PROVABLE MEMORIZATION VIA DEEP NEURAL NETWORKS USING SUB-LINEAR PARAMETERS

Anonymous authors
Paper under double-blind review

ABSTRACT

It is known that $\Theta(N)$ parameters are sufficient for neural networks to memorize arbitrary N input-label pairs. By exploiting depth, we show that $\Theta(N^{2/3})$ parameters suffice to memorize N pairs, under a mild condition on the separation of input points. In particular, deeper networks (even with width 3) are shown to memorize more pairs than shallow networks, which also agrees with the recent line of works on the benefits of depth for function approximation. We also provide empirical results that support our theoretical findings.

1 INTRODUCTION

The modern trend of over-parameterizing neural networks has shifted the focus of deep learning theory from analyzing their expressive power toward understanding the generalization capabilities of neural networks. While the celebrated universal approximation theorems state that over-parameterization enables us to approximate the target function with a smaller error (Cybenko, 1989; Pinkus, 1999), the theoretical gain is too small to satisfactorily explain the observed benefits of over-parameterizing already-big networks. Instead of "how well can models fit," the question of "why models do not overfit" has become the central issue (Zhang et al., 2017).

Ironically, a recent breakthrough on the phenomenon known as the *double descent* (Belkin et al., 2019; Nakkiran et al., 2020) suggests that answering the question of "how well can models fit" is in fact an essential element in fully characterizing their generalization capabilities. In particular, the double descent phenomenon characterizes two different phases according to the capability/incapability of the network size for memorizing training samples. If the network size is insufficient for memorization, the traditional bias-variance trade-off occurs. However, after the network reaches the capacity that memorizes the dataset, i.e., "interpolation threshold," larger networks exhibit better generalization. Under this new paradigm, identifying the *minimum size of networks* for memorizing *finite* input-label pairs becomes a key issue, rather than function approximation that considers *infinite* inputs.

The memory capacity of neural networks is relatively old literature, where researchers have studied the minimum number of parameters for memorizing arbitrary N input-label pairs. Existing results showed that $\Theta(N)$ parameters are sufficient for various activation functions (Baum, 1988; Huang and Babri, 1998; Huang, 2003; Yun et al., 2019; Vershynin, 2020). On the other hand, Sontag (1997) established the negative result that for any network using analytic definable activation functions with $o(N)$ parameters, there exists a set of N input-label pairs that the network cannot memorize. The sub-linear number of parameters also appear in a related topic, namely the VC-dimension of neural networks. It has been proved that there exists a set of N inputs such that a neural network with $o(N)$ parameters can "shatter," i.e., memorize arbitrary labels (Maass, 1997; Bartlett et al., 2019). Comparing the two results on $o(N)$ parameters, Sontag (1997) showed that *not all* sets of N inputs can be memorized for arbitrary labels, whereas Bartlett et al. (2019) showed that *at least one* set of N inputs can be shattered. This suggests that there may be a reasonably large family of N input-label pairs that can be memorized with $o(N)$ parameters, which is our main interest.

1.1 SUMMARY OF RESULTS

In this paper, we identify a mild condition satisfied by many practical datasets, and show that $o(N)$ parameters suffice for memorizing such datasets. In order to bypass the negative result by Sontag (1997), we introduce a condition to the set of inputs, called the Δ-separateness.

Definition 1. *For a set $\mathcal{X} \subset \mathbb{R}^{d_x}$, we say $\mathcal{X}$ is Δ-separated if*

$$\sup_{x,x' \in \mathcal{X}: x \neq x'} \|x - x'\|_2 < \Delta \times \inf_{x,x' \in \mathcal{X}: x \neq x'} \|x - x'\|_2.$$

This condition requires that the ratio of the maximum distance to the minimum distance between distinct points is bounded by Δ. Note that the condition is milder when Δ is bigger. By Definition 1, any given finite set of (distinct) inputs is Δ-separated for some Δ, so one might question why Δ-separateness is different from having distinct inputs in a dataset. The key difference is that even if the number of data points N grows, the ratio of the maximum to the minimum should remain bounded by Δ. Given the discrete nature of computers, there are many practical datasets that satisfy Δ-separateness, as we will see shortly. Also, this condition is more general than the minimum distance assumptions ($\forall i, \|x_i\|_2 = 1, \forall i \neq j, \|x_i - x_j\|_2 \geq \rho > 0$) that are employed in existing theoretical results (Hardt and Ma, 2017; Vershynin, 2020). To see this, note that the minimum distance assumption implies $2/\rho$-separateness. In our theorem statements, we will use the phrase "Δ-separated set of input-label pairs" to denote that the set of inputs is Δ-separated.

In our main theorem sketched below, we prove the sufficiency of $o(N)$ parameters for memorizing any Δ-separated set of N pairs (i.e., any Δ-separated set of N inputs with arbitrary labels) even for large Δ. More concretely, our result is of the following form:

Theorem 1 (Informal)**.** *For any $w \in (\frac{2}{3}, 1]$, there exists a $\Theta(N^{2-2w}/\log N + \log \Delta)$-layer, $\Theta(N^w + \log \Delta)$-parameter fully-connected network with sigmoidal or* ReLU *activation that can memorize any Δ-separated set of N input-label pairs.*

We note that $\log$ has base 2. Theorem 1 states that if the number of layers increases with the number of pairs N, then any Δ-separated set of N pairs can be memorized by a network with $o(N)$ parameters. Here, we can check from Definition 1 that the $\log \Delta$ term does not usually dominate the depth or the number of parameters, especially for modern deep architectures and practical datasets. For example, it is easy to check that any dataset consisting of 3-channel images (values from 0 to 255) of size $h \times w$ satisfies $\log \Delta < (9 + \frac{1}{2}\log(hw))$ (e.g., $\log \Delta < 17$ for the ImageNet dataset), which is often much smaller than the depth of modern deep architectures.

Furthermore, for practical datasets, we can show that networks with parameters fewer than the number of pairs can successfully memorize the dataset. For example, in order to perfectly classify one million images in ImageNet dataset[1] with 1000 classes, our result shows that 0.7 million parameters are sufficient. The improvement is more significant for large datasets. To memorize 15.8 million bounding boxes in Open Images V6[2] with 600 classes, our result shows that only 4.5 million parameters suffice.

Theorem 1 improves the sufficient number of parameters for memorizing N pairs from $\Theta(N)$ down to $\Theta(N^w)$ for any $w \in (\frac{2}{3}, 1)$, for deep networks. Then, it is natural to ask whether the depth increasing with N is *necessary* for memorization with $o(N)$ parameters. Combined with the following existing result on VC-dimension, we show that this is indeed necessary, at least for ReLU networks.

Theorem [Bartlett et al. (2019)]**.** (Informal) *For L-layer* ReLU *networks, $\Theta(N/(L \log N))$ parameters are necessary for memorizing at least a single set of N inputs with arbitrary labels.*

The above theorem states that for ReLU networks of constant depth, $\Theta(N/\log N)$ parameters are necessary for memorizing at least one set of N inputs with arbitrary labels. In contrast, by increasing depth with N, Theorem 1 shows that there is a class of datasets (i.e., Δ-separated) that can be memorized with $\Theta(N^{2/3})$ parameters. Comparing these two results, one can conclude that depth is *crucial* for memorization with sub-linear parameters.

Given that the depth is critical for the memorization power, is the width also critical? We prove that it is not the case, via the following theorem.

Theorem 2 (Informal)**.** *For a fully-connected network of width 3 with a sigmoidal or* ReLU *activation function, $\Theta(N^{2/3} + \log \Delta)$ parameters (i.e., layers) suffice for memorizing any Δ-separated set of N input-label pairs.*

Theorem 2 states that under $2^{O(N^{2/3})}$-separateness of inputs, the network width does not necessarily have to increase with N for memorization with sub-linear parameters. Furthermore, it shows that

[1] http://www.image-net.org/

[2] https://storage.googleapis.com/openimages/web/index.html

even a surprisingly narrow network of width 3 has a superior memorization power than a fixed-depth network, requiring only $\Theta(N^{2/3})$ parameters.

Theorems 1 and 2 show the *existence* of network architectures that memorize N points with $o(N)$ parameters, under the condition of Δ-separateness. This means that these theorems do not answer the question of how many such data points can a *given* network memorize. We provide generic criteria for identifying the maximum number of points given general networks (Theorem 3). In a nutshell, our criteria indicate that to memorize more pairs under the same budget for the number of parameters, a network must have a deep and narrow architecture at the final layers of the network. In contrast to the prior results that the number of arbitrary pairs that can be memorized is at most proportional to the number of parameters (Yamasaki, 1993; Yun et al., 2019; Vershynin, 2020), our criteria successfully incorporate with the characteristics of datasets, the number of parameters, and the architecture, which enable us to memorize Δ-separated datasets with number of pairs super-linear in the number of parameters.

Finally, we provide empirical results corroborating our theoretical findings that deep networks often memorize better than their shallow counterparts with a similar number of parameters. Here, we emphasize that better memorization power does not necessarily imply better generalization. We indeed observe that shallow and wide networks often generalize better than deep and narrow networks, given the same (or similar) training accuracy. This coincides with recent observations that more width leads to better generalization (Zhang et al., 2017; Nakkiran et al., 2020).

Organization. We first introduce related works in Section 2. In Section 3, we introduce necessary notation and the problem setup. We formally state our main results and discuss them in Section 4. In Section 5, we provide empirical observations on the effect of depth and width in neural networks. Finally, we conclude the paper in Section 6.

2 Related works

2.1 Number of parameters for memorization

Sufficient number of parameters for memorization. Identifying the sufficient number of parameters for memorizing arbitrary N pairs has a long history. Earlier works mostly focused on bounding the number of hidden neurons of shallow networks for memorization. Baum (1988) proved that for 2-layer Step[3] networks, $\Theta(N)$ hidden neurons (i.e., $\Theta(N)$ parameters) are sufficient for memorizing arbitrary N pairs when inputs are in general position. Huang and Babri (1998) showed that the same bound holds for any bounded and nonlinear activation function σ satisfying that either $\lim_{x\to-\infty}\sigma(x)$ or $\lim_{x\to\infty}\sigma(x)$ exists, without any condition on inputs. The $\Theta(N)$ bounds on the number of hidden neurons was improved to $\Theta(\sqrt{N})$ by exploiting an additional hidden layer by Huang (2003); nevertheless, this construction still requires $\Theta(N)$ parameters.

With the advent of deep learning, the study has been extended to modern activation functions and deeper architectures. Zhang et al. (2017) proved that $\Theta(N)$ hidden neurons are sufficient for 2-layer ReLU networks to memorize arbitrary N pairs. Yun et al. (2019) showed that for deep ReLU (or hard $\tanh$) networks having at least 3 layers, $\Theta(N)$ parameters are sufficient. Vershynin (2020) proved a similar result for Step (or ReLU) networks with an additional logarithmic factor, i.e., $\widetilde{\Theta}(N)$ parameters are sufficient, for memorizing arbitrary $\{x_i : \|x_i\|_2 = 1\}_{i=1}^N$ satisfying $\|x_i - x_j\|_2^2 = \Omega(\frac{\log\log d_{\max}}{\log d_{\min}})$ and $N, d_{\max} = e^{O(d_{\min}^{1/5})}$ where $d_{\max}$ and $d_{\max}$ denote the maximum and the minimum hidden dimensions, respectively.

In addition, the memorization power of modern network architectures has also been studied. Hardt and Ma (2017) showed that ReLU networks consisting of residual blocks with $\Theta(N)$ hidden neurons can memorize arbitrary $\{x_i : \|x_i\|_2 = 1\}_{i=1}^N$ satisfying $\|x_i - x_j\|_2 \geq \rho$ for some absolute constant $\rho > 0$. Nguyen and Hein (2018) studied a broader class of layers and proved that $\Theta(N)$ hidden neurons suffice for convolutional neural networks consisting of fully-connected, convolutional, and max-pooling layers for memorizing arbitrary N pairs having different patches.

Necessary number of parameters for memorization. On the other hand, the necessary number of parameters for memorization has also been studied. Sontag (1997) showed that for any neural network

[3] Step denotes the binary threshold activation function: $x \mapsto \mathbf{1}[x \geq 0]$.

using analytic definable activation functions, $\Omega(N)$ parameters are necessary for memorizing arbitrary N pairs. Namely, given any network using analytic definable activation with $o(N)$ parameters, there exists a set of N pairs that the network cannot memorize.

The Vapnik-Chervonenkis (VC) dimension is also closely related to the memorization power of neural networks. While the memorization power studies the number of parameters for memorizing *arbitrary* N pairs, the VC-dimension studies the number of parameters for memorizing *at least a single set* of N inputs with arbitrary labels. Hence, it naturally provides the lower bound on the necessary number of parameters for memorizing arbitrary N pairs. The VC-dimension of neural networks has been studied for various types of activation functions. For memorizing at least a single set of N inputs with arbitrary labels, it is known that $\Theta(N/\log N)$ parameters are necessary (Baum and Haussler, 1989) and sufficient (Maass, 1997) for STEP networks. Similarly, Karpinski and Macintyre (1997) proved that $\Omega(\sqrt{N}/U)$ parameters are necessary for sigmoid networks of U neurons. Recently, Bartlett et al. (2019) showed that $\Theta(N/(\bar{L}\log N))$ parameters are necessary and sufficient for L-layer networks using any piecewise linear activation function where $\bar{L} := \frac{1}{W_L}\sum_{\ell=1}^{L} W_\ell$ and W_ℓ denotes the number of parameters up to the ℓ-th layer.

2.2 BENEFITS OF DEPTH IN NEURAL NETWORKS

To understand deep learning, researchers have investigated the advantages of deep neural networks compared to shallow neural networks with a similar number of parameters. Initial results discovered examples of deep neural networks that cannot be approximated by shallow neural networks without using exponentially many parameters (Telgarsky, 2016; Eldan and Shamir, 2016; Arora et al., 2018). Recently, it is discovered that deep neural networks require fewer parameters than shallow neural networks to represent or approximate a class of periodic functions (Chatziafratis et al., 2020a;b). For approximating general continuous functions, Yarotsky (2018) proved that the number of required parameters for RELU networks of constantly bounded depth are square to that for deep RELU networks.

3 NOTATION AND PROBLEM SETUP

In this section, we introduce notation and the problem setup. We use $\log$ to denote the logarithm to the base 2. We let RELU be the function $x \mapsto \max\{x, 0\}$. For $x \in \mathbb{R}$, we denote $\lceil x \rceil := \min\{z \geq x : z \in \mathbb{Z}\}$ and $\lfloor x \rfloor := \max\{z \leq x : z \in \mathbb{Z}\}$.

Throughout this paper, we consider fully-connected feedforward networks. In particular, we consider the following setup: Given an activation function σ, we define a neural network f_θ of L layers (or equivalently $L-1$ hidden layers), input dimension d_x, output dimension 1, and hidden layer dimensions $d_1, \ldots, d_{L-1}$ parameterized by θ as

$$f_\theta := t_{L-1} \circ \sigma \circ \cdots \circ t_1 \circ \sigma \circ t_0.$$

Here, $t_\ell : \mathbb{R}^{d_\ell} \to \mathbb{R}^{d_{\ell+1}}$ is an affine transformation parameterized by θ.[4] We denote a neural network using an activation function σ by a "σ network." We define the width of f_θ as the maximum over $d_1, \ldots, d_{L-1}$.

As we introduced in Section 1.1, our main results hold for any sigmoidal activation function and RELU. Here, we formally define the *sigmoidal* functions as follows.

Definition 2. *We say a function $\sigma : \mathbb{R} \to \mathbb{R}$ is sigmoidal if the following conditions are satisfied.*

- *Both $\lim_{x\to-\infty} \sigma(x)$, $\lim_{x\to\infty} \sigma(x)$ exist and $\lim_{x\to-\infty} \sigma(x) \neq \lim_{x\to\infty} \sigma(x)$.*
- *There exists $z \in \mathbb{R}$ such that σ is continuously differentiable at z and $\sigma'(z) \neq 0$.*

A class of sigmoidal functions covers many activation functions including sigmoid, $\tanh$, hard $\tanh$, etc.[5] Furthermore, since hard $\tanh$ can be represented as a combination of two RELU functions, all results for sigmoidal activation functions hold for RELU as well.[6]

[4] We set $d_0 := d_x$ and $d_L := 1$.

[5] hard $\tanh$ activation function: $x \mapsto -\mathbf{1}[x \leq -1] + x \cdot \mathbf{1}[-1 < x \leq 1] + \mathbf{1}[x > 1]$.

[6] hard $\tanh(x) = \text{RELU}(x+1) - \text{RELU}(x-1) - 1 = \text{RELU}\big(2 - \text{RELU}(1-x)\big) - 1$

Lastly, we formally define the memorization as follows.

Definition 3. *Given $C, d_x \in \mathbb{N}$, a set of inputs $\mathcal{X} \subset \mathbb{R}^{d_x}$, a label function $y : \mathcal{X} \to \{0, \dots, C-1\}$, and a neural network $f_\theta : \mathbb{R}^{d_x} \to \mathbb{R}$ parameterized by θ, we say f_θ can memorize $\{(x, y(x)) : x \in \mathcal{X}\}$ in d_x dimension with C classes if for any $\varepsilon > 0$, there exists θ such that $|f_\theta(x) - y(x)| \le \varepsilon$ for all $x \in \mathcal{X}$.*

Definition 3 define the memorizability as the ability of a network f_θ to fit a set of input-label pairs. While existing results often define memorization only for the binary classification problem (i.e., $C = 2$), we consider arbitrary C classes, and prove our results for general multi-class classification problems. We will often write "f_θ can memorize arbitrary N pairs" without "in d_x dimension with C classes" throughout the paper.

4 Main Results

4.1 Memorization via sub-linear parameters

Efficacy of depth for memorization. Now, we are ready to introduce our main theorem on memorizing N pairs with $o(N)$ parameters. The proof of Theorem 1 is presented in Appendix C.

Theorem 1. *For any $C, N, d_x \in \mathbb{N}$, for any $w \in [2/3, 1]$, for any $\Delta \ge 1$, for any sigmoidal activation function σ, there exists a σ network f_θ of $\Theta\big(\log \Delta + \frac{N^{2-2w}}{1+(1.5w-1)\log N} \log C\big)$ hidden layers and $\Theta\big(d_x + \log \Delta + N^w + N^{1-w/2}(1 + (1.5w-1)\log N)\log C\big)$ parameters such that f_θ can memorize any Δ-separated set of N pairs in d_x dimension with C classes.*

Note that Theorem 1 covers $w = 2/3$ which is not included in its informal version presented in Section 1.1. However, we could only achieve $\Theta(N^{2/3})$ parameters at $w = 2/3$ as the $\log N$ term disappears. In addition, while we only address sigmoidal activation functions in the statement of Theorem 1, note that the same conclusion naturally holds for ReLU as we described in Section 3.

In Theorem 1, Δ only incurs $\Theta(\log \Delta)$ overhead to the number of layers and the number of parameters. As we introduced in Section 1.1, $\log \Delta$ for modern datasets is often very small (e.g., $\log \Delta < 17$ for the ImageNet dataset). In addition, the multiplicative constants for $\log \Delta$ hidden in Θ is also small: Δ incurs at most $\lceil \log \Delta \rceil$ more layers and $12\lceil \log \Delta \rceil$ more parameters (see Appendix C). Hence, the Δ-separateness condition is often negligible for practical datasets.

Suppose that d_x and C are treated as constants, as also assumed in existing results. Then, Theorem 1 implies that if $\log \Delta = O(N^w)$ for some $w < 1$, then $\Theta(N^w)$ (i.e., sub-linear to N) parameters are sufficient for sigmoidal or ReLU networks to memorize arbitrary Δ-separated set of N pairs. Furthermore, if $\log \Delta \le O(N^{2-2w}/\log N)$ and $w \in (\frac{2}{3}, 1)$, then the network construction in Theorem 1 has $\Theta(N^{2-2w}/\log N)$ layers and $\Theta(N^w)$ parameters. Note that the condition $\log \Delta \le O(N^{2-2w}/\log N)$ is very loose for many practical datasets, especially for those with huge N. Combined with the lower bound $\Theta(N/\log N)$ on the necessary number of parameters for ReLU networks of constant depth (Bartlett et al., 2019), Theorem 1 implies that depth growing in N is a necessary condition for memorizing with sub-linear number of parameters. In other words, deeper ReLU networks have more memorization power.

Unimportance of width for memorization. While depth is critical for the memorization power, we show that the width is not very critical. In particular, we prove that extremely narrow networks of width 3 can memorize using with $\Theta(N^{2/3})$ layers (i.e., $\Theta(N^{2/3})$ parameters) as stated in the following theorem. The proof of Theorem 2 is presented in Appendix D.

Theorem 2. *For any $C, N, d_x \in \mathbb{N}$, for any $\Delta \ge 1$, for any sigmoidal activation function σ, a σ network of $\Theta(N^{2/3} \log C)$ hidden layers and width 3 can memorize any Δ-separated set of N pairs in d_x dimension with C classes.*

The statement of Theorem 2 might be somewhat surprising since the network width for memorization does not depend to the input dimension d_x. This is in contrast with the recent universal approximation results that width at least $d_x + 1$ is necessary for approximating functions having d_x-dimensional domain (Lu et al., 2017; Hanin and Sellke, 2017; Johnson, 2019; Park et al., 2020). The main difference follows from the fundamental difference in the two approximation problems, i.e., approximating

a function at finite inputs versus infinite inputs, e.g., the unit cube. The memorization task aims to map finite input-label pairs; here, transforming d_x-dimensional inputs to distinct scalar values can be easily done even using a single layer of a network (e.g., simple projection using inner product). In other words, the dimensionality of inputs is not very important as long as they can be translated to distinct scalar values. In contrast, there is no "natural" way to map a unit cube in a d_x-dimensional space to lower dimension without losing any expressive power. Therefore, function approximation in general cannot be done with width independent of d_x.

Extension to regression problem. The results of Theorem 1 and Theorem 2 can be easily applied to the regression problem, i.e., when labels are from $[0, 1]$. This is because one can simply translate the regression problem with some $\varepsilon > 0$ error tolerance to the classification problem with $\lceil 1/\varepsilon \rceil$ classes. Here, each class $c \in \{0, 1, \ldots, \lceil 1/\varepsilon \rceil\}$ corresponds to the target value $c \cdot \varepsilon$. Hence, the regression problem can also be solved within ε error with $o(N)$ parameters, where the sufficient number of layers and the sufficient number of parameters are identical to the numbers in Theorem 1 and Theorem 2 with the replacement of $\log C$ with $\log(1/\varepsilon)$.

Relation with benefits of depth in neural networks. Our observation that deeper ReLU networks have more memorization power is closely related to the recent studies on the benefits of depth in neural networks (Telgarsky, 2016; Eldan and Shamir, 2016; Lin et al., 2017; Arora et al., 2018; Yarotsky, 2018; Chatziafratis et al., 2020a;b). While our observation indicates that depth is critical for the memorization power, these works mostly focused on showing the importance of depth for approximating functions. Here, the existing results on benefits of depth for function approximation cannot directly imply the benefits of depth for memorization since they often focus on specific classes of functions or require parameters far beyond $\Theta(N)$.

4.2 Generic criteria for identifying memorization power

While Theorem 1 proves the existence of networks having $o(N)$ parameters for memorization, the following theorem states the generic criteria for verifying the memorization power of network architectures. The proof of Theorem 3 is presented in Section E.

Theorem 3. *For some sigmoidal activation function σ, let f_θ be a σ network of L hidden layers having d_ℓ neurons at the ℓ-th hidden layer. Then, for $C, N, d_x \in \mathbb{N}$, for $\Delta \geq 1$, f_θ can memorize any Δ-separated set of N pairs in d_x dimension and C classes if the following statement holds:*

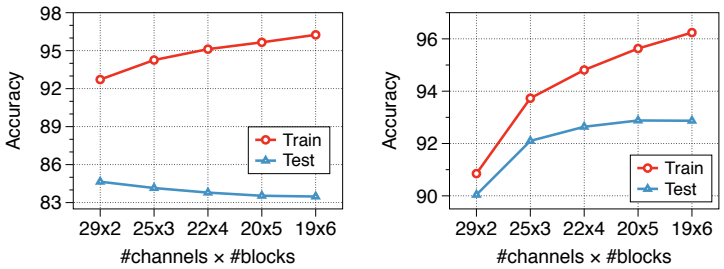
There exist $0 < L_1 < \cdots < L_K < L$ for some $2 \leq K \leq \log N$ satisfying Conditions 1–4 below.

1. $d_\ell \geq 3$ *for all* $\ell \leq L_K$ *and* $d_\ell \geq 7$ *for all* $L_K < \ell$.
2. $\prod_{\ell=1}^{L_1} \lfloor (d_\ell + 1)/2 \rfloor \geq \Delta \sqrt{2\pi d_x}$.
3. $\sum_{L_{i-1}+1}^{L_i} (d_\ell - 2) \geq 2^{i+3}$ *for all* $1 < i \leq K - 1$.
4. $2^K \cdot \left(\sum_{\ell=L_{K-1}+1}^{L_K} (d_\ell - 2) \right) \cdot \left\lfloor (L - L_K - 1)/\lceil \log C \rceil \right\rfloor - 4 \geq N^2$.

Our criteria in Theorem 3 require that the layers of the network can be "partitioned" into have $K + 1$ distinct parts characterized by $L_1, \ldots, L_K$ for some $K \geq 2$. Under this partition, we describe the four conditions in Theorem 3 in detail. The first condition suggests that the network width is not very critical. We note that $d_\ell \geq 7$ for $L_K < \ell$ does not contradict Theorem 2 as we highly optimize the network architecture to fit in width 3 for Theorem 2, while we provide generic criteria here.

The second condition considers the first L_1 hidden layers. In order to satisfy this condition, deep and narrow architectures are better than shallow and wide architectures under a similar budget for parameters, due to the product form $\prod_{\ell=1}^{L_1} \lfloor (d_\ell + 1)/2 \rfloor$. Nevertheless, the architecture of the first L_1 hidden layers is not very critical as only $\log \Delta + \frac{1}{2} \log(2\pi d_x)$ layers are sufficient even with width 3 (e.g., $\log \Delta < 17$ for the ImageNet dataset).

The third condition is closely related to the last condition. Namely, as K increases, the LHS in the last condition increases; however, the third condition states that it requires more hidden neurons. Simply, increasing K by one requires doubling hidden neurons from the $(L_1 + 1)$-th hidden layer to the L_{K-1}-th hidden layer. Nevertheless, this doubling hidden neurons would make the LHS of the last condition double as well.



(a) CIFAR-10 (b) SVHN

Figure 1: Depth-width trade-off under a similar number of parameters.

The last condition is simple. As we explained, 2^K is approximately proportional to the number of hidden neurons from the (L_1+1)-th hidden layer to the L_{K-1}-th hidden layer. The second term in the LHS of the last condition $\sum_{\ell=L_{K-1}+1}^{L_K}(d_\ell - 2)$ is even more simple; it only requires counting hidden neurons. On the other hand, the last term counts the number of layers. This indicates that to satisfy conditions in Theorem 3 using few parameters, the last layers of the network should be deep and narrow. In particular, we note that such a deep and narrow architecture in the last layers is indeed necessary for ReLU networks to memorize with $o(N)$ parameters (Bartlett et al., 2019).

Now, we describe how to show memorization with $o(N)$ parameters using our criteria. For simplicity, consider the network with the minimum width stated in the first condition, i.e., $d_\ell = 3$ for all $\ell \leq L_K$ and $d_\ell = 7$ for all $L_K < \ell$. As we explained, the second conditions can be easily satisfied. For the third condition, consider choosing $K = \log(N^{2/3})$, then $\Theta(N^{2/3})$ hidden neurons (i.e., $\Theta(N^{2/3})$ hidden layers) would be sufficient, i.e., $L_{K-1} - L_1 = \Theta(N^{2/3})$. Likewise, we choose remaining L_i to satisfy $L_K - L_{K-1} = \Theta(N^{2/3})$ and $L - L_K = \Theta(N^{2/3})$. Then, it naturally satisfy the last condition while using only $\Theta(N^{2/3})$ parameters.

5 Experiments

In this section, we study the effect of depth and width for training networks. In particular, we empirically compare deep and narrow networks and shallow and wide networks under a similar number of parameters. For the experiments, we use residual networks (He et al., 2016) of the following structure. First, a convolutional layer and ReLU maps a 3-channel input image to a c-channel feature map. Here the size of the feature map is identical to the size of input images. Then, we apply b residual blocks where each residual block maps $x \mapsto \text{ReLU}(\text{Conv} \circ \text{ReLU} \circ \text{Conv}(x) + x)$ while preserving the number of channels and the size of feature map. Finally, we apply an average pooling layer and a fully-connected layer. We train the model for 5×10^5 iterations with batch size 64 by the stochastic gradient descent. We use the initial learning rate 0.1, weight decay 10^{-4}, and the learning rate decay at the 1.5×10^5-th iteration and the 3.5×10^5-th iteration by a multiplicative factor 0.1. All presented results are averaged over three independent trials.

In the following experiments, we observe the training and test accuracy of networks by varying the number of channels (c) and the number of residual blocks (b).

5.1 Depth-width trade-off in memorization

We verify the memorization power of different network architectures having a similar number of parameters. Figure 1 illustrates training and test accuracy of five different architectures with approximately 50000 parameters for classifying the CIFAR-10 dataset (Krizhevsky and Hinton, 2009) and the SVHN dataset (Netzer et al., 2011). One can observe that as the network architecture becomes deeper and narrower, the training accuracy increases. Namely, deep and narrow networks memorize better than shallow and wide networks under a similar number of parameters. This observation agrees with Theorem 1, which states that increasing depth reduces the required number of parameters for memorizing the same number of pairs.

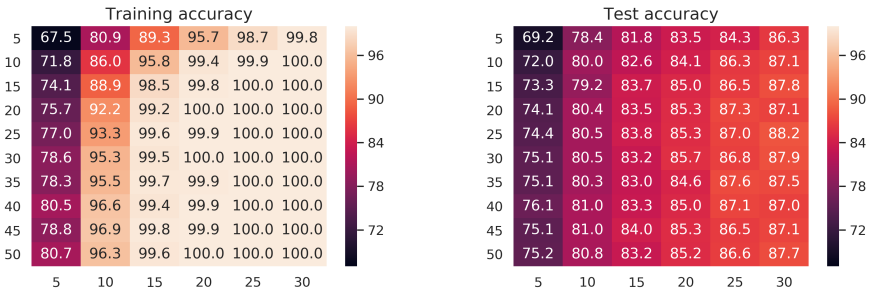


Figure 2: Training and test accuracy by varying width and depth for the CIFAR-10 dataset. The x-axis denotes the number of channels and the y-axis denotes the number of residual blocks.

However, more memorization power does not always imply better generalization. In Figure 1b, as the depth increases, the test accuracy also increases for the SVHN dataset. In contrast, the test accuracy decreases for the CIFAR-10 dataset as the depth increases in Figure 1a. In other words, overfitting occurs for the CIFAR-10 dataset while classifying the SVHN data receive benefits from more expressive power. Note that a similar observation has also been made in the recent double descent phenomenon (Belkin et al., 2019; Nakkiran et al., 2020). It conjectures a transition in the relationship between the expressive power and generalization: from the traditional learning theory that too much expressive power incurs overfitting to the recent observation that more expressive power results in better generalization.

In addition, this observation can provide guidance on the design of network architectures in applications where the training accuracy and small number of parameters are critical. For example, recent development in video streaming services tries to reduce the traffic by compressing a video by neural networks and send the compressed video and the decoder network (Yeo et al., 2018). Here, the corresponding decoder is often trained only for the video to send; hence, only the training accuracy/loss matters while the decoder's number of parameters should also be small for the traffic.

5.2 Effect of width and depth

In this section, we observe the effect of depth and width by varying both. Figure 2 reports the training and test accuracy for the CIFAR-10 dataset by varying the number for channels from 5 to 30 and the number of residual blocks from 5 to 50. We present the same experimental results for the SVHN dataset in Appendix A. First, we observe that a network of 15 channels with feature map size 32×32 successfully memorize (i.e., over 99% training accuracy). This size is much narrower than modern network architectures, e.g., ResNet-18 has 64 channels at the first hidden layer (He et al., 2016). On the other hand, too narrow network (e.g., 5-channels) fail to memorize. This result does not contradict with Theorem 2 as the test of memorization in experiments/practice involves the stochastic gradient descent. We note that similar phenomenons are observed for the SVHN dataset.

Furthermore, once the network memorize, we observe that increasing width is more effective than increasing depth for improving generalization. These results indicate that width is not very critical for the memorization power, while it is effective for generalization. Note that similar observations have been made in Zhang et al. (2017).

6 Conclusion

In this paper, we prove that $\Theta(N^{2/3})$ parameters are sufficient for memorizing arbitrary N input-label pairs under the mild Δ-separateness condition. Our result provides significantly improved results, compared to the prior results showing the sufficiency of $\Theta(N)$ parameters with/without conditions on pairs. In addition, Theorem 1 shows that deeper networks have more memorization power. This result coincides with the recent study on the benefits of depth for function approximation. On the other hand, Theorem 2 shows that network width is not important for the memorization power. We also provide generic criteria for identifying the memorization power of networks. Finally, we empirically confirm our theoretical results.

REFERENCES

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018.

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*, 2019.

Eric B. Baum. On the Capabilities of Multilayer Perceptrons. *Journal of Complexity*, 4(3):193–215, 1988. ISSN 0885-064X.

Eric B. Baum and David Haussler. What size net gives valid generalization? In *Advances in Neural Information Processing Systems*, 1989.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Vaggos Chatziafratis, Sai Ganesh Nagarajan, and Ioannis Panageas. Better Depth-Width Trade-offs for Neural Networks through the lens of Dynamical Systems. In *International Conference on Machine Learning*, 2020a.

Vaggos Chatziafratis, Sai Ganesh Nagarajan, Ioannis Panageas, and Xiao Wang. Depth-width trade-offs for relu networks via sharkovsky's theorem. In *International Conference on Learning Representations*, 2020b.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, 2016.

Walter Gautschi. Some elementary inequalities relating to the gamma and incomplete gamma function. *Journal of Mathematics and Physics*, 38(1-4):77–81, 1959.

Boris Hanin and Mark Sellke. Approximating continuous functions by ReLU nets of minimal width. *arXiv preprint arXiv:1710.11278*, 2017.

Moritz Hardt and Tengyu Ma. Identity matters in deep learning. In *International Conference on Learning Representations*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Guang-Bin Huang. Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*, 14(2):274–281, 2003.

Guang-Bin Huang and Haroon A Babri. Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE Transactions on Neural Networks*, 9(1):224–229, 1998.

Jesse Johnson. Deep, skinny neural networks are not universal approximators. In *International Conference on Learning Representations*, 2019.

Marek Karpinski and Angus Macintyre. Polynomial bounds for vc dimension of sigmoidal and general pfaffian neural networks. *Journal of Computer and System Sciences*, 54(1):169–176, 1997.

Patrick Kidger and Terry Lyons. Universal approximation with deep narrow networks. In *Conference on Learning Theory*, 2020.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems*, 2017.

Wolfgang Maass. Bounds for the computational power and learning complexity of analog neural nets. *SIAM Journal on Computing*, 26(3):708–732, 1997.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep Double Descent: Where Bigger Models and More Data Hurt. In *International Conference on Learning Representations*, 2020.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Deep Learning and Unsupervised Feature Learning Workshop*, 2011.

Quynh Nguyen and Matthias Hein. Optimization Landscape and Expressivity of Deep CNNs. In *International Conference on Machine Learning*, 2018.

Sejun Park, Chulhee Yun, Jaeho Lee, and Jinwoo Shin. Minimum Width for Universal Approximation. *arXiv preprint arXiv:2006.08859*, 2020.

Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8: 143–195, 1999.

Eduardo D. Sontag. Shattering all sets of k points in "general position" requires (k—1)/2 parameters. *Neural Computation*, 9(2):337–348, 1997.

Matus Telgarsky. Benefits of depth in neural networks. In *Conference on Learning Theory*, 2016.

Roman Vershynin. Memory capacity of neural networks with threshold and ReLU activations. *arXiv preprint 2001.06938*, 2020.

Masami Yamasaki. The lower bound of the capacity for a neural network with multiple hidden layers. In *International Conference on Artificial Neural Networks*, 1993.

Dmitry Yarotsky. Optimal approximation of continuous functions by very deep relu networks. In *Conference on Learning Theory*, 2018.

Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. Neural Adaptive Content-aware Internet Video Delivery. In *USENIX Symposium on Operating Systems Design and Implementation*, 2018.

Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small ReLU networks are powerful memorizers: a tight analysis of memorization capacity. In *Advances in Neural Information Processing Systems*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

A TRAINING AND TEST ACCURACY FOR SVHN DATASET

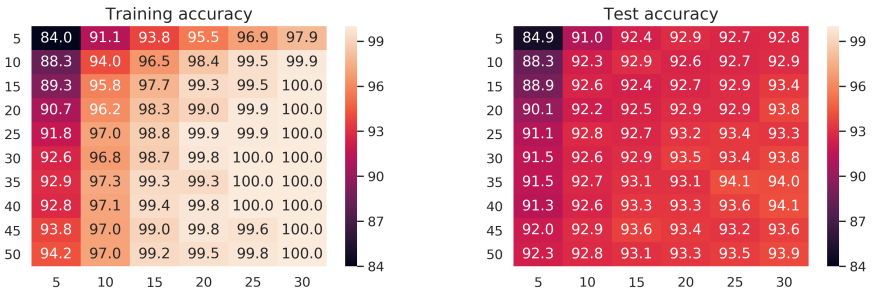


Figure 3: Training and test accuracy by varying width and depth for the SVHN dataset. The x-axis denotes the number of channels and the y-axis denotes the number of residual blocks.

B MAIN PROOF IDEA: MEMORIZATION USING STEP AND ID

B.1 ADDITION NOTATION

We introduce additional notation and notion frequently used in Appendix. For $n \in \mathbb{N}$ and a set $\mathcal{S}$, we denote $\binom{\mathcal{S}}{n} := \{\mathcal{X} \subset \mathcal{S} : |\mathcal{X}| = n\}$ and $[n] := \{0, 1, \dots, n-1\}$. We denote $\text{ID} : x \mapsto x$ and $\text{STEP} : x \mapsto \mathbf{1}[x \geq 0]$. For a set $\mathcal{X} \subset \mathbb{R}$ and a vector $v = (v_i)_{i=1}^d \in \mathbb{R}^d$, we denote $\lfloor \mathcal{X} \rfloor := \{\lfloor x \rfloor : x \in \mathcal{X}\}$ and $\lfloor v \rfloor := \left[\lfloor v_i \rfloor\right]_{i=1}^d$. For $x \geq 0$ and $y > 0$, we denote $x \mod y := x - y \cdot \lfloor \frac{x}{y} \rfloor$. For the notational simplicity, we often use $D := \lceil \log C \rceil$.

B.2 MAIN IDEA

In the proofs of Theorem 1, Theorem 2, and Theorem 3, we use each sigmoidal activation for approximating either the identity function or the binary step function. Thus, we construct our network for the proofs using only ID and STEP activation functions, which directly provide the network construction using sigmoidal activation functions (see Appendix B.3 and Appendix B.4 for details).

B.3 TOOLS

We present the following claims important for the proofs. In particular, Claim 4 and Claim 5 guide how to approximate STEP and ID by a single sigmoidal neuron.

Claim 4 [Kidger and Lyons (2020, Lemma 4.1)]**.** *For any sigmoidal activation function σ, for any bounded interval $\mathcal{I} \subset \mathbb{R}$ for any $\varepsilon > 0$, there exist $a, b, c, d \in \mathbb{R}$ such that $|a \cdot \sigma(c \cdot x + d) + b - x| < \varepsilon$ for all $x \in \mathcal{I}$.*

Claim 5. *For any sigmoidal activation function σ, for any $\varepsilon, \delta > 0$, there exist $a, b, c \in \mathbb{R}$ such that $|a \cdot \sigma(c \cdot x) + b - \mathbf{1}[x \geq 0]| < \varepsilon$ for all $x \notin [-\delta, \delta]$.*

Proof of Claim 5. We assume that $\alpha := \lim_{x \to -\infty} \sigma(x) < \lim_{x \to \infty} \sigma(x) =: \beta$ where the case that $\beta < \alpha$ can be proved using a similar argument. From the definition of α, β, there exists $k > 0$ such that $|\sigma(x) - \alpha| < (\beta - \alpha)\varepsilon$ if $x < -k$ and $|\sigma(x) - \beta| < (\beta - \alpha)\varepsilon$ if $x > k$. Then, choosing $a = \frac{1}{\beta - \alpha}$, $b = -\frac{\alpha}{\beta - \alpha}$, and $c = \frac{k}{\delta}$ completes the proof of Claim 5. □

Claim 6. *For any $a, x \in \mathbb{R}$ such that $a \neq 0$, for any $b \in \mathbb{N}$, it holds that $\lceil \frac{x}{a \cdot b} \rceil = \lceil \frac{\lceil x/a \rceil}{b} \rceil$.*

Proof of Claim 6. It trivially holds that $\lceil \frac{x}{a \cdot b} \rceil \leq \lceil \frac{\lceil x/a \rceil}{b} \rceil$. Now, we show a contradiction if $\lceil \frac{x}{a \cdot b} \rceil < \lceil \frac{\lceil x/a \rceil}{b} \rceil$. Suppose that $\lceil \frac{x}{a \cdot b} \rceil < \lceil \frac{\lceil x/a \rceil}{b} \rceil$. Since $\lceil \frac{x}{a \cdot b} \rceil < \lceil \frac{\lceil x/a \rceil}{b} \rceil$, there exists an integer m such that

$$\frac{x}{a \cdot b} \leq m < \frac{\lceil x/a \rceil}{b} \qquad \text{and hence,} \qquad \frac{x}{a} \leq \left\lceil \frac{x}{a} \right\rceil \leq b \cdot m < \left\lceil \frac{x}{a} \right\rceil$$

which leads to a contradiction. This completes the proof of Claim 6. □

B.4 TRANSFORMING STEP+ID NETWORK TO SIGMOIDAL NETWORK

In this section, we describe how to transform a STEP + ID network into a sigmoidal network within arbitrary error. Formally, we prove the following lemma.

Lemma 7. *For any finite set of inputs $\mathcal{X}$, for any* STEP + ID *network f, for any $\varepsilon > 0$, for any sigmoidal activation function σ, there exists a σ network g having the same architecture with f_θ such that*

$$|f(x) - g(x)| < \varepsilon$$

for all $x \in \mathcal{X}$.

Then, the construction in Lemma 7 enables us to construct STEP + ID networks instead of networks using a sigmoidal activation function for proving Theorem 1, Theorem 2, and Theorem 3.

Proof of Lemma 7. Without loss of generality, we first assume that for any STEP + ID network h, in the evaluation of $h(x)$, all inputs to STEP neurons (i.e., $\mathbf{1}[x \geq 0]$) is non-zero for all $x \in \mathcal{X}$. This assumption can be easily satisfied by adding some small bias to the inputs of STEP neurons where such a bias alwasy exists since $|\mathcal{X}| < \infty$. Furthermore, introducing this assumption does not change $h(x)$ for all $x \in \mathcal{X}$

Now, we describe our construction of g. Let $\delta > 0$ be a number satisfying that the absolute value of all inputs to STEP neurons in the evaluation of $f(x)$ for all $x \in \mathcal{X}$ is at least δ. Such δ always exists due to the assumption we made. Let L be the number of hidden layers in f. Starting from f, we iteratively substitute the STEP and ID hidden neurons into σ from the last hidden layer to the first hidden layer. In particular, by using Claim 4 and Claim 5, we replace ID and STEP neurons in the ℓ-th hidden layer by σ neurons approximating ID and STEP.

First, let g_L be a network identical to f except for its L-th hidden layer consisting of σ neurons approximating ID and STEP of f. Here, we accurately approximate ID and STEP neurons by σ using Claim 4 and Claim 5 so that $|g_L(x) - f(x)| < \varepsilon/L$ for all $x \in \mathcal{X}$. Note that such approximation always exists due to the existence of $\delta > 0$. Now, let g_{L-1} be a network identical to f_L except for its $(L-1)$-th hidden layer consisting of σ neurons approximating ID and STEP of g_L. Here, we also accurately approximate ID and STEP neurons by σ using Claim 4 and Claim 5 so that $|g_L(x) - g_{L-1}(x)| < \varepsilon/L$ for all $x \in \mathcal{X}$. If we repeat this procedure until replacing the first hidden layer, then $g := g_1$ would be the desired network satisfying that $|f(x) - g(x)| < \varepsilon$ for all $x \in \mathcal{X}$. This completes the proof of Lemma 7. $\square$

C Proof of Theorem 1

C.1 Proof sketch

From Lemma 7, constructing a Step + Id network suffices for the proof. Our construction consists of four steps: projection, compression 1, compression 2, and learning. First, in the projection step, we map a Δ-separated set of N input vectors $\mathcal{X}_1 \in \binom{\mathbb{R}^{d_x}}{N}$ to $\mathcal{X}_2 := \{u^\top x + b : x \in \mathcal{X}_1\} \in \binom{\mathbb{R}}{N}$ using some $u \in \mathbb{R}^{d_x}$ and $b \in \mathbb{R}$. Here, we carefully choose u, b so that $\lfloor \mathcal{X}_2 \rfloor \in \binom{[O(N^2\Delta)]}{N}$. In the compression 1 step, we map $\mathcal{X}_2$ to $\mathcal{X}_3$ such that $\lfloor \mathcal{X}_3 \rfloor \in \binom{[U]}{N}$ where $U := \lfloor \frac{N^2}{4} + 1 \rfloor$. In the compression 2 step, we further decrease the upper bound on hidden activation values by mapping $\mathcal{X}_3$ to $\mathcal{X}_4$ such that $\lfloor \mathcal{X}_4 \rfloor \in \binom{[V]}{N}$ for some $N \le V \le U$. Finally, in the learning step, we map $\mathcal{X}_4$ to their corresponding labels (labels given to the original input set $\mathcal{X}_1$), based on the construction for proving the lower bound of the VC dimension of ReLU networks (Bartlett et al., 2019).

For the implementation, the projection step can be implemented by a Step + Id network of no hidden layer and $d_x + 1$ parameters. The compression 1 step can be implemented by a Step + Id network of $\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil$ hidden layers and $12\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil - 3$ parameters. The compression 2 step can be implemented by a Step + Id network of $2\lceil \log \frac{U}{V} \rceil - 1$ hidden layers and $4\lceil \log \frac{U}{V} \rceil + 3 \times \sum_{i=1}^{\lceil \log(U/V) \rceil - 1} \lceil \frac{2N^2}{\lceil U/2^i \rceil} \rceil + 3\lceil \frac{2N}{\lfloor V/N \rfloor} \rceil$ parameters. Finally, for any $A, B, R \in \mathbb{N}$ such that $AB \ge V$, the learning step can be implemented by a Step + Id network of $2\lceil \frac{BD}{R} \rceil + 2$ hidden layers and $4A + \big((2R+5)2^R + 2R^2 + 8R + 7\big)\lceil \frac{BD}{R} \rceil - R2^R - R^2 + 3$ parameters where $D := \lceil \log C \rceil$.

Hence, for any $A, B, R, V \in \mathbb{N}$ such that $N \le V \le U := \lfloor \frac{N^2}{4} + 1 \rfloor$ and $AB \ge V$, the overall Step + Id network (and therefore, the overall σ network) consists of $\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil + 2\lceil \log \frac{U}{V} \rceil + 2\lceil \frac{BD}{R} \rceil + 4$ hidden layers (3 extra hidden layers are for concatenating consecutive steps) and $d_x + 12\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil + 4\lceil \log \frac{U}{V} \rceil + 3 \times \sum_{i=1}^{\lceil \log(U/V) \rceil - 1} \lceil \frac{2N^2}{\lceil U/2^i \rceil} \rceil + 3\lceil \frac{2N}{\lfloor V/N \rfloor} \rceil + 4A + \big((2R+5)2^R + 2R^2 + 8R + 7\big)\lceil \frac{BD}{R} \rceil - R2^R - R^2 + 1$ parameters. We present the detailed implementation of each step including precise required number of layers and number of parameters in Sections C.2-C.5.

Finally, substituting $V \leftarrow \lceil N^{2-w} \rceil$, $A \leftarrow \lceil N^w \rceil$, $B \leftarrow \lceil N^{2-2w} \rceil$, and $R \leftarrow \lceil 1 + (1.5w - 1)\log N \rceil$ and using Lemma 7 result in the statement of Theorem 1.

C.2 Projection: from vectors to scalar values

In the projection step, for a set of input vectors $\mathcal{X}_1 \in \binom{\mathbb{R}^{d_x}}{N}$ such that $\frac{\max_{\{x,x'\} \in \binom{\mathcal{X}}{2}} \|x - x'\|_2}{\min_{\{x,x'\} \in \binom{\mathcal{X}}{2}} \|x - x'\|_2} < \Delta$ to $\mathcal{X}_2$ such that $\lfloor \mathcal{X}_2 \rfloor \in \binom{[\lceil N^2 \Delta \sqrt{\pi d_x / 8} \rceil]}{N}$. Since the projection step is trivial when $d_x = 1$, we consider $d_x \ge 2$ in this section. To this end, we first project all $x \in \mathcal{X}_1$ to $u^\top x \in \mathbb{R}$ by choosing some unit vector $u \in \mathbb{R}^{d_x}$ so that

$$\frac{\max_{\{x,x'\} \in \binom{\mathcal{X}}{2}} |u^\top (x - x')|}{\min_{\{x,x'\} \in \binom{\mathcal{X}}{2}} |u^\top (x - x')|} < N^2 \Delta \sqrt{\frac{\pi d_x}{8}}. \tag{1}$$

Such a unit vector u always exists due to the following lemma. The proof of Lemma 8 is presented in Section F.1.

Lemma 8. *For any $N, d_x \in \mathbb{N}$, for any $\mathcal{X} \in \binom{\mathbb{R}^{d_x}}{N}$, there exists a unit vector $u \in \mathbb{R}^{d_x}$ such that $\sqrt{\frac{8}{\pi d_x}} \frac{1}{N^2} \|x - x'\|_2 \le |u^\top (x - x')| \le \|x - x'\|_2$ for all $x, x' \in \mathcal{X}$.*

Finally, we implement the projection step as: $\times(u^\top x - \min\{u^\top x : x \in \mathcal{X}_1\})$ for some $a \in \mathbb{R}$ so that the output $\mathcal{X}_2$ satisfies $\lfloor \mathcal{X}_2 \rfloor \in \binom{[\lceil N^2 \Delta \sqrt{\pi d_x / 8} \rceil]}{N}$. Note that such $a \in \mathbb{R}$ exists as $\{u^\top x : x \in \mathcal{X}_1\}$ is $(N^2 \Delta \sqrt{\pi d_x / 8})$-separated. Hence, the projection step can be implemented by a Step + Id network of 1 hidden layer and $d_x + 1$ parameters (d_x weights and 1 bias). This completes the construction of the projection step.

C.3 COMPRESSION 1: IMPROVING UPPER BOUND ON HIDDEN FEATURES

In the compression 1 step, we map $\mathcal{X}_2$ to $\mathcal{X}_3$ such that $\lfloor \mathcal{X}_3 \rfloor \in \binom{[U]}{N}$ where $U := \lfloor N^2/4 + 1 \rfloor$. To this end, we introduce the following lemma. The proof of Lemma 9 is presented in Section F.2.

Lemma 9. *For any $N, K, d \in \mathbb{N}$ such that $K > \frac{N^2}{4}$, for any $\mathcal{X}$ such that $\lfloor \mathcal{X} \rfloor \in \binom{[K]}{N}$, there exists a* STEP + ID *network f of 1 hidden layer and width d such that $\lfloor f(\mathcal{X}) \rfloor \in \binom{[T]}{N}$ where $T := \max\left\{\left\lceil \frac{K}{\lfloor (d+1)/2 \rfloor} \right\rceil, \left\lfloor \frac{N^2}{4} + 1 \right\rfloor\right\}$.*

From Lemma 9, one can observe that a STEP + ID network of 1 hidden layer consisting of 3 hidden neuron which can map any $\mathcal{Z}$ such that $\lfloor \mathcal{Z} \rfloor \in \binom{[K]}{N}$ to $\mathcal{Z}'$ such that $\lfloor \mathcal{Z}' \rfloor \in \binom{[\max\{\lceil K/2 \rceil, U]}{N}$. Hence, a STEP + ID network of $\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil$ hidden layers having 3 neurons at each hidden layers and 1 neuron at even hidden layers can map $\mathcal{X}_2$ to $\mathcal{X}_3$ such that $\lfloor \mathcal{X}_3 \rfloor \in \binom{[U]}{N}$ since the following inequality holds:

$$\frac{N^2 \Delta \sqrt{\pi d_x/8}}{2^{\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil}} \leq \frac{N^2 \Delta \sqrt{\pi d_x/8}}{\Delta \sqrt{2\pi d_x}} = \frac{N^2}{4} \leq \left\lfloor \frac{N^2}{4} + 1 \right\rfloor =: U$$

where we use Claim 6 for sequential application of Lemma 10 and we drop $\lceil \cdot \rceil$ from $\left\lceil \frac{N^2 \Delta \sqrt{\pi d_x/8}}{2^{\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil}} \right\rceil$ as $\lceil a \rceil \leq b$ if an only if $a \leq b$ for any $b \in \mathbb{Z}$. Note that this construction requires $12\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil - 2$ parameters ($9\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil - 3$ edges and $3\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil + 1$ biases). This completes the construction of the compression 1 step.

C.4 COMPRESSION 2: FURTHER IMPROVING UPPER BOUND ON HIDDEN FEATURES

In the compression 2 step, we further improve the bound $U := \lfloor \frac{N^2}{4} + 1 \rfloor$ on the hidden features to $V \in \mathbb{N}$ satisfying $N \leq V \leq U$. Namely, we map $\mathcal{X}_3$ to $\mathcal{X}_4$ such that $\lfloor \mathcal{X}_4 \rfloor \in \binom{[V]}{N}$. To construct such a mapping, we introduce the following lemma. The proof of Lemma 10 is presented in Section F.3.

Lemma 10. *For any $N, K, L, d_1, \ldots, d_L \in \mathbb{N}$ such that $N < K$ and $d_\ell \geq 3$ for all ℓ, for any $\mathcal{X}$ such that $\lfloor \mathcal{X} \rfloor \in \binom{[K]}{N}$, there exists a* STEP + ID *network f of L hidden layers having d_ℓ neurons at the ℓ-th hidden layer such that $\lfloor f(\mathcal{X}) \rfloor \in \binom{[T]}{N}$ where $T := \min\left\{K, \max\left\{N \times \lceil \frac{N}{C} \rceil, \lceil \frac{K}{2} \rceil\right\}\right\}$ where $C := \lfloor \frac{1}{2} + \frac{1}{2}\sum_{\ell=1}^{L}(d_\ell - 2) \rfloor$.*

From Lemma 10, one can observe that for any $K \in \mathbb{N}$, there exists a STEP + ID network of one hidden layer consisting of $\left\lceil \frac{2N^2}{\lceil K/2 \rceil} \right\rceil + 1$ hidden neurons which can map any $\mathcal{Z}$ such that $\lfloor \mathcal{Z} \rfloor \in \binom{[K]}{N}$ to $\mathcal{Z}'$ such that $\lfloor \mathcal{Z}' \rfloor \in \binom{\lceil K/2 \rceil}{N}$. Hence, a STEP + ID network of $2\lceil \log \frac{U}{V} \rceil - 1$ hidden layers and d_ℓ hidden neurons at the ℓ-th layer where

$$d_\ell := \begin{cases} \left\lceil \frac{2N^2}{\lceil U/2^{2\ell-1} \rceil} \right\rceil + 1 & \text{if } \ell \text{ is odd and } \ell \neq 2\lceil \log \frac{U}{V} \rceil - 1 \\ \left\lceil \frac{2N}{\lfloor V/N \rfloor} \right\rceil + 1 & \text{if } \ell = 2\lceil \log \frac{U}{V} \rceil - 1 \\ 1 & \text{if } \ell \text{ is even} \end{cases}.$$

can map $\mathcal{X}_3$ to $\mathcal{X}_4$ such that $\lfloor \mathcal{X}_4 \rfloor \in \binom{[V]}{N}$. Here, we use Claim 6 for sequential application of Lemma 10. Note that this construction requires $4\lceil \log \frac{U}{V} \rceil + 3 \times \sum_{i=1}^{\lceil \log(U/V) \rceil - 1} \left\lceil \frac{2N^2}{\lceil U/2^i \rceil} \right\rceil + 3 \times \left\lceil \frac{2N}{\lfloor V/N \rfloor} \right\rceil$ parameters ($2\lceil \log \frac{U}{V} \rceil + 2 \times \sum_{i=1}^{\lceil \log(U/V) \rceil} \left\lceil \frac{2N^2}{\lceil U/2^i \rceil} \right\rceil + 2 \times \left\lceil \frac{2N}{\lfloor V/N \rfloor} \right\rceil$ edges and $2\lceil \log \frac{U}{V} \rceil + \sum_{i=1}^{\lceil \log(U/V) \rceil} \left\lceil \frac{2N^2}{\lceil U/2^i \rceil} \right\rceil + \left\lceil \frac{2N}{\lfloor V/N \rfloor} \right\rceil$ biases). This completes the construction of the compression 2 step.

C.5 LEARNING: FROM BOUNDED VALUES TO LABELS

In the learning step, we map $\mathcal{X}_4$ to its binary labels. To this end, we introduce the following lemma. We note that Lemma 11 follows the construction for proving VC-dimension lower bound of RELU networks by Bartlett et al. (2019). The proof of Lemma 11 is presented in Section F.4.

Lemma 11. *For any $A, B, D, K, R \in \mathbb{N}$ such that $AB \geq K$, there exists a* STEP + ID *network f_θ of $2\lceil \frac{BD}{R} \rceil + 2$ hidden layers and $4A + \big((2R+5)2^R + 2R^2 + 8R + 7\big)\lceil \frac{BD}{R} \rceil - R2^R - R^2 + 3$ parameters satisfying the following property: For any finite set $\mathcal{X} \subset [0, K)$, for any $y : [K] \to [2^D]$, there exists θ such that $f_\theta(x) = y(\lfloor x \rfloor)$ for all $x \in \mathcal{X}$.*

Given some $A, B, R \in \mathbb{N}$ satisfying $AB \geq V$ Lemma 11 directly provides a STEP + ID network of $2\lceil \frac{BD}{R} \rceil + 2$ hidden layers and $4A + \big((2R+5)2^R + 2R^2 + 8R + 7\big)\lceil \frac{BD}{R} \rceil - R2^R - R^2 + 3$ parameters which can map $\mathcal{X}_4$ to their labels. This completes the construction of the learning step.

D PROOF OF THEOREM 2

In this proof, we construct a STEP + ID network and transform it to a σ network using Lemma 7. The network construction for Theorem 2 follows that of Theorem 1 in Appendix C. In particular, Theorem 2 is a direct consequence of Lemma 8, Lemma 9, Lemma 10, and Lemma 12 presented as follows. The proof Lemma 12 is presented in Section F.7.

Lemma 12. *For any $A, B, D, K \in \mathbb{N}$ such that $AB \geq K$, there exists a* STEP + ID *network f_θ of $A + (2D+1)B$ hidden layers and width 3 satisfying the following property: For any finite set $\mathcal{X} \in [0, K)$, for any $y : [K] \to [2^D]$, there exists θ such that $f_\theta(x) = y(\lfloor x \rfloor)$.*

From Lemma 8, Lemma 9, and Claim 6, a STEP + ID network of $\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil$ hidden layers and width 3 can map a Δ-separated set of inputs $\mathcal{X}_1$ to $\mathcal{X}_2$ such that $\lfloor \mathcal{X}_2 \rfloor \in \binom{[U]}{N}$ where $U := \lfloor \frac{N^2}{4} + 1 \rfloor$. From Lemma 10, a STEP + ID network of $\sum_{i=1}^{\lceil \log(U/V) \rceil - 1} \left(\left\lceil \frac{2N^2}{\lceil U/2^i \rceil} \right\rceil - 1 \right) + \left\lceil \frac{2N}{\lfloor V/N \rfloor} \right\rceil - 1$ hidden layers and width 3 can map $\mathcal{X}_2$ to $\mathcal{X}_3$ such that $\lfloor \mathcal{X}_3 \rfloor \in \binom{[V]}{N}$ for some $N \leq V \leq U$. Finally, from Lemma 12, for any $A, B \in \mathbb{N}$ such that $AB \geq V$, a STEP + ID network of $A + (2D+1)B$ hidden layers and width 3 can map $\mathcal{X}_3$ to their labels where $D := \lceil \log C \rceil$.

Hence, for any $A, B, V \in \mathbb{N}$ such that $N \leq V \leq U$ and $AB \geq V$, a STEP + ID network of $\lceil \log(\Delta\sqrt{2\pi d_x}) \rceil + \sum_{i=1}^{\lceil \log(U/V) \rceil - 1} \left(\left\lceil \frac{2N^2}{\lceil U/2^i \rceil} \right\rceil - 1 \right) + \left\lceil \frac{2N}{\lfloor V/N \rfloor} \right\rceil + A + (2D+1)B - 1$ hidden layers and width 3 can memorize arbitrary Δ-separated set of size N. Note that combining functions does not require additional hidden layer as the linear maps constructing the outputs of functions can be absorbed into the first linear map in the next function.

Finally, substituting $V \leftarrow \lceil N^{4/3} \rceil$, $A \leftarrow \lceil \sqrt{(2D+1)V} \rceil$, and $B \leftarrow \lceil \sqrt{V/(2D+1)} \rceil$ and using Lemma 7 result in the statement of Theorem 2. This completes the proof of Theorem 2.

E PROOF OF THEOREM 3

The proof of Theorem 3 have the same structure of the proof of Theorem 1 consisting of four steps: projection, compression 1, compression 2, and learning. Note that we construct a STEP + ID network and use Lemma 7 to complete the proof.

For the network construction, we divide the function of f_θ into four disjoint parts. The first part does not utilize a hidden layers but construct the projection step as in Appendix C.2. The second part corresponds to the first L_1 hidden layers construct the compression 1 step as in Appendix C.3. The third part corresponds to the next $L_{K-1} - L_1$ hidden layers construct the compression 2 step as in Appendix C.4. The last part corresponds to the rest hidden layers construct the learning step as in Appendix C.5.

Now, we describe our construction in detail. To begin with, let us denote a Δ-separated set of inputs by $\mathcal{X}_1$. First, from Lemma 8, one can project $\mathcal{X}_1$ to $\mathcal{X}_2$ such that $\lfloor \mathcal{X}_2 \rfloor \in \binom{[\lceil N^2 \Delta \sqrt{\pi d_x/8} \rceil]}{N}$. Note that the projection step does not require to use hidden layers as it can be absorbed into the linear map before the next hidden layer. Then, from Lemma 9, the first L_1 hidden layers can map $\mathcal{X}_2$ to $\mathcal{X}_3$ such that $\lfloor \mathcal{X}_3 \rfloor \in \binom{[\lfloor N^2/4+1 \rfloor]}{N}$ since

$$
\begin{aligned}
\prod_{\ell=1}^{L_1} \left\lfloor \frac{d_\ell + 1}{2} \right\rfloor \geq \Delta \sqrt{2\pi d_x} \Rightarrow & \prod_{\ell=1}^{L_1} \left\lfloor \frac{d_\ell + 1}{2} \right\rfloor \geq \frac{N^2 \Delta \sqrt{\pi d_x/8}}{N^2/4} \\
\Rightarrow & \frac{N^2}{4} \geq \frac{N^2 \Delta \sqrt{\pi d_x/8}}{\prod_{\ell=1}^{L_1} \lfloor (d_\ell + 1)/2 \rfloor} \\
\Rightarrow & \left\lfloor \frac{N^2}{4} + 1 \right\rfloor \geq \frac{N^2 \Delta \sqrt{\pi d_x/8}}{\prod_{\ell=1}^{L_1} \lfloor (d_\ell + 1)/2 \rfloor} \\
\Rightarrow & \left\lfloor \frac{N^2}{4} + 1 \right\rfloor \geq \left\lceil \frac{N^2 \Delta \sqrt{\pi d_x/8}}{\prod_{\ell=1}^{L_1} \lfloor (d_\ell + 1)/2 \rfloor} \right\rceil.
\end{aligned}
$$

Note that we also utilize Claim 6 for the sequential application of Lemma 9. Consecutively, from Lemma 10, the next $L_{K-1} - L_1$ hidden layers can map $\mathcal{X}_3$ to $\mathcal{X}_4$ such that $\lfloor \mathcal{X}_4 \rfloor \in \binom{[\lceil U/2^{K-2} \rceil]}{N}$ since the third condition holds and

$$
\begin{aligned}
\sum_{\ell=L_{i-1}+1}^{L_i} (d_\ell - 2) \geq 2^{i+3} \Rightarrow & \frac{1}{2} \sum_{\ell=L_{i-1}+1}^{L_i} (d_\ell - 2) \geq 2^{i+2} \\
\Rightarrow & \left\lfloor \frac{1}{2} + \frac{1}{2} \sum_{\ell=L_{i-1}+1}^{L_i} (d_\ell - 2) \right\rfloor \geq 2^{i+2} \\
\Rightarrow & \frac{N}{\left\lfloor \frac{1}{2} + \frac{1}{2} \sum_{\ell=L_{i-1}+1}^{L_i} (d_\ell - 2) \right\rfloor} \leq \frac{N}{2^{i+2}} \\
\Rightarrow & \left\lceil \frac{N}{\left\lfloor \frac{1}{2} + \frac{1}{2} \sum_{\ell=L_{i-1}+1}^{L_i} (d_\ell - 2) \right\rfloor} \right\rceil \leq \frac{N}{2^{i+1}} \\
\Rightarrow & N \times \left\lceil \frac{N}{\left\lfloor \frac{1}{2} + \frac{1}{2} \sum_{\ell=L_{i-1}+1}^{L_i} (d_\ell - 2) \right\rfloor} \right\rceil \leq \frac{N^2/4}{2^{i-1}} \\
\Rightarrow & N \times \left\lceil \frac{N}{\left\lfloor \frac{1}{2} + \frac{1}{2} \sum_{\ell=L_{i-1}+1}^{L_i} (d_\ell - 2) \right\rfloor} \right\rceil \leq \frac{\lfloor N^2/4 + 1 \rfloor}{2^{i-1}} \\
\Rightarrow & N \times \left\lceil \frac{N}{\left\lfloor \frac{1}{2} + \frac{1}{2} \sum_{\ell=L_{i-1}+1}^{L_i} (d_\ell - 2) \right\rfloor} \right\rceil \leq \left\lceil \frac{\lfloor N^2/4 + 1 \rfloor}{2^{i-1}} \right\rceil
\end{aligned}
$$

where we use the inequality $\lceil a \rceil \leq 2a$ for $a \geq 1/2$ and the assumption $K \leq \log N$, i.e., $N/2^{i+1} \geq 1$ for $i \leq \log N - 1$. Here, we also utilize Claim 6 for the sequential application of Lemma 10.

Now, we reinterpret the last condition as follows.

$$
\begin{aligned}
& 2^K \cdot \left(\sum_{\ell=L_{K-1}+1}^{L_K} (d_\ell - 2) \right) \cdot \left\lfloor (L - L_K - 1)/\lceil \log C \rceil \right\rfloor \geq N^2 - 4 \\
\Rightarrow & \left(\sum_{\ell=L_{K-1}+1}^{L_K} (d_\ell - 2) \right) \cdot \left\lfloor (L - L_K - 1)/\lceil \log C \rceil \right\rfloor \geq \frac{N^2/4 + 1}{2^{K-2}} \\
\Rightarrow & \left(\sum_{\ell=L_{K-1}+1}^{L_K} (d_\ell - 2) \right) \cdot \left\lfloor (L - L_K - 1)/\lceil \log C \rceil \right\rfloor \geq \left\lceil \frac{\lfloor N^2/4 + 1 \rfloor}{2^{K-2}} \right\rceil
\end{aligned}
$$

Finally, using the following lemma and the above inequality, the rest hidden layers can map $\mathcal{X}_4$ to their corresponding labels (by choosing $L' := L_3$). The proof of Lemma 13 is presented in Section F.8.

Lemma 13. *For $D, K, L, d_1, \ldots, d_L \in \mathbb{N}$, suppose that there exist $0 < L' < L$ and $r_{L'+1}, \ldots, r_{L-1} \in \mathbb{N}$ satisfying that for $r_{L'} = r_L = 1$*

$$
\left(\sum_{\ell=1}^{L'} (d_\ell - 2) \right) \cdot \left\lfloor \frac{\sum_{\ell=L'+1}^{L-1} r_\ell}{D} \right\rfloor \geq K \quad \textit{and} \quad 2^{r_\ell} + r_\ell + r_{\ell-1} + 3 \leq d_\ell \ \textit{ for all } \ L' + 1 \leq \ell \leq L.
$$

Then, there exists a STEP + ID *network f_θ of L hidden layers having d_ℓ hidden neurons at the ℓ-th hidden layer such that for any finite $\mathcal{X} \subset [0, K)$, for any $y : [K] \to [2^D]$, there exists θ satisfying $f_\theta(x) = y(\lfloor x \rfloor)$ for all $x \in \mathcal{X}$.*

Choose $K \leftarrow \lceil U/2^{K-2} \rceil$ and $r_\ell = 1$ for all ℓ in Lemma 13 completes our construction of the STEP + ID network. Note that we utilize the condition that $d_\ell \geq 7$ for all $L_K < \ell$ here. Using Lemma 7 completes the proof of Theorem 3.

F PROOFS OF TECHNICAL LEMMAS

F.1 PROOF OF LEMMA 8

Since the upper bound holds for any unit vector $u \in \mathbb{R}^{d_x}$, we focus on the lower bound. In addition, if $N = 1$, the result of Lemma 8 trivially holds. Hence, we assume $N \geq 2$, i.e., $\sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2} < 1$. In this proof, we show that given any vector $v \in \mathbb{R}^{d_x}$ such that $\|v\|_2 \neq 0$, a random unit vector $u \in \mathbb{R}^{d_x}$ from the uniform distribution satisfies that

$$\mathbb{P}\left(\frac{|u^\top v|}{\|v\|_2} < \sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right) < \frac{2}{N^2}. \tag{2}$$

This implies that there exists a unit vector $u \in \mathbb{R}^{d_x}$ such that $\sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\|x - x'\|_2 \leq |u^\top(x - x')|$ for all $x, x' \in \mathcal{X}$, due to the following union bound: for $\mathcal{V} = \left\{x - x' : \{x, x'\} \in \binom{\mathcal{X}}{2}, x \leq x'\right\}$ for some total order $\leq$ on $\mathcal{X}$,

$$\mathbb{P}\left(\bigcup_{v\in\mathcal{V}}\left\{\frac{|u^\top v|}{\|v\|_2} < \sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right\}\right) \leq \sum_{v\in\mathcal{V}}\mathbb{P}\left(\frac{|u^\top v|}{\|v\|_2} < \sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right) < \frac{N(N-1)}{2} \times \frac{2}{N^2} < 1.$$

Now we prove (2). To begin with, we show that the following equality holds for any $v \in \mathbb{R}$:

$$\mathbb{P}\left(\frac{|u^\top v|}{\|v\|_2} < \sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right) = \mathbb{P}\left(|u_1| < \sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right).$$

Here, the equality follows from choosing $v/\|v\|_2 = (1, 0, \ldots, 0)$ using the symmetry. Furthermore, $\mathbb{P}\left(|u_1| < \sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right)$ can be bounded by

$$\begin{aligned}
\mathbb{P}\left(|u_1| < \sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right) &= \mathbb{P}\left(0 < u_1 < \sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right) + \mathbb{P}\left(-\sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2} < u_1 \leq 0\right) \\
&= 2 \times \mathbb{P}\left(0 < u_1 < \sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right) \\
&= \frac{2}{\texttt{Area}_{d_x}(1)} \times \int_{\arccos\left(\sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right)}^{\frac{\pi}{2}} \texttt{Area}_{d_x-1}(\sin\phi)d\phi \\
&= 2 \times \frac{\texttt{Area}_{d_x-1}(1)}{\texttt{Area}_{d_x}(1)} \times \int_{\arccos\left(\sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right)}^{\frac{\pi}{2}} \sin^{d_x-2}\phi d\phi \\
&= 2 \times \frac{2\pi^{\frac{d_x-1}{2}}/\Gamma(\frac{d_x-1}{2})}{2\pi^{\frac{d_x}{2}}/\Gamma(\frac{d_x}{2})} \times \int_{\arccos\left(\sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right)}^{\frac{\pi}{2}} \sin^{d_x-2}\phi d\phi \\
&< 2 \times \sqrt{\frac{d_x}{2\pi}} \times \int_{\arccos\left(\sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right)}^{\frac{\pi}{2}} 1 d\phi \\
&= \sqrt{\frac{2d_x}{\pi}}\left(\frac{\pi}{2} - \arccos\left(\sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right)\right) \\
&= \sqrt{\frac{2d_x}{\pi}}\arcsin\left(\sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2}\right) \\
&\leq \sqrt{\frac{2d_x}{\pi}} \times \frac{\pi}{2} \times \sqrt{\frac{8}{\pi d_x}}\frac{1}{N^2} = \frac{2}{N^2}
\end{aligned}$$

where $\texttt{Area}_d(r) := 2\pi^{\frac{d}{2}}r^{d-1}/\Gamma\left(\frac{d}{2}\right)$ denotes the surface area of a hypersphere of radius r in $\mathbb{R}^d$ and $\Gamma(x)$ denotes the gamma function. Here, the second equality follows from the symmetry and $\mathbb{P}(u_1 = 0) = 0$. The first inequality follows from $\sin\phi \leq 1$ and $\frac{\Gamma\left(\frac{d_x}{2}\right)}{\Gamma\left(\frac{d_x-1}{2}\right)} < \sqrt{\frac{d_x}{2}}$ from the Gautschi's inequality (see Lemma 14). The second inequality follows from $\phi \leq \frac{\pi}{2}\sin\phi$ for $0 \leq \phi \leq \frac{\pi}{2}$. This completes the proof of Lemma 8.

Lemma 14 [Gautschi's inequality (Gautschi, 1959)]**.** *For any* $x > 0$*, for any* $s \in (0, 1)$*,*

$$x^{1-s} < \frac{\Gamma(x+1)}{\Gamma(x+s)} < (x+1)^{1-s}.$$

F.2 Proof of Lemma 9

In this proof, we assume that $T = \lceil \frac{K}{\lfloor (d+1)/2 \rfloor} \rceil$ where the other case can be trivially follows from this case. To begin with, we first define a network $f_b(x) : [0, K) \to [0, T)$ for $b = [b_i]_{i=1}^{\lfloor (d-1)/2 \rfloor} \in [T]^{\lfloor (d-1)/2 \rfloor}$ as

$$f_b(x) := \begin{cases} x & \text{if } x < T \\ x + b_1 \mod T & \text{if } T \le x < 2T \\ x + b_2 \mod T & \text{if } 2T \le x < 3T \\ & \vdots \\ x + b_{\lfloor (d-1)/2 \rfloor} \mod T & \text{if } \lfloor \frac{d-1}{2} \rfloor T \le x \end{cases}$$
$$= x - T \times \mathbf{1}[x \ge T] + \sum_{i=1}^{\lfloor (d-1)/2 \rfloor} \left(\left(b_i - \sum_{j=1}^{i-1} b_j \right) \times \mathbf{1}[x \ge iT] - T \times \mathbf{1}[x + b_i \ge (i+1)T] \right). \tag{3}$$

Here, one can easily observe that f_b can be implemented by a Step + Id network of 1 hidden layer and width d as $T \times \mathbf{1}[x \ge T]$ in (3) can be absorbed into $\left(b_i - \sum_{j=1}^{i-1} b_j\right) \times \mathbf{1}[x \ge iT]$ in (3) for $i = 1$.

Now, we show that if $T > \frac{N^2}{4}$, then there exist $b \in [T]^{\lfloor (d-1)/2 \rfloor}$ such that $\left| \lfloor f_b(\mathcal{X}) \rfloor \right| = N$ to complete the proof. Our proof utilizes the mathematical induction on i: If there exist $b_1, \dots, b_{i-1} \in [T]$ such that

$$\lfloor f_b(\{x \in \mathcal{X} : x < iT\}) \rfloor = \begin{pmatrix} [T] \\ \left| \lfloor \{x \in \mathcal{X} : x < iT\} \rfloor \right| \end{pmatrix}, \tag{4}$$

then there exists $b_i \in [T]$ such that

$$\lfloor f_b(\{x \in \mathcal{X} : x < (i+1)T\}) \rfloor = \begin{pmatrix} [T] \\ \left| \lfloor \{x \in \mathcal{X} : x < (i+1)T\} \rfloor \right| \end{pmatrix}. \tag{5}$$

Here, one can observe that the statement trivially holds for the base case, i.e., for $\{x \in \mathcal{X} : x < T\}$. Now, using the induction hypothesis, suppose that there exist $b_1, \dots, b_{i-1} \in [T]$ satisfying (4). Now, we prove that there exists $b_i \in [T]$ such that $\mathcal{S}_{b_i} := \lfloor f_b(\{x \in \mathcal{X} : iT \le x < (i+1)T\}) \rfloor$ does not intersect with $\mathcal{T} := \lfloor f_b(\{x \in \mathcal{X} : x < iT\}) \rfloor$, i.e, (5) holds. Consider the following inequality:

$$\sum_{b_i \in [2^{K-1}]} |\mathcal{S}_{b_i} \cap \mathcal{T}| = |\mathcal{S}_{b_i}| \times |\mathcal{T}| \le \frac{N^2}{4}$$

where the equality follows from the fact that for each $x \in \{x \in \mathcal{X} : iT \le x < (i+1)T\}$, there exists exactly $|\mathcal{T}|$ values of b_i so that $\lfloor x \mod T \rfloor \in \mathcal{T}$. However, since the number of possible choices of b_i is T, if $T > \frac{N^2}{4}$, then, there exists $b_i \in [T]$ such that $\mathcal{S}_{b_i} \cap \mathcal{T} = \emptyset$, i.e., (5) holds. This completes the proof of Lemma 9.

F.3 PROOF OF LEMMA 10

The proof of Lemma 10 is similar to that of Lemma 9. To begin with, we first define a network $f_b(x) : [0, K) \to [0, T)$ for $b = [b_i]_{i=1}^C \in [T]^C$ and $T =: M_1 < M_2 < \cdots < M_{C+1} := K$ so that $|\{x \in \mathcal{X} : M_i \le x < M_{i+1}\}| \le \lceil \frac{N}{C} \rceil \le \lfloor \frac{T}{N} \rfloor$ for all i as

$$
\begin{aligned}
f_b(x) &:= \begin{cases} x & \text{if } x < T \\ x + b_1 \mod T & \text{if } M_1 = T \le x < M_2 \\ x + b_2 \mod T & \text{if } M_2 \le x < M_3 \\ & \vdots \\ x + b_C \mod T & \text{if } M_C \le x \end{cases} \\
&= x - 2T \times \mathbf{1}[x \ge T] + \sum_{i=1}^{C} \left(\left(T + b_i - \sum_{j=1}^{i-1} b_j \right) \times \mathbf{1}[x \ge M_i] \right. \\
&\qquad \left. - T \times \mathbf{1}\left[x \ge \min\{2T - b_i, M_{i+1}\}\right] \right) \qquad (6)
\end{aligned}
$$

where (6) holds as $T \ge \lceil \frac{K}{2} \rceil$. Here, one can easily implement f_b by a STEP + ID network of L hidden layer and d_ℓ neurons at the ℓ-th hidden layer by utilizing one neuron for storing the input x, another one neuron for storing the temporary output, and other neurons implement indicator functions at each layer. Note that one do not require to store x in the last hidden layer and there exists $2C$ indicator functions to implement as $T \times \mathbf{1}[x \ge T]$ in (6) can be absorbed into $\left(T + b_i - \sum_{j=1}^{i-1} b_j\right) \times \mathbf{1}[x \ge M_i]$ in (6) for $i = 1$.

Now, we show that if $T \ge N$, then there exist $b \in [T]^C$ such that $\left|\lfloor f_b(\mathcal{X}) \rfloor\right| = N$ to completes the proof. Our proof utilizes the mathematical induction on i: If there exist $b_1, \ldots, b_{i-1} \in [T]$ such that

$$
\lfloor f_b(\{x \in \mathcal{X} : x < M_i\}) \rfloor = \left(\begin{matrix} [T] \\ \left|\lfloor \{x \in \mathcal{X} : x < M_i\} \rfloor\right| \end{matrix} \right), \qquad (7)
$$

then there exists $b_i \in [T]$ such that

$$
\lfloor f_b(\{x \in \mathcal{X} : x < M_{i+1}\}) \rfloor = \left(\begin{matrix} [T] \\ \left|\lfloor \{x \in \mathcal{X} : x < M_{i+1}\} \rfloor\right| \end{matrix} \right). \qquad (8)
$$

Here, one can observe that the statement trivially holds for the base case, i.e., for $\{x \in \mathcal{X} : x < T\}$. From the induction hypothesis, suppose that there exist $b_1, \ldots, b_{i-1} \in [T]$ satisfying (7). Now, we prove that there exists $b_i \in [T]$ such that $\mathcal{S}_{b_i} := \lfloor f_b(\{x \in \mathcal{X} : M_i \le x < M_{i+1}\}) \rfloor$ does not intersect with $\mathcal{T} := \lfloor f_b(\{x \in \mathcal{X} : x \le M_i\}) \rfloor$, i.e., (8) holds. Consider the following inequality:

$$
\sum_{b_i \in [T]} |\mathcal{S}_{b_i} \cap \mathcal{T}| \le \lfloor \tfrac{T}{N} \rfloor \times \left(N - \lfloor \tfrac{T}{N} \rfloor\right) < T
$$

where the first inequality follows from the fact that $|\mathcal{S}_{b_i}| \le \lfloor \frac{T}{N} \rfloor$, $|\mathcal{T}| \le \left(N - \lfloor \frac{T}{N} \rfloor\right)$, and for each $x \in \{x \in \mathcal{X} : M_i \le x < M_{i+1}\}$, there exists exactly $|\mathcal{T}|$ values of b_i so that $\lfloor x \mod T \rfloor \in \mathcal{T}$. However, since the number of possible choices of b_i is T, there exists $b_i \in [T]$ such that $\mathcal{S}_{b_i} \cap \mathcal{T} = \emptyset$, i.e, (8) holds. This completes the proof of Lemma 10.

F.4 PROOF OF LEMMA 11

In this proof, we explicitly construct f_θ satisfying the desired property stated in Lemma 11. To begin with, we describe the high-level idea of the construction. First we construct a map $g : x \mapsto \left(\lfloor \frac{x}{B} \rfloor, x \mod B\right)$ to transforming an input $x \in [0, K)$ to a pair (a, b) such that $a \in [A]$ and $\lfloor b \rfloor \in [B]$. Here, we give labels to the pair (a, b) corresponding to the input x as $y(a, \lfloor b \rfloor) := y(\lfloor x \rfloor)$. Note that this label is well-defined as if $\lfloor x_1 \rfloor \neq \lfloor x_2 \rfloor$, then $\lfloor g(x_1) \rfloor \neq \lfloor g(x_2) \rfloor$. Now, we construct parameters $w_0, \ldots, w_{A-1}$ containing the label information of $\mathcal{X}$ as

$$w_a := \sum_{c \in [B]} y(a, c) \times 2^{-(c+1)D}, \tag{9}$$

i.e., from the $(\lfloor b \rfloor \cdot D + 1)$-th bit to the $(\lfloor b \rfloor \cdot D + D)$-th bit of the a-th parameter (w_a) contains the label information of $(a, \lfloor b \rfloor)$. Under this construction, we recover the label of $x \in \mathcal{X}$ by first mapping x to the pair (a, b) and extracting the z-th parameter w_a. Then, using STEP, we extract from the $(\lfloor b \rfloor \cdot D + 1)$-th bit to the $(\lfloor b \rfloor \cdot D + D)$-th bit of w_a to recover the label.

Now, we explicitly construct the above procedure. To this end, we introduce the following lemma. The proof of Lemma 15 is presented in Section F.5.

Lemma 15. *For any $A, B, K, L, d_1, \ldots d_L \in \mathbb{N}$ such that $AB \geq K$ and $d_\ell \geq 2$ for all ℓ, for any $w_0, \ldots, w_{A-1} \in \mathbb{R}$, for any finite set $\mathcal{X} \subset [0, K)$, if $\sum_{\ell=1}^{L}(d_\ell - 2) \geq A$, then there exists a* STEP + ID *network f of L layers and d_ℓ neurons at the ℓ-th layer such that $f(x) = (w_{\lfloor x/B \rfloor}, x \mod B)$ for all $x \in \mathcal{X}$.*

From Lemma 15, a STEP + ID network of 1 hidden layer consisting of $A + 2$ hidden neurons can map $x \in \mathcal{X}$ to $(w_{\lfloor x/B \rfloor}, x \mod B)$. Note that this network requires overall $4A + 10$ parameters ($3A + 6$ edges and $A + 4$ biases).

Finally, we introduce the following lemma for extracting from the $(\lfloor b \rfloor \cdot D + 1)$-th bit to the $(\lfloor b \rfloor \cdot D + D)$-th bit of w_a. The proof of Lemma 16 is presented in Section F.6.

Lemma 16. *For any $D, B, R \in \mathbb{N}$, for any finite set $\mathcal{X} \subset [0, B)$, for any $w = \sum_{i=1}^{BD} u_i \times 2^{-i}$ such that $u_i \in \{0, 1\}$, there exists a* STEP + ID *network f of $2\lceil \frac{BD}{R} \rceil$ hidden layers and $\left((2R+5)2^R + 2R^2 + 8R + 7\right)\lceil \frac{BD}{R} \rceil - R2^R - R^2 + 3$ parameters such that $f(x, w) = \sum_{i=1}^{D} u_{\lfloor x \rfloor \cdot D + i} \times 2^{D-i}$ for all $x \in \mathcal{X}$.*

From Lemma 16, a STEP + ID network of $2\lceil \frac{BD}{R} \rceil$ hidden layers and $\left((2R+5)2^R + 2R^2 + 8R + 7\right)\lceil \frac{BD}{R} \rceil - R2^R - R^2 + 3$ parameters. Hence, by combining Lemma 15 and Lemma 16, f_θ can be implemented by a STEP + ID network of $2\lceil \frac{BD}{R} \rceil + 2$ hidden layers and $4A + \left((2R+5)2^R + 2R^2 + 8R + 7\right)\lceil \frac{BD}{R} \rceil - R2^R - R^2 + 3$ parameters. This completes the proof of Lemma 11.

F.5 PROOF OF LEMMA 15

We design f as $f := f_L \circ \cdots \circ f_1(0, x)$ where each f_ℓ represents the function of the ℓ-th layer consisting of d_ℓ neurons. In particular, we construct f_ℓ as follows:

$$f_\ell(w, x) := \Bigg(w + \sum_{i=1}^{d_\ell - 2} u_i \times \mathbf{1}[x \geq B \times (A - c_\ell - i)], \\ x - \sum_{i=1}^{d_\ell - 2} z_i \times \mathbf{1}[x \geq B \times (A - c_\ell - i)] \Bigg)$$

where

$$c_\ell := \sum_{i=1}^{\ell - 1} (d_\ell - 2)$$

$$u_i := \begin{cases} w_{A - c_\ell - i} & \text{if } i = d_\ell - 2 \\ w_{A - c_\ell - i} - w_{A - c_\ell - i - 1} & \text{if } i < d_\ell - 2 \end{cases}$$

$$z_i := \begin{cases} B \times (A - c_\ell - i - 1) & \text{if } i = d_\ell - 2 \\ B & \text{if } i < d_\ell - 2 \end{cases}.$$

Then, f is the desired function and each f_ℓ can be implemented by a STEP + ID networks of 1 hidden layer consisting of d_ℓ hidden neurons. This completes the proof of Lemma 15.

F.6 PROOF OF LEMMA 16

We construct $f(x,w) := 2^{R\lceil BD/R\rceil+D} \times f_{\lceil BD/R\rceil} \circ \cdots \circ f_1(x,w)$ where f_ℓ is defined as

$$f_\ell(x,v) := \begin{cases} \Big(x, v - \sum_{i=1}^R u_{(\ell-1)R+i} \times 2^{-(\ell-1)R-i} \\ \qquad + \sum_{i=1}^R \big(u_{(\ell-1)R+i} \wedge \mathbf{1}[m_{i,\ell} \le x < m_{i,\ell}+1]\big) \times 2^{-r_{i,\ell}}\Big) & \text{if } \ell < \lceil \frac{BD}{R} \rceil \\ v - \sum_{i=1}^R u_{(\ell-1)R+i} \times 2^{-(\ell-1)R-i} \\ \qquad + \sum_{i=1}^R \big(u_{(\ell-1)R+i} \wedge \mathbf{1}[m_{i,\ell} \le x < m_{i,\ell}+1]\big) \times 2^{-r_{i,\ell}} & \text{if } \ell = \lceil \frac{BD}{R} \rceil \end{cases}.$$

where u_i denotes the i-th bit of w in the binary representation, $\wedge$ denotes the binary 'and' operation, and $m_{i,\ell}, r_{i,\ell}$ are defined as

$$m_{i,\ell} := \left\lfloor \frac{(\ell-1)R+i}{D} \right\rfloor$$
$$r_{i,\ell} := \left\lceil \frac{BD}{R} \right\rceil R + \big((\ell-1)R + i - 1 \mod D\big) + 1.$$

Namely, each f_ℓ extracts R bits from the input w and it store the extracted bits to the last bits of v if the extracted bits are in from $(\lfloor x \rfloor \cdot D + 1)$-th bit to the $(\lfloor x \rfloor \cdot D + D)$-th bit of w. Thus, $f(x,w)$ is the desired function for Lemma 16.

To implement each f_ℓ by a STEP+ID network, we introduce Lemma 17. From Lemma 17, a STEP+ID network of $2\lceil \frac{BD}{R} \rceil$ hidden layers consisting of $2^R + R + 1$ and $R+2$ hidden neurons alternatively can map (x,w) to $\sum_{i=1}^D u_{\lfloor x \rfloor \cdot D + i} \times 2^{D-i}$ for all $x \in \mathcal{X}$. By considering the input dimension 2 and the output dimension 1, this network requires $\big((2R+5)2^R + 2R^2 + 8R + 7\big)\lceil \frac{BD}{R} \rceil - R2^R - R^2 + 3$ parameters ($\big((2R+4)2^R + 2R^2 + 6R + 4\big)\lceil \frac{BD}{R} \rceil - R2^R - R^2 + 2$ edges and $(2^R + 2R + 3)\lceil \frac{BD}{R} \rceil + 1$ biases). This completes the proof of Lemma 16.

Lemma 17. *A* STEP + ID *network of* 2 *hidden layers having* $2^R + R + 1$ *and* $R+2$ *hidden neurons at the first and the second hidden layer, respectively, can implement* f_ℓ.

Proof of Lemma 17. We construct $f_\ell := g_3 \circ (g_2 \oplus g_1)$ where $g_2 \oplus g_1$ denotes the function concatenating the outputs of g_1, g_2. In this proof, we mainly focus on constructing f_ℓ for $\ell < \lceil \frac{BD}{R} \rceil$ since $f_{\lceil BD/R \rceil}$ can be implemented similarly. We define g_1, g_2, g_3 as

$$\begin{aligned} g_1(x,v) :=& \Bigg(x, v, \sum_{i=0}^{2^R-1} \eta_{1,i} \times \mathbf{1}[i \times 2^{-\ell R} \le v < (i+1) \times 2^{-\ell R}], \\ & \qquad \vdots, \\ & \qquad \sum_{i=0}^{2^R-1} \eta_{R,i} \times \mathbf{1}[i \times 2^{-\ell R} \le v < (i+1) \times 2^{-\ell R}]\Bigg) \\ =& (x, v, u_{(\ell-1)R+1}, \ldots, u_{\ell R}) \\ g_2(x,v) :=& \Big(\mathbf{1}[m_{1,\ell} \le x < m_{1,\ell}+1], \ldots, \mathbf{1}[m_{R,\ell} \le x < m_{R,\ell}+1]\Big) \\ g_3 \circ (g_1 \oplus g_2) :=& \Bigg(x, v - \sum_{i=1}^R u_{(\ell-1)R+i} \times 2^{-(\ell-1)R-i} \\ & \qquad + \sum_{i=1}^R \mathbf{1}\big[u_{(\ell-1)R+i} + \mathbf{1}[m_{i,\ell} \le x < m_{i,\ell}+1] \ge 2\big] \times 2^{-r_{i,\ell}}\Bigg) \\ =& \Bigg(x, v - \sum_{i=1}^R u_{(\ell-1)R+i} \times 2^{-(\ell-1)R-i}) \\ & \qquad + \sum_{i=1}^R (u_{(\ell-1)R+i} \wedge \mathbf{1}[m_{i,\ell} \le x < m_{i,\ell}+1]) \times 2^{-r_{i,\ell}}\Bigg). \end{aligned}$$

where $\eta_{r,i}$ is a constant such that $\eta_{r,i} = 1$ if $i \times 2^{-\ell R} \leq x < (i+1) \times 2^{-\ell R}$ implies that the $((\ell-1)R + r)$-th bit of x is 1 and $\eta_{r,i} = 0$ otherwise. Here, one can easily observe that g_1 can be implemented by a linear combinations of $\mathbf{1}[v \geq 2^{-\ell R}], \ldots, \mathbf{1}[v \geq (2^R - 1) \times 2^{-\ell R}]$ as it trivially holds that $\mathbf{1}[v \geq 0]$ and $\mathbf{1}[v < 2^{-(\ell-1)R}]$, i.e., $2^R - 1$ indicator functions are enough for g_1. Hence, g_1 can be implemented by a STEP + ID network of 1 hidden layer consisting of $2^R + 1$ hidden neurons where additional 2 neurons are for passing x, v. In addition, g_2 can be implemented by a STEP + ID network of R hidden neurons. Finally, g_3 can be implemented by a STEP + ID network of 1 hidden layer consisting of $R + 2$ hidden neurons (R neurons for R indicator functions and 2 neurons for passing x, v).

Therefore, f_ℓ can be implemented by a STEP + ID network of 2 hidden layers consisting of $2^R + R + 1$ hidden neurons for the first hidden layer and $R + 2$ hidden neurons for the second hidden layer. Note that implementation within two hidden layer is possible since the outputs of g_1, g_2 are simply linear combination of their hidden activation values and hence, can be absorbed into the linear map between hidden layers. This completes the proof of Lemma 17. □

F.7 PROOF OF LEMMA 12

The main idea of the proof of Lemma 12 is identical to that of Lemma 11. Recall A, B and $w_0, \dots, w_{A-1} \in \mathbb{R}$ from the proof of Lemma 11. From Lemma 15, for any finite set $\mathcal{X} \subset [0, K)$, for any $w_0, \dots, w_{A-1} \in \mathbb{R}$, a STEP + ID network of A hidden layers and width 3 can map x to $(w_{\lfloor x/B \rfloor}, x \mod B)$ for all $x \in \mathcal{X}$. Now, we introduce the following lemma replacing Lemma 16.

Lemma 18. *For any $D, B \in \mathbb{N}$, for any finite set $\mathcal{X} \subset [0, B)$, for any $w = \sum_{i=1}^{DB} u_i \times 2^{-i}$ for some $u_i \in \{0, 1\}$, there exists a* STEP + ID *network f of $(2D+1)B$ hidden layers and width 3 such that $f(x, w) = \sum_{i=1}^{D} u_{\lfloor x \rfloor \cdot D + i} \times 2^{-i}$ for all $x \in \mathcal{X}$.*

Using Lemma 15 and Lemma 18, one can easily find a STEP + ID network of $A + (2D+1)B$ hidden layers and width 3 satisfying the condition in Lemma 12. This completes the proof of Lemma 12.

Proof of Lemma 18. We construct $f(x, w) := 2^{D(B+1)} \times f_{DB} \circ g_{DB} \circ h_{DB} \circ \cdots \circ f_1 \circ g_1 \circ h_1$ where f_ℓ, g_ℓ, h_ℓ are defined as

$$
\begin{aligned}
h_\ell(x, w) &:= \begin{cases} (x, w) & \text{if } \ell \mod D \neq 1 \\ (x - 1 + 3K \times \mathbf{1}[x < 0], w) & \text{if } \ell \mod D = 1 \end{cases} \\
g_\ell(x, w) &:= \left(x, w - 2^{-\ell} \times \mathbf{1}[w \geq 2^{-\ell}], \mathbf{1}[w \geq 2^{-\ell}]\right) \\
f_\ell(x, w, n) &= \left(x, w + 2^{-DK - d(\ell)} \times \mathbf{1}[x - n < -1]\right)
\end{aligned}
$$

where $d(\ell) := \ell - \lfloor \frac{\ell - 1}{D} \rfloor + 1$. Now, we explain the constructions of f_ℓ, g_ℓ, h_ℓ. Let $b \in [0, B)$ and $v \in [0, 1)$ be inputs of f, i.e., consider $f(b, v)$. First, the indicator function in h_ℓ is activated only at $\ell = (\lfloor b \rfloor + 1) \cdot D + 1$ as $b < B$. In particular, the first entry of the output of $h_{(\lfloor b \rfloor + 1) \cdot D + 1}$ is greater than $2B$ and this is the maximum value of the first entry of the output of $h_{(\lfloor b \rfloor + 1) \cdot D + 1}$ as it monotonically decreases as ℓ grows. The indicator function in g_ℓ extracts and outputs the ℓ-th bit of u. Lastly, f_ℓ add the ℓ-th bit of u if and only if $\ell \in \{\lfloor b \rfloor \cdot D + 1, \dots, (\lfloor b \rfloor + 1) \cdot D\}$. This is because $x \in [-1, 0)$ if and only if $\ell \in \{\lfloor b \rfloor \cdot D + 1, \dots, (\lfloor b \rfloor + 1) \cdot D\}$.

Here, h_ℓ at $\ell \mod D = 1$ can be implemented by a STEP + ID network of 1 hidden layer and width 3, g_ℓ can be implemented by a STEP + ID network of 1 hidden layers and width 3, and f_ℓ can be implemented by a STEP + ID network of 1 layers and width 3. Hence, f can be implemented by a STEP + ID network of $(2D+1)B$ hidden layers and width 3. This completes the proof of Lemma 18. □

F.8 PROOF OF LEMMA 13

The proof of Lemma 13 is almost identical to the proof of Lemma 11. Recall A, B and $w_0, \dots, w_{A-1} \in \mathbb{R}$ as in the proof of Lemma 11. From Lemma 19, for any finite set $\mathcal{X} \subset [0, K)$, for any $w_0, \dots, w_{A-1} \in \mathbb{R}$, the first L' hidden layers of the STEP + ID network f_θ can map x to $(w_{\lfloor x/B \rfloor}, x \mod B)$ for all $x \in \mathcal{X}$. Now, we introduce the following lemma replacing Lemma 16. Using Lemma 19 completes the proof of Lemma 13.

Lemma 19. *For any $D, B, R, L, d_1, \dots, d_L \in \mathbb{N}$, for any finite set $\mathcal{X} \subset [0, B)$, for any $w = \sum_{i=1}^{BD} u_i \times 2^{-i}$ such that $u_i \in \{0, 1\}$, suppose that there exists $r_1, \dots, r_{L-1} \in \mathbb{N}$ satisfying that for $r_0 = r_L = 1$,*

$$\sum_{\ell=1}^{L-1} r_\ell \geq BD,$$
$$2^{r_\ell} + r_\ell + r_{\ell-1} + 3 \leq d_\ell \qquad \textit{for all} \qquad 1 \leq \ell \leq L.$$

Then, there exists a STEP + ID *network f of L hidden layers having d_ℓ hidden neurons at the ℓ-th hidden layer such that $f(x, w) = \sum_{i=1}^{D} u_{\lfloor x \rfloor \cdot D + i} \times 2^{D-i}$ for all $x \in \mathcal{X}$.*

Proof of Lemma 19. The proof of Lemma 19 utilizes the network constructions in the proofs of Lemma 16 and Lemma 17. In particular, we construct $f(x, w) := f_{L-1} \circ \cdots \circ f_1(x, w)$ defined as

$$f_\ell(x, v) := \begin{cases} \Big(x, v - \sum_{i=1}^{r_\ell} u_{R_\ell + i} \times 2^{-R_\ell - i} \\ \qquad + \sum_{i=1}^{r_\ell} \big(u_{R_\ell+i} \wedge \mathbf{1}[m_{i,\ell} \leq x < m_{i,\ell} + 1]\big) \times 2^{-s_{i,\ell}}\Big) & \text{if } \ell < \lceil \frac{BD}{R} \rceil \\ v - \sum_{i=1}^{r_\ell} u_{R_\ell+i} \times 2^{-R_\ell - i} \\ \qquad + \sum_{i=1}^{r_\ell} \big(u_{R_\ell+i} \wedge \mathbf{1}[m_{i,\ell} \leq x < m_{i,\ell} + 1]\big) \times 2^{-s_{i,\ell}} & \text{if } \ell = \lceil \frac{BD}{R} \rceil \end{cases}.$$

where u_i denotes the i-th bit of w in the binary representation, $\wedge$ denotes the binary 'and' operation, and $R_\ell, m_{i,\ell}, s_{i,\ell}$ are defined as

$$R_\ell := \sum_{i=1}^{\ell-1} r_\ell,$$
$$m_{i,\ell} := \left\lfloor \frac{R_\ell + i}{D} \right\rfloor,$$
$$s_{i,\ell} := R_L + \big(R_\ell + i - 1 \mod D\big) + 1.$$

Note that $R_1 = 0$ as the summation starts from $i = 1$ and $R_L = \sum_{\ell=1}^{L-1} r_\ell$. Namely, each f_ℓ extracts r_ℓ bits from the input w and it store the extracted bits to the last bits of v if the extracted bits are in from $(\lfloor x \rfloor \cdot D + 1)$-th bit to the $(\lfloor x \rfloor \cdot D + D)$-th bit of w. Thus, $f(x, w)$ is the desired function for Lemma 19.

We construct $f_\ell(x, v)$ using the ℓ-th hidden layer and the $(\ell + 1)$-th hidden layer, i.e., there exists an overlap between constructions of $f_\ell(x, v)$ and $f_{\ell+1}(x, v)$. In particular, Lemma 17 directly allows us to obtain such a construction under the assumption in Lemma 13 that

$$2^{r_\ell} + r_\ell + r_{\ell-1} + 3 \leq d_\ell \qquad \text{for all} \qquad 1 \leq \ell \leq L.$$

This completes the proof of Lemma 19. □