \boldsymbol{x}' \in X$. In other words, there exists a k-dimensional subspace L such that

$$(1-\alpha)\beta \left\|\boldsymbol{x}-\boldsymbol{x}'\right\|_2 \leq \left\|\phi(\boldsymbol{x})-\phi(\boldsymbol{x}')\right\|_2 \leq (1+\alpha)\beta \left\|\boldsymbol{x}-\boldsymbol{x}'\right\|_2,$$

for all $\boldsymbol{x} \neq \boldsymbol{x}'$. By Theorem 26, $\beta \geq \frac{1}{2}\sqrt{\frac{m}{d}}$ whenever $m \geq 10 \log d$. This concludes the lemma. ■

Proposition 28 (Lipschitz Projection with Separation) *For $N \geq 2$, let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \in \mathcal{D}_{d,N,C}$. For any $\alpha \in (0,1)$ and $24\alpha^{-2}\log N \leq m \leq d$, there exists 1-Lipschitz linear mapping $\phi : \mathbb{R}^d \to \mathbb{R}^m$ and $\beta > 0$ such that $\mathcal{D}' := \{(\phi(\boldsymbol{x}_i), y_i)\}_{i=1}^N \in \mathcal{D}_{m,N,C}$ satisfies*

$$\epsilon'_{\mathcal{D}} \geq (1-\alpha)\beta\epsilon_{\mathcal{D}}.$$

In particular, $\mathcal{D}' \in \mathcal{D}_{m,N,C}$ whenever $\mathcal{D} \in \mathcal{D}_{d,N,C}$. Moreover, $\beta \geq \frac{1}{2}\sqrt{\frac{m}{d}}$ whenever $m \geq 10\log d$.

Proof Let $X = \{\boldsymbol{x}_i\}_{i=1}^N$. By Theorem 27, there exists 1-Lipchitz linear mapping $\phi : \mathbb{R}^d \to \mathbb{R}^m$ and $\beta > 0$ such that

$$(1-\alpha)\beta \left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2 \leq \left\|\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{x}_j)\right\|_2 \leq (1+\alpha)\beta \left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2 \tag{28}$$

for all $i, j \in [N]$.

The inequality $\epsilon_{\mathcal{D}',2} \geq (1-\alpha)\beta\epsilon_{\mathcal{D},2}$ follows from the inequality from Theorem 27. In particular,

$$\begin{aligned}
\epsilon_{\mathcal{D}',2} &= \frac{1}{2}\min\{\left\|\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{x}_j)\right\|_2 \mid i,j \in [N] \text{ and } y_i \neq y_j\} \\
&\overset{(a)}{\geq} \frac{1}{2}\min\{(1-\alpha)\beta\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2 \mid i,j \in [N] \text{ and } y_i \neq y_j\} \\
&= (1-\alpha)\beta \times \frac{1}{2}\min\{\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2 \mid i,j \in [N] \text{ and } y_i \neq y_j\} \\
&= (1-\alpha)\beta\epsilon_{\mathcal{D},2},
\end{aligned}$$

where we use Equation (28) at (a).

We next show $\mathcal{D}' \in \mathcal{D}_{m,N,C}$ whenever $\mathcal{D} \in \mathcal{D}_{d,N,C}$. To show this, we need to prove $\phi(\boldsymbol{x}_i) \neq \phi(\boldsymbol{x}_j)$ for all $i \neq j$. Since $1-\alpha > 0$ and $\beta > 0$, we have $\left\|\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{x}_j)\right\|_2 \geq (1-\alpha)\beta\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2 > 0$ whenever $\boldsymbol{x}_i \neq \boldsymbol{x}_j$. Moreover, $\mathcal{D} \in \mathcal{D}_{d,N,C}$ indicates that $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ whenever $i \neq j$. All together, we have $\phi(\boldsymbol{x}_i) \neq \phi(\boldsymbol{x}_j)$ for all $i \neq j$ so that $\mathcal{D}' \in \mathcal{D}_{m,N,C}$. ■

Appendix F. Extension to ℓ_p-norm

In this section, we extend the previous results on ℓ_2-norm to arbitrary p-norm, where $p \in [1, \infty]$.

In the following, we use $\mathrm{dist}_p(\cdot, \cdot)$ to denote the ℓ_p-norm distance between two points, a point and a set, or two sets. For the case $d = 1$, we omit the notation p since every ℓ_p-norm in 1-dimension denotes the absolute value.

We denote $\mathcal{B}_p(\boldsymbol{x}, \mu) = \left\{\boldsymbol{x}' \in \mathbb{R}^d \middle| \|\boldsymbol{x}' - \boldsymbol{x}\|_p < \mu\right\}$ an open ℓ_p-ball centered at $\boldsymbol{x}$ with a radius μ.

Definition 29 *For $\mathcal{D} \in \mathcal{D}_{d,N,C}$, the separation constant $\epsilon_{\mathcal{D},p}$ under ℓ_p-norm is defined as*

$$\epsilon_{\mathcal{D},p} := \frac{1}{2} \min \left\{ \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_p \,|\, (\boldsymbol{x}_i, y_i), (\boldsymbol{x}_j, y_j) \in \mathcal{D},\ y_i \neq y_j \right\}.$$

As we consider $\mathcal{D}$ with $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ for all $i \neq j$, we have $\epsilon_{\mathcal{D},p} > 0$. Next, we define robust memorization under ℓ_p-norm.

Definition 30 *For $\mathcal{D} \in \mathcal{D}_{d,N,C}$, $p \in [1, \infty]$, and a given robustness ratio $\rho \in (0, 1)$, define the robustness radius as $\mu = \rho\epsilon_{\mathcal{D},p}$. We say that a function $f : \mathbb{R}^d \to \mathbb{R}$ ρ-robustly memorizes $\mathcal{D}$ under the ℓ_p-norm if*

$$f(\boldsymbol{x}') = y_i, \quad \text{for all } (\boldsymbol{x}_i, y_i) \in \mathcal{D} \text{ and } \boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}_i, \mu),$$

and $\mathcal{B}_p(\boldsymbol{x}_i, \mu)$ is referred as the robustness ball of $\boldsymbol{x}_i$.

Similarly, we extend the notion of ρ-robust memorization error to ℓ_p-norm.

Definition 31 *Let $\mathcal{D} \in \mathcal{D}_{d,N,C}$ be a class(or point)-separated dataset. The ρ-robust error of a network $f : \mathbb{R}^d \to \mathbb{R}$ on $\mathcal{D}$ under the ℓ_p-norm is defined as*

$$\mathcal{L}_{\rho,p}(f, \mathcal{D}) = \max_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}} \mathbb{P}_{\boldsymbol{x}' \sim \mathit{Unif}(\mathcal{B}_p(\boldsymbol{x}_i, \mu))}[f(\boldsymbol{x}') \neq y_i], \text{ where } \mu = \rho\epsilon_{\mathcal{D},p} \text{ (or } \mu = \rho\epsilon'_{\mathcal{D},p}).$$

Lemma 32 (Inclusion Between Balls) *Let $0 < p < q \leq \infty$. Then, for any $\boldsymbol{x} \in \mathbb{R}^d$ and $\mu > 0$,*

$$\mathcal{B}_p(\boldsymbol{x}, \mu) \subseteq \mathcal{B}_q(\boldsymbol{x}, \mu) \subseteq \mathcal{B}_p(\boldsymbol{x}, d^{\frac{1}{p} - \frac{1}{q}}\mu),$$

or equivalently,

$$\mathcal{B}_q(\boldsymbol{x}, d^{\frac{1}{q} - \frac{1}{p}}\mu) \subseteq \mathcal{B}_p(\boldsymbol{x}, \mu) \subseteq \mathcal{B}_q(\boldsymbol{x}, \mu).$$

For any $p \in [1, \infty]$, let us denote $\gamma_p(d) := d^{\left|\frac{1}{2} - \frac{1}{p}\right|}$ throughout this section. For $0 < p < q \leq \infty$, we have

$$\epsilon_{\mathcal{D},q} \leq \epsilon_{\mathcal{D},p} \leq d^{\frac{1}{p} - \frac{1}{q}}\epsilon_{\mathcal{D},q}, \tag{29}$$

since $\|\boldsymbol{x}\|_q \leq \|\boldsymbol{x}\|_p \leq d^{\frac{1}{p} - \frac{1}{q}} \|\boldsymbol{x}\|_q$. In particular, we have

$$\epsilon_{\mathcal{D},p} \leq \epsilon_{\mathcal{D},2} \qquad \text{when } p \geq 2, \tag{30}$$

$$\epsilon_{\mathcal{D},p} \leq \gamma_p(d)\epsilon_{\mathcal{D},2} \qquad \text{when } p < 2. \tag{31}$$

F.1. Extension of Necessity Condition to ℓ_p-norm

F.1.1. Results Using Careful Analysis of ℓ_p-distance

Theorem 33 *Let $N \geq 2, d \geq 1$ and $\rho \in (1/2, 1)$. Then, there exists a point separated $\mathcal{D} \in \mathcal{D}_{d,N,2}$ such that any neural network that ρ-robustly memorizes $\mathcal{D}$ under ℓ_∞-norm should have the first hidden layer width at least* $\min\{d, N-1\}$.

Proof We first prove the statement for the case $N - 1 \leq d$. To prove the statement for this case, we construct $\mathcal{D} \in \mathcal{D}_{d,N,2}$ in which ρ-robustly memorizing the dataset requires the first hidden layer width at least $N - 1$.

Let $\mathcal{D} = \{(\boldsymbol{e}_j, 2)\}_{j\in[N-1]} \cup \{(\mathbf{0}, 1)\}$. Then, $\mathcal{D}$ has a separation constant $\epsilon_{\mathcal{D},\infty} = 1/2$ under ℓ_∞-norm. Let f be a ρ-robust memorizer of $\mathcal{D}$ under ℓ_∞-norm whose first hidden layer width is m. Let $\boldsymbol{W} \in \mathbb{R}^{d\times m}$ denote the first hidden weight matrix. Suppose for a contradiction, $m < N - 1$.

Let $\mu = \rho\epsilon_{\mathcal{D},\infty}$ denote the robustness radius. Then, f has to distinguish every point in each $B_\mu(\boldsymbol{e}_j)$ from every point in $B_\mu(\mathbf{0})$ for all $j \in [N-1]$. Therefore, for $\boldsymbol{x} \in B_\infty(\boldsymbol{e}_j, \mu)$ and $\boldsymbol{x}' \in B_\infty(\mathbf{0}, \mu)$, we have

$$\boldsymbol{W}\boldsymbol{x} \neq \boldsymbol{W}\boldsymbol{x}',$$

or equivalently, $\boldsymbol{x} - \boldsymbol{x}' \notin \mathrm{Null}(\boldsymbol{W})$. Moreover

$$B_\infty(\boldsymbol{e}_j, \mu) - B_\infty(\mathbf{0}, \mu) := \{\boldsymbol{x} - \boldsymbol{x}' : \boldsymbol{x} \in B_\infty(\boldsymbol{e}_j, \mu) \text{ and } \boldsymbol{x}' \in B_\infty(\mathbf{0}, \mu)\} = B_\infty(\boldsymbol{e}_j, 2\mu).$$

Hence, it is necessary to have $B_\infty(\boldsymbol{e}_j, 2\mu) \cap \mathrm{Null}(\boldsymbol{W}) = \emptyset$ for all $j \in [N-1]$, or equivalently,

$$\mathrm{dist}_\infty(\boldsymbol{e}_j, \mathrm{Null}(\boldsymbol{W})) \geq 2\mu \tag{32}$$

for all $j \in [N-1]$.

Since $\dim \mathrm{Col}(W^\top) \leq \dim \mathbb{R}^m = m$, we have $\dim \mathrm{Null}(W) \geq d - m$. Using Theorem 37, we can upper bounds the maximum possible distance between $\{\boldsymbol{e}_j\}_{j\in[N-1]} \subseteq \mathbb{R}^d$ and arbitrary subspace of a fixed dimension.

Take $Z \subseteq \mathrm{Null}(W)$ such that $\dim Z = d - m$ and substitute $d = d$, $t = N - 1$, $k = d - m$ and $Z = Z$ into Theorem 37. The assumptions $t \leq d$ for the lemma are satisfied since $N - 1 \leq d$. The additional assumption $k \geq d - t + 1$ is equivalent to $d - m \geq d - (N-1) + 1$ and is satisfied since $m < N - 1$. Therefore, we have

$$\min_{j\in[N-1]} \mathrm{dist}_\infty(\boldsymbol{e}_j, Z) \leq \frac{1}{2}.$$

By combining the above inequality with Equation (32),

$$2\mu \leq \min_{j\in[N-1]} \mathrm{dist}_\infty(\boldsymbol{e}_j, \mathrm{Null}(W)) \overset{(a)}{\leq} \min_{j\in[N-1]} \mathrm{dist}_\infty(\boldsymbol{e}_j, Z) \leq \frac{1}{2}, \tag{33}$$

where (a) is due to $Z \subseteq \mathrm{Null}(W)$. Since $\epsilon_{\mathcal{D},\infty} = 1/2$, we have $2\mu = 2\rho\epsilon_{\mathcal{D},\infty} = \rho$ so that Equation (33) becomes $\rho \leq 1/2$. This contradicts our assumption $\rho \in (1/2, 1)$, and therefore the width requirement $m \geq N - 1$ is necessary. This concludes the proof for the case $N - 1 \leq d$.

For the case $N - 1 > d$, we construct the first $d + 1$ data points as for the case $N = d + 1$. For the remaining $N - d - 1$ data points, we set them sufficiently distant from the first $d + 1$ data points

to keep $\epsilon_{\mathcal{D},\infty} = 1/2$. In particular, we can set $\boldsymbol{x}_{d+2} = 2\boldsymbol{e}_1, \boldsymbol{x}_{d+3} = 3\boldsymbol{e}_1, \cdots, \boldsymbol{x}_N = (N-d)\boldsymbol{e}_1$ and $y_{d+2} = y_{d+3} = \cdots = y_N = 2$. Compared to the case $N = d+1$, we have $\epsilon_{\mathcal{D},\infty}$ unchanged while having more data points to memorize. By the necessity for the case $N = d+1$, this dataset also requires the first hidden layer width at least $(d+1) - 1 = d$. This concludes the statement for the case $N - 1 > d$.

Combining the result of the two cases $N - 1 \le d$ and $N - 1 > d$ concludes the proof of the theorem. ■

Theorem 34 *For $p \in [1, \infty]$, let $\rho \in \left(0, \left(1 - \frac{1}{d}\right)^{1/p}\right]$. Suppose for any $\mathcal{D} \in \mathcal{D}_{d,N,2}$ there exists $f \in \mathcal{F}_{d,P}$ that ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm. Then, the number of parameters P must satisfy $P = \Omega(\sqrt{\frac{N}{1-\rho^p}})$.*

Proof The main idea of the proof is the same as Theorem 5. We construct $\lfloor \frac{N}{2} \rfloor \times \lfloor \frac{1}{1-\rho^p} \rfloor$ number of data points that can be shattered by $\mathcal{F}_{d,P}$. This proves $\text{VC-dim}(\mathcal{F}_{d,P}) \ge \lfloor \frac{N}{2} \rfloor \times \lfloor \frac{1}{1-\rho^p} \rfloor = \Omega(N/(1-\rho^p))$. Since $\text{VC-dim}(\mathcal{F}_{d,P}) = O(P^2)$, this proves $P = \Omega(\sqrt{N/(1-\rho^p)})$.

For simplicity of the notation, let us denote $k := \lfloor \frac{1}{1-\rho^p} \rfloor$. To prove the lower bound on the VC-dimension, we construct $k \times \lfloor \frac{N}{2} \rfloor$ points in $\mathbb{R}^d$ that can be shattered by $\mathcal{F}_{d,P}$. As in the proof of Theorem 4, we define $\lfloor \frac{N}{2} \rfloor \times k$ number of points as $\lfloor \frac{N}{2} \rfloor$ groups, where each group consists of k points.

We start by constructing the first group. Since $\rho \in (0, \left(\frac{d-1}{d}\right)^{1/p}]$, we have $k = \lfloor \frac{1}{1-\rho^p} \rfloor \in [1, d]$. The first group $\mathcal{X}_1 := \{\boldsymbol{e}_j\}_{j=1}^k \subseteq \mathbb{R}^d$ is defined as the set of the first k vectors in the standard basis of $\mathbb{R}^d$. The remaining $\lfloor \frac{N}{2} \rfloor - 1$ groups are simply constructed as a translation of $\mathcal{X}_1$. In particular, for $l \in [\lfloor \frac{N}{2} \rfloor]$, we define

$$\mathcal{X}_l := \boldsymbol{c}_l + \mathcal{X}_1 = \{\boldsymbol{c}_l + \boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{X}_1\}$$

where $\boldsymbol{c}_l := 2d^2(l-1) \times \boldsymbol{e}_1$ ensures that each group is sufficiently far from one another. Note that $\boldsymbol{c}_1 = \boldsymbol{0}$ ensures $\mathcal{X}_1$ also satisfies the consistency of the notation. Now, define $\mathcal{X} = \cup_{l \in [\lfloor N/2 \rfloor]} \mathcal{X}_l$, the union of all $\lfloor \frac{N}{2} \rfloor$ groups which consists of $k \times \lfloor \frac{N}{2} \rfloor$ points.

We claim that if for any $\mathcal{D} \in \mathcal{D}_{d,N,2}$, there exists $f \in \mathcal{F}_{d,P}$ that ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm, then $\mathcal{X}$ is shattered by $\mathcal{F}_{d,P}$. To prove the claim, suppose we are given arbitrary label $\mathcal{Y} = \{y_{l,j}\}_{l \in [\lfloor N/2 \rfloor], j \in [d]}$ of $\mathcal{X}$, where $y_{l,j} \in \{\pm 1\}$ denotes the label for $\boldsymbol{x}_{l,j} := \boldsymbol{c}_l + \boldsymbol{e}_j \in \mathcal{X}$. Given the label $\mathcal{Y}$, we construct $\mathcal{D} \in \mathcal{D}_{d,N,2}$ such that whenever $f \in \mathcal{F}_{d,P}$ ρ-robustly memorize $\mathcal{D}$ under ℓ_p-norm, then its affine translation $f' = 2f - 3 \in \mathcal{F}_{d,P}$ satisfies $f'(\boldsymbol{x}_{l,j}) = y_{l,j}$ for all $\boldsymbol{x}_{l,j} \in \mathcal{X}$.

For each $l \in [\lfloor N/2 \rfloor]$, let $J_l^+ = \{j \in [k] \mid y_{l,j} = +1\}$ and $J_l^- = \{j \in [k] \mid y_{l,j} = -1\}$. Define

$$\boldsymbol{x}_{2l-1} = \boldsymbol{c}_l + \sum_{j \in J_l^+} \boldsymbol{e}_j - \sum_{j \in J_l^-} \boldsymbol{e}_j$$
$$\boldsymbol{x}_{2l} = \boldsymbol{c}_l + \sum_{j \in J_l^-} \boldsymbol{e}_j - \sum_{j \in J_l^+} \boldsymbol{e}_j$$

Furthermore, define $y_{2l-1} = 2, y_{2l} = 1$ and let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i\in[N]} \in \mathcal{D}_{d,N,2}$. To consider the separation $\epsilon_{\mathcal{D},2}$, notice that

$$\|\boldsymbol{x}_{2l-1} - \boldsymbol{x}_{2l}\|_p = \left\| 2\left(\sum_{j\in J_l^+} \boldsymbol{e}_j - \sum_{j\in J_l^-} \boldsymbol{e}_j \right) \right\|_p \overset{(a)}{=} 2k^{1/p},$$

where (a) is due to $J_l^+ \cap J_l^- = \emptyset$ and $J_l^+ \cup J_l^- = [k]$. For $l \neq l'$,

$$\begin{aligned} d_p(\boldsymbol{x}_{2l-1}, \boldsymbol{x}_{2l'}) &\overset{(a)}{\geq} d_p(\boldsymbol{c}_l, \boldsymbol{c}_{l'}) - d_p(\boldsymbol{c}_l, \boldsymbol{x}_{2l-1}) - d_p(\boldsymbol{c}_{l'}, \boldsymbol{x}_{2l'}) \\ &\overset{(b)}{\geq} 2d^2 - k^{1/p} - k^{1/p} \\ &\overset{(c)}{\geq} 2d^2 - 2d^{1/p} \\ &\overset{(d)}{\geq} 2d^{1/p} \\ &\overset{(e)}{\geq} 2k^{1/p}, \end{aligned}$$

where (a) is by the triangle inequality under ℓ_p-norm (namely, the Minkowski inequality), (b) uses $d_p(\boldsymbol{c}_l, \boldsymbol{x}_{2l-1}) = d_p(\boldsymbol{c}_{l'}, \boldsymbol{x}_{2l'}) = k^{1/p}$, (c),(e) is by $k \leq d$, and (d) holds for all $d \geq 2$ and $p \geq 1$. Thus, we have $\epsilon_{\mathcal{D},p} \geq k^{1/p}$.

Take $f \in \mathcal{F}_{d,P}$ that ρ-robustly memorize $\mathcal{D}$. We first lower bound the robustness radius μ. Since $t \overset{\phi}{\mapsto} \sqrt[p]{\frac{t-1}{t}}$ is an strictly increasing function from $t \geq 1$ onto $[0, 1)$ [2] , it has a well defined inverse mapping $\phi^{-1} : [0, 1) \to [1, \infty)$ defined as $\phi^{-1}(\rho) = \frac{1}{1-\rho^p}$. Therefore,

$$\rho = \phi(\phi^{-1}(\rho)) = \phi\left(\frac{1}{1-\rho^p}\right) \geq \phi\left(\lfloor \frac{1}{1-\rho^p} \rfloor\right) = \phi(k) = \sqrt[p]{\frac{k-1}{k}}.$$

Since $\epsilon_{\mathcal{D},p} \geq k^{1/p}$ and $\rho \geq (\frac{k-1}{k})^{1/p}$, we have $\mu = \rho\epsilon_{\mathcal{D},p} \geq \rho k^{1/p} \geq (k-1)^{1/p}$. Thus, every f that ρ-robustly memorizes $\mathcal{D}$ must also memorize $(k-1)^{1/p}$ radius open ℓ_p-ball around each point in $\mathcal{D}$ as the same label as the data point.

Moreover, for $\boldsymbol{x}_{l,j} \in \mathcal{X}$ with positive label $y_{l,j} = +1$, we have

$$\begin{aligned} \|\boldsymbol{x}_{l,j} - \boldsymbol{x}_{2l-1}\|_p &= \left\| (\boldsymbol{c}_l + \boldsymbol{e}_j) - (\boldsymbol{c}_l + \sum_{j'\in J_l^+} \boldsymbol{e}_{j'} - \sum_{j'\in J_l^-} \boldsymbol{e}_{j'}) \right\|_p \\ &= \left\| \sum_{\substack{j'\in J_l^+ \\ j'\neq j}} \boldsymbol{e}_{j'} - \sum_{j'\in J_l^-} \boldsymbol{e}_{j'} \right\|_p \end{aligned}$$

2. ϕ is a composition of two strictly increasing one-to-one corresponding functions $t \mapsto \frac{t-1}{t}$ from $[1, \infty)$ onto $[0, 1)$ and $u \mapsto \sqrt[p]{u}$ from $[0, 1)$ onto $[0, 1)$

$$= (k-1)^{1/p}.$$

Take a sequence of points $\{\boldsymbol{z}_n\}_{n\in\mathbb{N}}$ such that $\boldsymbol{z}_n \to \boldsymbol{x}_{l,j}$ as $n \to \infty$ [3] and

$$\|\boldsymbol{z}_n - \boldsymbol{x}_{2l-1}\|_p < (k-1)^{1/p},$$

for all $n \in \mathbb{N}$. In particular,

$$\boldsymbol{z}_n := \frac{n-1}{n}\boldsymbol{x}_{l,j} + \frac{1}{n}\boldsymbol{x}_{2l-1}$$

satisfies such properties. Then, we have $f(\boldsymbol{z}_n) = f(\boldsymbol{x}_{2l-1}) = 2$ for all $n \in \mathbb{N}$. Moreover, by the continuity of f (under the usual topology),

$$f(\boldsymbol{x}_{l,j}) = f(\lim_{n\to\infty} \boldsymbol{z}_n) = \lim_{n\to\infty} f(\boldsymbol{z}_n) = \lim_{n\to\infty} 2 = 2.$$

Similarly, for $\boldsymbol{x}_{l,j}$with negative label $y_{l,j} = -1$, we have $\|\boldsymbol{x}_{l,j} - \boldsymbol{x}_{2l}\|_p = (k-1)^{1/p}$, so that $f(\boldsymbol{x}_{l,j}) = 1$.

Since we can adjust the weight and the bias of the last hidden layer, $\mathcal{F}_{d,P}$ is closed under affine transformation; that is, $af + b \in \mathcal{F}_{d,P}$ whenever $f \in \mathcal{F}_{d,P}$. In particular, $f' := 2f - 3 \in \mathcal{F}_{d,P}$. This f' satisfies $f'(\boldsymbol{x}_{l,j}) = 2f(\boldsymbol{x}_{l,j}) - 3 = 2 \cdot 2 - 3 = +1$ whenever $y_{l,j} = +1$ and $f'(\boldsymbol{x}_{l,j}) = 2f(\boldsymbol{x}_{l,j}) - 3 = 2 \cdot 1 - 3 = -1$ whenever $y_{l,j} = -1$. Thus, $\operatorname{sign} \circ f'$ perfectly classify $\mathcal{X}$ with the label $\mathcal{Y}$. Since we can take such $f' \in \mathcal{F}_{d,P}$ given an arbitrary label $\mathcal{Y}$ of $\mathcal{X}$, it follows that $\mathcal{F}_{d,P}$ shatters $\mathcal{X}$, concluding the proof of the theorem. ■

F.1.2. RESULTS USING INCLUSION BETWEEN BALLS

Proposition 35 *There exists $\mathcal{D} \in \mathcal{D}_{d,N,2}$ such that any neural network $f : \mathbb{R}^d \to \mathbb{R}$ that ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm must have the first hidden layer width at least*

- $\rho^2 \min\{N-1, d\}$ *if* $p \geq 2$
- $\left(\frac{\rho^2}{\gamma_p(d)}\right)^2 \min\{N-1, d\}$ *if* $1 \leq p < 2$

Proof We take $\mathcal{D}$ the same dataset as in Theorem 4. Recall that in the proof of Theorem 4, we take the dataset $\mathcal{D} = \{\boldsymbol{e}_j, 2\}_{j\in[N-1]} \cup \{\boldsymbol{0}, 1\}$ when $N \leq d+1$, with additional data points $(2\boldsymbol{e}_1, 2), (3\boldsymbol{e}_1, 2), \cdots, ((N-d)\boldsymbol{e}_1, 2)$ when $N > d+1$. This has a separation $\epsilon_{\mathcal{D},p} = \frac{1}{2}$ under ℓ_p-norm for all $p \geq 1$, on the both case $N \leq d+1$ and $N > d+1$. Let f be a neural network that robustly memorizes $\mathcal{D}$ under ℓ_p-norm. Since $\epsilon_{\mathcal{D},p} = \epsilon_{\mathcal{D},2}$, the robustness radius μ under ℓ_2-norm satisfies $\mu = \rho\epsilon_{\mathcal{D},p} = \rho\epsilon_{\mathcal{D},2}$. With this in mind, we now prove the proposition. The statement of the proposition consists of two parts, $p \geq 2$ and $1 \leq p < 2$.

Part I: $p \geq 2$**.** First, we prove the result under $p \geq 2$ Robust memorization under ℓ_p-norm implies

$$f(\boldsymbol{x}) = y_i \text{ for all } (\boldsymbol{x}_i, y_i) \in \mathcal{D} \text{ and } \boldsymbol{x} \in \mathcal{B}_p(\boldsymbol{x}_i, \mu),$$

where $\mu = \rho\epsilon_{\mathcal{D},p} = \rho\epsilon_{\mathcal{D},2}$. For $p \geq 2$, we have $\mathcal{B}_2(\boldsymbol{x}_i, \mu) \leq \mathcal{B}_p(\boldsymbol{x}_i, \mu)$ by Theorem 32. Thus,

$$f(\boldsymbol{x}) = y_i \text{ for all } (\boldsymbol{x}_i, y_i) \in \mathcal{D} \text{ and } \boldsymbol{x} \in \mathcal{B}_2(\boldsymbol{x}_i, \mu).$$

Since $\mu = \rho\epsilon_{\mathcal{D},2}$ this implies that f ρ-robustly memorize $\mathcal{D}$ under ℓ_2-norm. By Theorem 4, f should have the first hidden layer width at least $\rho^2 \min\{N-1, d\}$.

3. We consider the convergence of the sequence on the usual topology induced by ℓ_2-norm.

Part II: $1 \le p < 2$**.** Next, we prove the result under $1 \le p < 2$. Robust memorization under ℓ_p-norm implies

$$f(\boldsymbol{x}) = y_i \text{ for all } (\boldsymbol{x}_i, y_i) \in \mathcal{D} \text{ and } \boldsymbol{x} \in \mathcal{B}_p(\boldsymbol{x}_i, \mu),$$

where $\mu = \rho\epsilon_{\mathcal{D},p} = \rho\epsilon_{\mathcal{D},2}$. For $1 \le p < 2$, we have $\mathcal{B}_2(\boldsymbol{x}, d^{\frac{1}{2}-\frac{1}{p}}\mu) \subseteq \mathcal{B}_p(\boldsymbol{x}_i, \mu)$ by applying $p = p$ and $q = 2$ to Theorem 32. Since $\gamma_p(d) = d^{\frac{1}{p}-\frac{1}{2}}$, we have $\mathcal{B}_2(\boldsymbol{x}_i, \mu/\gamma_p(d)) \subseteq \mathcal{B}_p(\boldsymbol{x}_i, \mu)$. In particular, f memorize every $\mu/\gamma_p(d)$ neighbor around the data point under ℓ_2-norm. Let

$$\rho' := \frac{\mu/\gamma_p(d)}{\epsilon_{\mathcal{D},2}} = \frac{\rho\epsilon_{\mathcal{D},2}/\gamma_p(d)}{\epsilon_{\mathcal{D},2}} = \frac{\rho}{\gamma_p(d)}$$

Then, f memorize every $\mu/\gamma_p(d) = \rho'\epsilon_{\mathcal{D},2}$ radius neighbor around each data point under ℓ_2-norm. In other words, f ρ'-robustly memorize $\mathcal{D}$ under ℓ_2-norm. By Theorem 4, f should have the first hidden layer width at least $(\rho')^2 \min\{N-1, d\}$. Putting back $\rho' = \frac{\rho}{\gamma_p(d)}$ concludes the desired statement.

■

F.1.3. LEMMAS FOR APPENDIX F.1

Lemma 36 *Let $\{\boldsymbol{e}_j\}_{j\in[d]} \subseteq \mathbb{R}^d$ denote the standard basis in $\mathbb{R}^d$. Then, for any k-dimensional subspace Z of $\mathbb{R}^d$ with $k \ge 1$ we have,*

$$\min_{j\in[d]} \operatorname{dist}_\infty(\boldsymbol{e}_j, Z) \le \frac{1}{2}.$$

Proof For any subspace Z' of Z, we have

$$\min_{j\in[d]} \operatorname{dist}_\infty(\boldsymbol{e}_j, Z) \le \min_{j\in[d]} \operatorname{dist}_\infty(\boldsymbol{e}_j, Z').$$

As every k-dimensional subspace of $\mathbb{R}^d$ with $k \ge 1$ has a one-dimensional subspace, it suffices to prove the second statement for $k = 1$. i.e., for any one-dimensional subspace Z of $\mathbb{R}^d$,

$$\min_{j\in[d]} \operatorname{dist}_\infty(\boldsymbol{e}_j, Z) \le \frac{1}{2}.$$

Let $Z = \operatorname{Span}(\boldsymbol{z})$, where $\boldsymbol{z} = (z_1, \cdots, z_d) \neq \boldsymbol{0}$. Without loss of generality, let $\|\boldsymbol{z}\|_\infty = 1$ and take $j \in [d]$ such that $|z_j| = 1$. Let $\boldsymbol{z}' = \frac{z_j}{2}\boldsymbol{z} \in Z$. Then,

$$\begin{aligned}
\|\boldsymbol{z}' - \boldsymbol{e}_j\|_\infty &= \left\|\left(\frac{z_j z_1}{2}, \cdots, \frac{z_j z_{j-1}}{2}, \frac{z_j z_j}{2} - 1, \frac{z_j z_{j+1}}{2}, \cdots, \frac{z_j z_d}{2}\right)\right\| \\
&\overset{(a)}{=} \left\|\left(\frac{z_j z_1}{2}, \cdots, \frac{z_j z_{j-1}}{2}, -\frac{1}{2}, \frac{z_j z_{j+1}}{2}, \cdots, \frac{z_j z_d}{2}\right)\right\| \\
&\overset{(b)}{\le} \frac{1}{2},
\end{aligned}$$

where (a) is by $|z_j| = 1$, and (b) is by $\|\boldsymbol{z}\|_\infty = 1$. Therefore,

$$\min_{j' \in [d]} \operatorname{dist}_\infty(\boldsymbol{e}'_j, Z) \leq \operatorname{dist}_\infty(\boldsymbol{e}_j, Z) \leq \left\|\boldsymbol{z}' - \boldsymbol{e}_j\right\| \leq \frac{1}{2},$$

concluding the statement. ■

The following lemma generalizes Theorem 36 to the case where we consider only the distance to a subset of the standard basis, instead of the whole standard basis.

Lemma 37 *For $1 \leq t \leq d$, let $\{\boldsymbol{e}_j\}_{j \in [t]} \subseteq \mathbb{R}^d$ denote the first t vectors from the standard basis in $\mathbb{R}^d$. Then, for any k-dimensional subspace Z of $\mathbb{R}^d$ with $k \geq d - t + 1$,*

$$\min_{j \in [t]} \operatorname{dist}_\infty(\boldsymbol{e}_j, Z) \leq \frac{1}{2}.$$

Proof Similar to Theorem 11, we start by considering the dimension of the intersection between Z and $\mathbb{R}^t$, both as a subspace of $\mathbb{R}^d$. Let $Q = [\boldsymbol{e}_1 \boldsymbol{e}_2 \cdots \boldsymbol{e}_t]^\top \in \mathbb{R}^{t \times d}$. Then,

$$\mathbb{R}^d = \operatorname{Col}(Q^\top) \oplus \operatorname{Null}(Q) = (Z \cap \operatorname{Col}(Q^\top)) \oplus (Z^\perp \cap \operatorname{Col}(Q^\top)) \oplus \operatorname{Null}(Q).$$

By considering the dimension,

$$\begin{aligned}
\dim(Z \cap \operatorname{Col}(Q^\top)) &= \dim \mathbb{R}^d - \dim(Z^\perp \cap \operatorname{Col}(Q^\top)) - \dim \operatorname{Null}(Q) \\
&\geq \dim \mathbb{R}^d - \dim Z^\perp - \dim \operatorname{Null}(Q) \\
&= d - (d - k) - (d - t) \\
&= k - (d - t)
\end{aligned}$$

Under the assumption $k \geq d - t + 1$, we have

$$\dim(Z \cap \operatorname{Col}(Q^\top) = \dim \phi(Z \cap \operatorname{Col}(Q^\top) \geq k - (d - t) \geq 1.$$

Then,

$$\begin{aligned}
\min_{j \in [t]} \operatorname{dist}_\infty(\boldsymbol{e}_j, Z) &\leq \min_{j \in [t]} \operatorname{dist}_\infty(\boldsymbol{e}_j, Z \cap \operatorname{Col}(Q^\top)) \\
&= \min_{j \in [t]} \operatorname{dist}_\infty(\phi(\boldsymbol{e}_j), \phi(Z \cap \operatorname{Col}(Q^\top)) \\
&\overset{(b)}{\leq} \frac{1}{2},
\end{aligned}$$

where (b) is by Theorem 36. ■

F.2. Extension of Sufficiency Condition to ℓ_p-norm

Theorem 38 *Let $p \in [1, \infty]$. For any dataset $\mathcal{D} \in \mathcal{D}_{d,N,C}$ and $\eta \in (0,1)$, the following statements hold:*

(i) *If $\rho \in \left(0, \frac{1}{2N\sqrt{d}\gamma_p(d)}\right]$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(\sqrt{N})$ that ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm.*

(ii) *If $\rho \in \left(\frac{1}{2N\sqrt{d}\gamma_p(d)}, \frac{1}{5\sqrt{d}\gamma_p(d)}\right]$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(Nd^{\frac{1}{4}}\rho^{\frac{1}{2}}\gamma_p(d)^{\frac{1}{2}})$ that ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm with error at most η.*

(iii) *If $\rho \in \left(\frac{1}{5\sqrt{d}\gamma_p(d)}, \frac{1}{\gamma_p(d)}\right)$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(Nd\rho^2\gamma_p(d)^2)$ that ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm.*

To prove Theorem 7, we decompose it into three theorems (Theorems 39 to 41), each corresponding to one of the cases in the statement. They are following.

Lemma 39 *Let $\rho \in \left(0, \frac{1}{2N\sqrt{d}\gamma_p(d)}\right]$ and $p \in [1, \infty]$. For any dataset $\mathcal{D} \in \mathcal{D}_{d,N,C}$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(\sqrt{N})$ that ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm.*

Proof Let $\rho' = \gamma_p(d)\rho$. Then, we have $\rho' \in \left(0, \frac{1}{2N\sqrt{d}}\right]$ from the condition of ρ. By Theorem 7(i), there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(\sqrt{N})$ that ρ'-robustly memorizes $\mathcal{D}$ under ℓ_p-norm. In other words, it holds $f(\boldsymbol{x}') = y_i$, for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$ and $\boldsymbol{x}' \in \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2})$.

We consider two cases depending on whether $p \geq 2$ or $p < 2$, which affect the direction of inclusion between ℓ_p and ℓ_2 balls.

Case I : $p \geq 2$. In this case, we have

$$\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}) \overset{(a)}{\subseteq} \mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},2}) \overset{(b)}{\subseteq} \mathcal{B}_2(\boldsymbol{x}_i, \gamma_p(d)\rho\epsilon_{\mathcal{D},2}) = \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}),$$

where (a) holds by Equation (30) and (b) holds by Theorem 32 applying $p = 2$ and $q = p$.

Thus, for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$ and $\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p})$, it also holds $f(\boldsymbol{x}') = y_i$. In other words, f ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm with $\tilde{O}(\sqrt{N})$ parameters.

Case II : $p < 2$. In this case, we have

$$\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}) \overset{(a)}{\subseteq} \mathcal{B}_p(\boldsymbol{x}_i, \gamma_p(d)\rho\epsilon_{\mathcal{D},2}) \overset{(b)}{\subseteq} \mathcal{B}_2(\boldsymbol{x}_i, \gamma_p(d)\rho\epsilon_{\mathcal{D},2}) = \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}),$$

where (a) holds by Equation (31) and (b) holds by Theorem 32 applying $p = p$ and $q = 2$.

Thus, for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$ and $\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p})$, it also holds $f(\boldsymbol{x}') = y_i$. In other words, f ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm with $\tilde{O}(\sqrt{N})$ parameters.

■

Lemma 40 *Let $\rho \in \left(\frac{1}{2N\sqrt{d}\gamma_p(d)}, \frac{1}{5\sqrt{d}\gamma_p(d)}\right]$ and $p \in [1, \infty]$. For any dataset $\mathcal{D} \in \mathcal{D}_{d,N,C}$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(Nd^{\frac{1}{4}}\rho^{\frac{1}{2}}\gamma_p(d)^{\frac{1}{2}})$ that ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm with error at most η.*

Proof Let $\rho' = \gamma_p(d)\rho$. Then, we have $\rho' \in \left(\frac{1}{2N\sqrt{d}}, \frac{1}{5\sqrt{d}}\right)$ from the condition of ρ.

We consider two cases depending on whether $p \geq 2$ or $p < 2$, which affect the direction of inclusion between ℓ_p and ℓ_2 balls.

Case I : $p \geq 2$**.** In this case, we have:

$$\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}) \overset{(a)}{\subseteq} \mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},2}) \overset{(b)}{\subseteq} \mathcal{B}_2(\boldsymbol{x}_i, \gamma_p(d)\rho\epsilon_{\mathcal{D},2}) = \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}),$$

where (a) holds by Equation (30) and (b) holds by Theorem 32 applying $p = 2$ and $q = p$.

Case II : $p < 2$**.** In this case, we have:

$$\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}) \overset{(a)}{\subseteq} \mathcal{B}_p(\boldsymbol{x}_i, \gamma_p(d)\rho\epsilon_{\mathcal{D},2}) \overset{(b)}{\subseteq} \mathcal{B}_2(\boldsymbol{x}_i, \gamma_p(d)\rho\epsilon_{\mathcal{D},2}) = \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}),$$

where (a) holds by Equation (31) and (b) holds by Theorem 32 applying $p = p$ and $q = 2$.

Thus, in both cases, it holds:

$$\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}) \subseteq \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}). \tag{34}$$

We define $\eta' = \eta\frac{\mathrm{Vol}(\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))}{\mathrm{Vol}(\mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}$. We apply Theorem 7(ii) with the robustness ratio ρ' and the error rate η', then we obtain $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(Nd^{\frac{1}{4}}\rho'^{\frac{1}{2}}) = \tilde{O}(Nd^{\frac{1}{4}}\rho^{\frac{1}{2}}\gamma_p(d)^{\frac{1}{2}})$ that ρ'-robustly memorizes $\mathcal{D}$ with error at most η' under ℓ_p-norm. In other words, for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$, it holds that

$$\mathbb{P}_{\boldsymbol{x}' \sim \mathrm{Unif}(\mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}[f(\boldsymbol{x}') \neq y_i] < \eta'. \tag{35}$$

For simplicity, we denote $E = \{\boldsymbol{x} \in \mathbb{R}^d \mid f(\boldsymbol{x}') \neq y_i\}$. Then, we have

$$\begin{aligned}
&\mathbb{P}_{\boldsymbol{x}' \sim \mathrm{Unif}(\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))}[f(\boldsymbol{x}') \neq y_i] \\
=&\mathbb{P}_{\boldsymbol{x}' \sim \mathrm{Unif}(\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))}[\boldsymbol{x} \in E] \\
=&\frac{\mathrm{Vol}(E \cap \mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))}{\mathrm{Vol}(\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))} \\
\overset{(a)}{\leq}&\frac{\mathrm{Vol}(E \cap \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}{\mathrm{Vol}(\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))} \\
=&\frac{\mathrm{Vol}(E \cap \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}{\mathrm{Vol}(\mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}\frac{\mathrm{Vol}(\mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}{\mathrm{Vol}(\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))} \\
=&\mathbb{P}_{\boldsymbol{x}' \sim \mathrm{Unif}(\mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}[\boldsymbol{x} \in E] \cdot \frac{\mathrm{Vol}(\mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}{\mathrm{Vol}(\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))} \\
=&\mathbb{P}_{\boldsymbol{x}' \sim \mathrm{Unif}(\mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}[f(\boldsymbol{x}') \neq y_i] \cdot \frac{\mathrm{Vol}(\mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}{\mathrm{Vol}(\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))} \\
\overset{(b)}{<}&\eta'\frac{\mathrm{Vol}(\mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}))}{\mathrm{Vol}(\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))} \\
\overset{(c)}{=}&\eta,
\end{aligned}$$

where (a) holds by Equation (34), (b) holds by Equation (35), and (c) holds by the definition of η'.

Thus, for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$, it holds:

$$\mathbb{P}_{\boldsymbol{x}' \sim \text{Unif}(\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}))}[f(\boldsymbol{x}') \neq y_i] < \eta.$$

In other words, f ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm with error at most η and $\tilde{O}(Nd^{\frac{1}{4}}\rho^{\frac{1}{2}}\gamma_p(d)^{\frac{1}{2}})$ parameters.

■

Lemma 41 *Let $\rho \in \left(\frac{1}{5\sqrt{d}\gamma_p(d)}, \frac{1}{\gamma_p(d)}\right)$ and $p \in [1, \infty]$. For any dataset $\mathcal{D} \in \mathcal{D}_{d,N,C}$, there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(Nd\rho^2\gamma_p(d)^2)$ that ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm.*

Proof Let $\rho' = \gamma_p(d)\rho$. Then, we have $\rho' \in \left(\frac{1}{5\sqrt{d}}, 1\right)$ from the condition of ρ. By Theorem 7(iii), there exists $f \in \mathcal{F}_{d,P}$ with $P = \tilde{O}(Nd\rho'^2) = \tilde{O}(Nd\rho^2\gamma_p(d)^2)$ that ρ'-robustly memorizes $\mathcal{D}$ under ℓ_p-norm. In other words, it holds $f(\boldsymbol{x}') = y_i$, for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$ and $\boldsymbol{x}' \in \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2})$.

We consider two cases depending on whether $p \geq 2$ or $p < 2$, which affect the direction of inclusion between ℓ_p and ℓ_2 balls.

Case I : $p \geq 2$**.** In this case, we have:

$$\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}) \overset{(a)}{\subseteq} \mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},2}) \overset{(b)}{\subseteq} \mathcal{B}_2(\boldsymbol{x}_i, \gamma_p(d)\rho\epsilon_{\mathcal{D},2}) = \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}),$$

where (a) holds by Equation (30) and (b) holds by Theorem 32 applying $p = 2$ and $q = p$.

Thus, for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$ and $\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p})$, it also holds $f(\boldsymbol{x}') = y_i$. In other words, f ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm with $\tilde{O}(Nd\rho^2\gamma_p(d)^2)$ parameters.

Case II : $p < 2$**.** In this case, we have:

$$\mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p}) \overset{(a)}{\subseteq} \mathcal{B}_p(\boldsymbol{x}_i, \gamma_p(d)\rho\epsilon_{\mathcal{D},2}) \overset{(b)}{\subseteq} \mathcal{B}_2(\boldsymbol{x}_i, \gamma_p(d)\rho\epsilon_{\mathcal{D},2}) = \mathcal{B}_2(\boldsymbol{x}_i, \rho'\epsilon_{\mathcal{D},2}),$$

where (a) holds by Equation (31) and (b) holds by Theorem 32 applying $p = p$ and $q = 2$.

Thus, for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$ and $\boldsymbol{x}' \in \mathcal{B}_p(\boldsymbol{x}_i, \rho\epsilon_{\mathcal{D},p})$, it also holds $f(\boldsymbol{x}') = y_i$. In other words, f ρ-robustly memorizes $\mathcal{D}$ under ℓ_p-norm with $\tilde{O}(Nd\rho^2\gamma_p(d)^2)$ parameters.

■

Appendix G. Comparision to Existing Bounds

G.1. Parameter Complexity of the Construction by Yu et al. [17]

We now analyze the number of parameters of the network construction proposed by Yu et al. [17], which gives the upper bound not depending on ρ, but applying to all $\rho \in (0, 1)$.

Lemma 42 (Theorem B.6, Yu et al. [17]) *Let $p \in \mathbb{N}$. For any class-separated $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i\in[N]} \in \mathcal{D}_{d,N,C}$, let $R > 1$ by any real value with $\|\boldsymbol{x}_i\|_\infty \leq R$ for all $i \in [N]$. For $\rho \in (0, 1)$, define $\gamma := (1 - \rho)\epsilon_{\mathcal{D},p} > 0$. Then, there exists a network with width $O(d)$, and depth $O(Np(\log(\frac{d}{\gamma^p}) + p \log R + \log p)$ that ρ-robustly memorize $\mathcal{D}$ under ℓ_p-norm.*

We note that in the Yu et al. [17] uses the notation $\lambda^p_{\mathcal{D}}/2$ for $\epsilon_{\mathcal{D},p}$ and the radius $\lambda^p_{\mathcal{D}}/2 - \gamma$ in the original statement corresponds to the value $\mu := \rho\epsilon_{\mathcal{D},p}$ in our notation.

Lemma 43 *For any $\mathcal{D} \in \mathcal{D}_{d,N,C}$ and $\rho \in (0, 1)$, define $\gamma := (1 - \rho)\epsilon_{\mathcal{D},p} > 0$ and $R > 1$ with $\|\boldsymbol{x}_i\|_\infty \leq R$ for all $i \in [N]$. Then, there exists a neural network f such that ρ-robustly memorizes $\mathcal{D}$ using at most $O(Nd^2(\log(\frac{d}{\gamma^2}) + \log R))$ parameters.*

Proof By applying Theorem 42 with $p = 2$, we obtain a neural network f that ρ-robustly memorizes $\mathcal{D}$ with width $O(d)$, and depth $O(N(\log(\frac{d}{\gamma^2} + \log R)))$. We count all parameters as defined in Equation (1), so we can upper bound the number of parameters of f as following:

$$\begin{aligned}\sum_{l=1}^{L}(d_{l-1} + 1) \cdot d_l &= \sum_{l=1}^{L} O(d) \cdot O(d) \\ &= O(N(\log(\frac{d}{\gamma^2} + \log R))) \cdot O(d^2) \\ &= O(Nd^2(\log(\frac{d}{\gamma^2}) + \log R)).\end{aligned}$$

■

G.2. Parameter Complexity of the Construction by Egosi et al. [6]

We observe that although Egosi et al. [6] do not explicitly quantify the total number of parameters in their construction, it implicitly yields a network with $O(Nd^3\rho^6)$ parameters. Specifically, we can establish the following:

> For any $\mathcal{D} \in \mathcal{D}_{d,N,C}$ and $\rho \in (\frac{1}{\sqrt{d}}, 1)$, there exists a neural network f that ρ-robustly memorizes $\mathcal{D}$ using $\tilde{O}(Nd^3\rho^6)$ parameters.

This results follows from the network constructed in Theorem 4.4 of Egosi et al. [6]. The proof of Theorem 4.4 proceeds under the assumption that for $7 \leq k \leq d + 5$, and $\rho \leq \frac{1}{4\sqrt{e}}\sqrt{\frac{k-6}{d}}N^{-\frac{2}{k-6}}$. Given this range, Theorem 4.2 of Egosi et al. [6] is applied to construct a robust memorizer of the projected data from $\mathbb{R}^d$ to $\mathbb{R}^k$. Figure 4 and 5 in their paper illustrate this construction. In this construction, projected point propagates through the network $\Theta(Nk)$ times. The width of the

network scales with k, while the other component that is not propagating the point remain constant in width. Thus, the number of parameters is given by:

$$\sum_{l=1}^{L}(d_{l-1}+1)\cdot d_l = \sum_{l=1}^{\Theta(Nk)} \Theta(k^2) = \Theta(Nk\cdot k^2) = \Theta(Nk^3).$$

To translate this to a bound in terms of ρ, we analyze the relationship between ρ and k. For $k \geq 4\log N + 6$, we verify the following inequality:

$$\frac{1}{4\sqrt{e}}\sqrt{\frac{k-6}{d}}N^{-\frac{2}{k-6}} \geq \frac{1}{4\sqrt{e}}\sqrt{\frac{k-6}{d}}N^{-\frac{1}{2\log N}} \overset{(a)}{=} \frac{1}{4e}\sqrt{\frac{k-6}{d}}$$

where (a) holds by $N = e^{\log N}$. Therefore, for $\rho = \frac{1}{4e}\sqrt{\frac{k-6}{d}}$, the network ρ-robustly memorizes $\mathcal{D}$ with $\Theta(Nk^3)$ parameters. From the relationship between ρ and k, solving for k in terms of ρ yields $k = \Theta(d\rho^2)$. Since the minimum value of k under the assumption is 7, the minimum achievable ρ is $\frac{1}{4e}\frac{1}{\sqrt{d}}$.

Thus, for $\rho > \frac{1}{\sqrt{d}}$, the construction yields a network that ρ-robustly memorizes $\mathcal{D}$ with $\Theta(Nk^3) = \Theta(Nd^3\rho^6)$ parameters, as desired.