

756 A Proof of theorem 1

757 Recall that

$$\mathcal{H} = \mathcal{H}(k, m_0, \dots, m_L, r), \quad (26)$$

758 where r is the cardinality of the discretized group $G^r := \{g_1, g_2, \dots, g_r\}$. The parameter k determines
759 the number of basis functions

$$\mathcal{K}_s : G^r \rightarrow \mathbb{R}, \quad s = 1, \dots, k, \quad (27)$$

760 in the parametrization of the kernel function

$$\mathcal{K}_{\mathbf{w}} = \sum_{s=1}^k w_s \mathcal{K}_s.$$

761 The other parameters m_0, \dots, m_L define the network architecture, and W_ℓ represents the number of
762 parameters in the GCNN up to layer ℓ . The class \mathcal{H} consists of all functions that can be represented
763 by a neural network with this architecture.

764 We restate Theorem 1 for convenience:

765 **Theorem 3** (Theorem 1). *The VC dimension of the GCNN class $\mathcal{H} = \mathcal{H}(k, m_0, \dots, m_L, r)$ with*
766 *$r > 1$, is bounded from above by*

$$UB(\mathcal{H}) := L + 1 + 4 \left(\sum_{\ell=1}^L W_\ell \right) \log_2 \left(8er \sum_{\ell=1}^L m_\ell \right). \quad (28)$$

767 For the proof, we consider an input consisting of m functions from G^r to \mathbb{R}^{m_0} , denoted by

$$F_m := \{f_1, \dots, f_m\}. \quad (29)$$

768 To prove we use the following known result:

769 **Lemma 4.** [Lemma 1, [2]] *Let $p_1, \dots, p_{\tilde{m}}$ be polynomials of degree at most t depending on $n \leq \tilde{m}$*
770 *variables. Then*

$$\Pi := |\{(\text{sign}(p_1(x)), \dots, \text{sign}(p_{\tilde{m}}(x))) : x \in \mathbb{R}^n\}| \leq 2 \left(\frac{2e\tilde{m}t}{n} \right)^n.$$

771 We denote $S(\ell)$ the number of regions in the parameter space \mathbb{R}^{W_ℓ} , such that in each region, the
772 GCNN units in the ℓ -th layer (denoted by $\{h_{\ell,j}(g) \mid j \leq m_\ell, f \in F_m, g \in G^r\}$) behave like a fixed
773 polynomial of degree at most ℓ in the W_ℓ network parameters that occur up to layer ℓ .

774 **Lemma 5.** *Let \mathcal{H} be the class of GCNNs defined in (26), with at most W_ℓ parameters up to layer*
775 *$\ell \in \{1, \dots, L\}$. If F_m is the class of functions defined in (29), and $S(\ell)$ is as defined above, then for*
776 *$\ell = 0, 1, \dots, L-1$,*

$$S(\ell+1) \leq 2 \left(\frac{2em_{\ell+1}mr(\ell+1)}{W_{\ell+1}} \right)^{W_{\ell+1}} S(\ell). \quad (30)$$

777 *Moreover, the GCNN units $\{h_{\ell+1,j}(g) \mid j \leq m_{\ell+1}, f \in F_m, g \in G^r\}$ with $h_{\ell+1,j}$ defined for*
778 *different functions $f \in F_m$, are piecewise polynomials of degree $\leq \ell+1$ in the network parameters.*

779 *Proof.* As a first step of the proof, we show that any GCNN unit $h_{\ell,j}$ of any layer $\ell \in \{0, \dots, L\}$
780 and $j \in \{1, \dots, m_\ell\}$ is a piecewise polynomial of degree at most ℓ . We proceed by induction on the
781 layers ℓ .

782 For the base case $\ell = 0$, the GCNN units $h_{0,j}$ for $j \leq m_0$ correspond to the input of the network. As
783 the inputs are independent of the network parameters, $h_{0,j}$ are polynomials of degree 0.

784 Assume the statement holds for all layers up to ℓ . We now prove it for layer $\ell+1$. The GCNN unit in
785 layer $\ell+1$ is defined by a convolution with the feature maps from the previous layer, that is,

$$h_{\ell+1,j} = \sigma \left(\sum_{i=1}^{m_\ell} \mathcal{K}_{\mathbf{w}_{ij}^{(\ell)}} * h_{\ell,i} - b_j^{(\ell)} \right),$$

where the convolutional filter is expanded in terms of the fixed basis functions K_s via $K_{\mathbf{w}} = \sum_{s=1}^k w_s K_s$. For fixed network parameters, $g \mapsto h_{\ell,j}(g)$ is a function of the group, with $h_{\ell,j}(g) \in \mathbb{R}$ for any $g \in G^r$. By the induction hypothesis, $h_{\ell,j}(g)$ are piecewise polynomials of degree at most ℓ with respect to the network parameters, with the polynomial pieces depending on the network input and the group element g .

Next, for any input and any group element g , the term $(K_s * h_{\ell,j})(g)$ can be written as

$$(K_s * h_{\ell,j})(g) = \sum_{g' \in G^r} K_s(g^{-1} \circ g') \cdot h_{\ell,j}(g').$$

Since $h_{\ell,j}(g')$ is a piecewise polynomial of degree at most ℓ , it follows that $(K_s * h_{\ell,j})(g)$ is also a piecewise polynomial of degree at most ℓ . Thus, the convolution

$$(K_{\mathbf{w}} * h_{\ell,j})(g) = \sum_{s=1}^k w_s (K_s * h_{\ell,j})(g)$$

is a weighted sum of piecewise polynomials, which remains a piecewise polynomial. However, multiplying by the weights w_s increases the degree of the polynomial, making it at most $\ell + 1$. Subtracting the bias term $b_j^{(\ell)}$ and applying the ReLU activation function may increase the number of pieces, but does not increase the degree of the polynomials. Therefore, for any input and any group element g , the GCNN unit $h_{\ell+1,j}(g)$ remains a piecewise polynomial with degree $\leq \ell + 1$. This completes the proof by induction.

Next, we show (30). Each GCNN unit in layer $\ell + 1$ is computed by

$$\sigma \left(\sum_{i=1}^{m_\ell} K_{\mathbf{w}_{ij}^{(\ell)}} * h_{\ell,i} \right),$$

with $\sigma(x) = \max\{x, 0\}$ the ReLU activation function. As mentioned above, applying the ReLU function can increase the number of regions in the parameter space where the GCNN units behave as polynomials. This occurs, as the ReLU function either outputs the input itself (for positive values) or zero (otherwise). As a result its application decomposes each of the $S(\ell)$ regions of the parameter space in layer ℓ in multiple subregions. To bound this number of subregions, we need to count the number of possible sign pattern that can arise after applying the ReLU activation.

Fixing one of the $S(\ell)$ regions of layer ℓ , by definition, all functions $h_{\ell,j}(g)$ are polynomials in the parameters of degree at most ℓ . Each of the $m_{\ell+1}$ GCNN units in layer $\ell + 1$, is then also a polynomial of degree at most $\ell + 1$, leading to at most $m_{\ell+1}mr$ polynomials, where m is the number of input functions defined in (29) and r is the resolution. Applying Lemma 4 to the $\tilde{m} = m_{\ell+1}mr$ polynomials of degree $t = \ell + 1$ depending on $n = W_{\ell+1}$ parameters leads to at most

$$2 \left(\frac{2em_{\ell+1}mr(\ell+1)}{W_{\ell+1}} \right)^{W_{\ell+1}}.$$

different sign patterns for each region of $S(\ell)$. This shows (30) and concludes the proof. \square

Lemma 6. Let \mathcal{H} be the class of GCNNs defined in (26), with at most W_ℓ parameters up to layer $\ell \leq L$, and m_ℓ GCNN units in layer ℓ . For any integer $m > 0$, the growth function of this class can be bounded by

$$\Pi_{\mathcal{H}}(m) \leq 2^L \prod_{\ell=1}^L \left(\frac{2emr m_\ell \ell}{W_\ell} \right)^{W_\ell} 2 \left(\frac{2emL}{W_L + 1} \right)^{W_L + 1}.$$

Proof. Lemma 5 shows that after L layers, there are at most

$$2^L \prod_{\ell=1}^L \left(\frac{2emr m_\ell \ell}{W_\ell} \right)^{W_\ell}$$

regions in the parameter space \mathbb{R}^W , on which the GCNN units in the last layer $\{h_{L,i}(g) \mid i \leq m_L, f \in F_m, g \in G^r\}$ behave like a fixed polynomial function of degree $\leq L$ in W variables.

Recall that the final output of the neural network, is obtained by applying average pooling to the outputs of the GCNN units in the last layer. This implies that, for a fixed network architecture and input, the output of the neural network is a piecewise polynomial of degree at most L , depending on all W_L network parameters. Since there are m possible inputs f_1, \dots, f_m , we get m piecewise polynomials, each corresponding to one of these inputs. Bounding the growth function $\Pi_{\mathcal{H}}(m)$ now means we need to count the number of different sign patterns that arises for classifiers in $\text{sign}(\mathcal{H})$. For that, we recall that by Definition 3.2 in the main article,

$$\text{sign}(\mathcal{H}) := \{\text{sign}(h_{\mathbf{w}} - b) \mid h_{\mathbf{w}} \in \mathcal{H}, \mathbf{w} \in R^{W_L}, b \in \mathbb{R}\}.$$

Applying Lemma 4 to m polynomials of the form $h_{\mathbf{w}} - b$ of degree at most L and $W_L + 1$ variables leads to no more than

$$2 \left(\frac{2emL}{W_L + 1} \right)^{W_L + 1} \quad (31)$$

distinct sign patterns that the classifiers in $\text{sign}(\mathcal{H})$ can produce.

Thus, the growth function within each region, where the GCNN units in the last layer $\{h_{L,i}(g) \mid i \leq m_L, f \in F_m, g \in G^r\}$ behave like a fixed polynomial function in W variables, is bounded by (31). As a result, we conclude that the overall growth function $\Pi_{\mathcal{H}}(m)$ is bounded by

$$S(L) \cdot 2 \left(\frac{2emL}{W_L + 1} \right)^{W_L + 1} = 2^L \prod_{\ell=1}^L \left(\frac{2emr m_{\ell} \ell}{W_{\ell}} \right)^{W_{\ell}} \cdot 2 \left(\frac{2emL}{W_L + 1} \right)^{W_L + 1}.$$

This completes the proof. \square

For the proof of the Theorem 3 we also use the following technical lemma

Lemma 7. Suppose $2^{\tilde{m}} \leq 2^{\kappa} \left(\frac{\tilde{m} \cdot \tilde{r}}{\tilde{w}} \right)^{\tilde{w}}$ for some $\tilde{r} \geq 16$ and $\tilde{m} \geq \tilde{w} \geq \kappa \geq 0$. Then $\tilde{m} \leq \kappa + \tilde{w} \log_2(2\tilde{r} \log_2 \tilde{r})$.

Proof of Theorem 3. Let $m := \text{VC}(\mathcal{H})$. For convenience, define the sum

$$\tilde{W} := \sum_{i=1}^L W_i, \quad (32)$$

where W_i denotes the number of parameters of a GCNN in \mathcal{H} up to layer i .

We consider two complementary cases and prove the theorem for each of them separately.

Case 1: $m < \tilde{W} + W_L + 1$. In this case, we have $\tilde{W} + W_L + 1 < 3\tilde{W} < UB(\mathcal{H})$, where $UB(\mathcal{H})$ is defined in (28). For the latter inequality we use that $\log_2 \left(8er \sum_{\ell=1}^L m_{\ell} \right) > 1$. Therefore, Theorem 3 holds.

Case 2: $m \geq \tilde{W} + W_L + 1$. Since m represents the VC dimension of \mathcal{H} , it follows from the definition of the VC dimension (see Definition 3.1 in the main article) that $\Pi_{\mathcal{H}}(m) = 2^m$. Applying Lemma 6 gives us

$$\Pi(m) = 2^m \leq 2^{L+1} \prod_{\ell=1}^L \left(\frac{2emr m_{\ell} \ell}{W_{\ell}} \right)^{W_{\ell}} \left(\frac{2emL}{W_L + 1} \right)^{W_L + 1}. \quad (33)$$

Next, we apply the weighted arithmetic-geometric mean (AM-GM) inequality to the right side of (33), using weights $W_{\ell}/(\tilde{W} + W_L + 1)$ for $\ell = 1, 2, \dots, L$, and $W_L/(\tilde{W} + W_L + 1)$, where \tilde{W} is defined in (32). This yields

$$2^m \leq 2^{L+1} \left(\frac{2em(r \sum_{\ell=1}^L \ell m_{\ell} + L)}{\tilde{W} + W_L + 1} \right)^{\tilde{W} + W_L + 1}.$$

849 The last step of the proof involves applying Lemma 7 in [2] to this inequality, which provides an
 850 upper bound for m . Before doing so, we must verify that all conditions of the lemma are satisfied.
 851 In our case, \tilde{m} corresponds to m , κ to $L + 1$, \tilde{w} to $\tilde{W} + W_L + 1$, and \tilde{r} to $2e(r \sum_{\ell=1}^L \ell m_\ell + L)$.
 852 Since $r \sum_{\ell=1}^L \ell m_\ell + L > 2$, we have $\tilde{r} > 16$. Moreover, we are considering the case where
 853 $m \geq \tilde{W} + W_L + 1$, and it is straightforward to verify that $\tilde{W} + W_L + 1 \geq L + 1 > 0$. Therefore all
 854 conditions of Lemma 7 in [2] are indeed satisfied and we obtain

$$m \leq (L + 1) + 2\tilde{W} \log_2 \left(4e \left(r \sum_{\ell=1}^L \ell m_\ell + L \right) \cdot \log_2 \left(2e \left(r \sum_{\ell=1}^L \ell m_\ell + L \right) \right) \right).$$

855 To simplify this inequality, we use that for all $a \geq 1$, $\log_2(2a \log_2 a) = \log_2(2a) + \log_2(\log_2 a) \leq$
 856 $2 \log_2(2a)$. Substituting $a = 2e \left(r \sum_{\ell=1}^L \ell m_\ell + L \right)$, we note that $a \leq 4er \sum_{\ell=1}^L \ell m_\ell$ and obtain

$$m \leq (L + 1) + 4\tilde{W} \log_2 \left(8er \sum_{\ell=1}^L \ell m_\ell \right),$$

857 completing the proof of the theorem. \square

858 B Proof of theorem 2

859 In this section, we provide the detailed proof of lower bound on VC dimension, along with the proofs
 860 for Lemmas 2, 3, and their corresponding Corollaries 2 and 3.

861 The class

$$\mathcal{H}_{W,L,r} := \{ \mathcal{H}(k, m_0, \dots, m_L, r) \mid \ell \leq L, W_L \leq W \}, \quad (34)$$

862 includes all GCNN architectures with a total number of parameters bounded by W , a maximum depth
 863 of L , and r representing the cardinality of the discretized group $G^r := \{g_1, g_2, \dots, g_r\}$ containing
 864 the identity element e .

865 Next, we recall that $\mathcal{F}(m_0, \dots, m_L)$ represents the class of fully connected feedforward ReLU
 866 networks with L layers, where m_i denotes the number of units in the i -th layer for $i = 1, \dots, L$. The
 867 output of the last hidden layer of any neural network $\tilde{h}_{\mathbf{w}} \in \mathcal{F}(m_0, \dots, m_L)$, with parameters \mathbf{w} , can
 868 be written as a vector of size m_L , that is, $(\tilde{h}_{\mathbf{w}}^{(1)}, \dots, \tilde{h}_{\mathbf{w}}^{(m_L)})$.

869 Finally, we define the class

$$\mathcal{F}_{W,L} := \{ \mathcal{F} = \mathcal{F}(m_0, \dots, m_L) \mid \ell \leq L, W_L(\mathcal{F}) \leq W \}, \quad (35)$$

870 consisting of DNNs with at most L hidden layers and a total number of weights not exceeding W .

871 **Lemma 8.** *Consider GCNNs where the G -correlation uses kernels from a one-dimensional vector*
 872 *space with a fixed basis given by the indicator function of the identity element e . For every $\tilde{h}_{\mathbf{w}} \in$*
 873 *$\mathcal{F}(m_0, \dots, m_L)$, there exists a GCNN $h_{\mathbf{w}}$ with the same number of channels in each layer, i.e.,*
 874 *$h_{\mathbf{w}} \in \mathcal{H}(1, m_0, \dots, m_L, r)$, and parameters \mathbf{w} , such that for any input function $f : G^r \rightarrow \mathbb{R}^{m_0}$,*

$$\sum_{i=1}^{m_L} \sum_{j=1}^r \tilde{h}_{\mathbf{w}}^{(i)}(f(g_j)) = h_{\mathbf{w}}(f).$$

875 *Proof.* Write $\mathcal{H} := \mathcal{H}(1, m_0, \dots, m_L, r)$. Consider a fixed input function $f : G^r \rightarrow \mathbb{R}^{m_0}$ and a
 876 weight vector $\mathbf{w} \in \mathbb{R}^W$. Recall that the number of parameters in a GCNN is given by

$$W_L := \sum_{j=1}^L m_j(km_{j-1} + 1), \quad (36)$$

877 where k is the dimension of the kernel space. In our case $k = 1$ and the number of parameters for a
 878 GCNN with architecture \mathcal{H} is

$$W_L = \sum_{j=1}^L m_j(m_{j-1} + 1). \quad (37)$$

879 This coincides with the number of parameters in a DNN with architecture $\mathcal{F}(m_0, \dots, m_L)$. Conse-
 880 quently, the same weight vector $\mathbf{w} \in \mathbb{R}^W$ defines both, a DNN function $\tilde{h}_{\mathbf{w}}$ and a GCNN function
 881 $h_{\mathbf{w}} \in \mathcal{H}$ when the input is fixed to f .

882 We now show that the outputs of the computational units in $\tilde{h}_{\mathbf{w}}$ and $h_{\mathbf{w}}$ are equal when applied to
 883 $f(g)$ and g , respectively. Specifically, we aim to prove that

$$\tilde{h}_{\ell,i}(f(g_j)) = h_{\ell,i}(g_j),$$

884 where $\tilde{h}_{\ell,i}$ denotes a DNN computational unit in layer ℓ of $\tilde{h}_{\mathbf{w}}$, with parameters fixed to \mathbf{w} , and $h_{\ell,i}$
 885 represents a GCNN computational unit in layer ℓ , with parameters fixed to \mathbf{w} and input set to f . We
 886 prove this by induction on the layer ℓ .

887 The statement holds trivially for the input layer, as $\tilde{h}_{0,i}(f(g_j)) = h_{0,i}(g_j)$ for any $g_j \in G^r$. Assuming
 888 it holds for all layers up to $\ell - 1$, we now prove it for layer ℓ .

889 Let \mathbb{K} denote the indicator of the identity element $e \in G^r$. By calculating the G-correlation between
 890 \mathbb{K} and f , we obtain $\mathbb{K} * f = f$. Combining this with the definition of the GCNN unit (see (9) in the
 891 main article) and the induction hypothesis, we have

$$\begin{aligned} \tilde{h}_{\ell,i}(g_j) &:= \sigma \left(\sum_{t=1}^{m_{\ell-1}} \mathbf{w}_{t,i}^{(\ell-1)} \tilde{h}_{\ell-1,t}(g_j) - b_i^{(\ell)} \right) \\ &= \sigma \left(\sum_{t=1}^{m_{\ell-1}} \mathbf{w}_{t,i}^{(\ell-1)} h_{\ell-1,t}(g_j) - b_i^{(\ell)} \right) && \text{(induction assumption)} \\ &= \sigma \left(\sum_{t=1}^{m_{\ell-1}} \left(\mathbf{w}_{t,i}^{(\ell-1)} \mathbb{K} * h_{\ell-1,t} \right) (g_j) - b_i^{(\ell)} \right) && \text{(property of } \mathbb{K}) \\ &= \sigma \left(\sum_{t=1}^{m_{\ell-1}} \left(\mathcal{K}_{\mathbf{w}_{t,i}^{(\ell-1)}} * h_{\ell-1,t} \right) (g_j) - b_i^{(\ell)} \right) && \text{(definition of learned kernel)} \\ &= h_{\ell,i}(g_j) && \text{(definition of GCNN unit).} \end{aligned}$$

892 This shows that the outputs of the computational units in $\tilde{h}_{\mathbf{w}}$ and $h_{\mathbf{w}}$ are equal when applied to $f(g)$
 893 and g , respectively.

894 Finally, the outputs of $\tilde{h}_{\mathbf{w}} := (\tilde{h}_{\mathbf{w}}^{(1)}, \dots, \tilde{h}_{\mathbf{w}}^{(m_L)})$ can be rewritten into the form

$$\sum_{i=1}^{m_L} \sum_{j=1}^r \tilde{h}_{\mathbf{w}}^{(i)}(f(g_j)) = \sum_{i=1}^{m_L} \sum_{j=1}^r \tilde{h}_{L,i}(g_j) = \sum_{i=1}^{m_L} \sum_{j=1}^r h_{L,i}(g_j) = h_{\mathbf{w}}(f),$$

895 concluding the proof of the lemma. \square

896 Next, we prove Lemma 2. The key ideas and steps of the proof have already been outlined in the
 897 main article, so here we will focus on the formal statements that still needs to be established.

898 Recall that the indicator neural network

$$\mathbf{1}_{(a,b,\epsilon)} \tag{38}$$

899 is a shallow ReLU network with four neurons in the hidden layer (see (25) in the main article). It
 900 approximates the indicator function on the interval $[a, b]$ in the sense that $\mathbf{1}_{(a,b,\epsilon)}(x) = 1$ if $x \in [a, b]$,
 901 and $\mathbf{1}_{(a,b,\epsilon)}(x) = 0$ if $x < a - \epsilon$ or $x > b + \epsilon$.

902 **Lemma 9.** [Lemma 2] For $L > 3$ let $\mathcal{H}_{6W,L+1,r}$ be the class of GCNNs defined as in (34) and $\mathcal{F}_{W,L}$
 903 be the class of DNNs defined as in (35). Then

$$\text{VC}(\mathcal{H}_{6W,L+1,r}) \geq \text{VC}(\mathcal{F}_{W,L}).$$

904 *Proof.* Let m be the VC dimension of the class of DNNs $\mathcal{F}_{W,L}$. There are 2^m possible binary
 905 classifications for a set of m elements, subsequently denoted by $d = 2^m$.

906 By definition, there exists a natural number m_0 and a set of m vectors

$$\mathcal{Y} := \{\mathbf{y}_1, \dots, \mathbf{y}_m\} \subset \mathbb{R}^{m_0}, \tag{39}$$

that can be shattered by a subset of networks $\tilde{\mathcal{H}} \subset \mathcal{F}_{W,L}$. Since there are no more than d distinct classifiers for \mathcal{Y} , the class $\tilde{\mathcal{H}}$ consists of at most d DNN functions.

Next, we construct a DNN architecture using $m_0 + 1$ smaller DNN classes. One of these classes is $\tilde{\mathcal{H}}$, while the remaining m_0 classes consist of "indicator" networks, as described in (38). These indicator networks ensure that the combined DNN vanishes outside a certain m_0 -dimensional hypercube. To define this hypercube, we use the set \mathcal{Y} from above.

Specifically, we choose numbers $A > \max_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{y}\|_\infty + 1$ and $B > A$, and define the m_0 -dimensional hypercube

$$\Pi := \{\mathbf{y} = (y_1, \dots, y_{m_0}) \mid A \leq y_i \leq B\}.$$

To construct a DNN that vanishes on Π , we define an approximate indicator function for Π , using a DNN with m_0 -dimensional input $\mathbf{y} = (y_1, \dots, y_{m_0})$:

$$I : \mathbb{R}^{m_0} \rightarrow \mathbb{R}, \quad I(\mathbf{y}) := \frac{1}{m_0} \sum_{i=1}^{m_0} \mathbf{1}_{(A,B,0.5)}(y_i),$$

where $\mathbf{1}_{(A,B,0.5)}(y_i)$ is an indicator network that approximates the indicator function for values within (A, B) .

The final DNN is formed by combining functions from $\tilde{\mathcal{H}}$ with the indicator function I . Since DNNs can be summed if they have the same depth, we adjust the depth of I to match the depth of the functions from $\tilde{\mathcal{H}}$ while ensuring that I remains constant on \mathcal{Y} and Π . Specifically, we use the fact that for $I(\mathbf{y}) > 0$, $\sigma(I(\mathbf{y})) = I(\mathbf{y})$ for the ReLU activation function $\sigma(x) = \max\{x, 0\}$ (for any \mathbf{y} from Π or \mathcal{Y}). This means that by composing I with the required number of ReLU functions, we can construct a DNN that satisfies the desired properties. This construction requires at most $L < W$ additional weights.

To complete the proof, we need to show that there are m input functions $F_m := \{f_1, \dots, f_m\} \subset \{f : G^r \rightarrow \mathbb{R}^{m_0}\}$ that can be shattered by GCNNs from $\mathcal{H}_{6W, L+1, r}$. As the set F_m , we choose functions defined by $f_i(e) = \mathbf{y}_i$ and $f_i(g) \in \Pi$ for $g \in G^r$ and $g \neq e$.

By the definition of shattering (see definition in the main article), to prove that F_m is shattered, it is sufficient to show that for any binary classifier $\mathcal{C} : F_m \rightarrow \{0, 1\}$, there exists a corresponding function in $\text{sign}(\mathcal{H}_{6W, L+1, r})$ whose values coincide with those of \mathcal{C} on F_m .

Choose $\tilde{h} \in \tilde{\mathcal{H}}$ such that for some $b \in \mathbb{R}$, $\text{sign}(\tilde{h}(\mathbf{y}_i) - b) = \mathcal{C}(f_i)$ for $i = 1, \dots, m$.

Since Π is compact, we define

$$T := \max_{\mathbf{y} \in \Pi} |\tilde{h}(\mathbf{y})|.$$

The final DNN $\tilde{h}_{\mathcal{C}}$ adjusts \tilde{h} such that it vanishes on Π but coincides with $\text{sign}(\tilde{h} - b)$ on \mathcal{Y} ,

$$\tilde{h}_{\mathcal{C}} := \sigma(\tilde{h} - (T - b)I - b),$$

with $\sigma(x) = \max\{x, 0\}$

Thus, for any $f_i \in F_m$,

$$\text{sign}\left(\sum_{j=1}^r \tilde{h}_{\mathcal{C}}(f_i(g_j))\right) = \text{sign}(\tilde{h}(\mathbf{y}_i) - b) = \mathcal{C}(f_i).$$

By Lemma 8, we can define a GCNN $h_{\mathcal{C}}$ such that $h_{\mathcal{C}}(f) = \sum_{s=1}^r \tilde{h}_{\mathcal{C}}(f(g_s))$ for any $f \in F_m$. This implies that $\text{sign}(h_{\mathcal{C}}(f_i) - b) = \mathcal{C}(f_i)$ for any $f_i \in F_m$.

As the number of weights in $\tilde{h}_{\mathcal{C}}$ is $W + L + 4m_0 < 6W$, this shows that our GCNN is in the class $\mathcal{H}_{6W, L+1, r}$, completing the proof of the lemma. \square

Corollary 4. [Corollary 2] In the setting of Lemma 2, if the number of layers $L > 3$ and $L \leq W^{0.99}$, then there exists a constant c such that

$$\text{VC}(\mathcal{H}_{W, L, r}) \geq c \cdot \text{VC}(\mathcal{F}_{W, L}).$$

943 *Proof.* From Equation (2) in [3], we know that for the class of fully connected neural networks
 944 $\mathcal{F}_{W,L}$ with L layers and at most W overall parameters, there exist constants c_0 and C_0 such that

$$c_0 \cdot WL \log \left(\frac{W}{L} \right) \leq \text{VC}(\mathcal{F}_{W,L}) \leq C_0 \cdot WL \log W. \quad (40)$$

945 Moreover, by Lemma 9, we have

$$\text{VC}(\mathcal{H}_{6W',L'+1,r}) \geq \text{VC}(\mathcal{F}_{W',L'}).$$

946 By choosing $W = 6W'$ and $L = L' + 1$, this shows that

$$\text{VC}(\mathcal{H}_{W,L,r}) \geq \text{VC}(\mathcal{F}_{\lfloor \frac{1}{6}W \rfloor, L-1}).$$

947 To obtain the statement in the lemma, we combine this bound with the left inequality in (40), leading
 948 to

$$\text{VC}(\mathcal{H}_{W,L,r}) \geq \text{VC}(\mathcal{F}_{\lfloor \frac{1}{6}W \rfloor, L-1}) \geq c_0 \cdot \left(\frac{1}{6}W - 1 \right) (L-1) \log \left(\frac{\frac{1}{6}W - 1}{L-1} \right).$$

949 For some constant $c_1 > 0$, the right-hand side of this inequality is bounded from below by

$$c_1 \cdot WL \log W.$$

950 By using the right inequality in (40), this can be further bounded,

$$c_1 \cdot WL \log W \geq c \cdot \text{VC}(\mathcal{F}_{W,L}),$$

951 showing the assertion. \square

952 Next, we provide the proof for the second part of Theorem 2, which states that for some universal
 953 constant $c > 0$, the VC dimension $\text{VC}(\mathcal{H}_{W,L,r})$ is bounded by $c \cdot W \log_2(r)$. As mentioned in the
 954 main article, the first step of the proof is Lemma 3.

955 **Lemma 10.** [Lemma 3] Let $\mathcal{H}_{4,L,r}$ be the class of GCNNs defined in (34). Then

$$\text{VC}(\mathcal{H}_{4,L,r}) \geq \lfloor \log_2 r \rfloor.$$

956 Moreover, for any two numbers $A < B$, there exists a finite subclass of GCNNs $\mathcal{H} \subset \mathcal{H}_{4,L,r}$ that
 957 shatters a set of $\lfloor \log_2 r \rfloor$ input functions

$$F_m := \{f_i : G^r \rightarrow [A, B] \mid i = 1, \dots, \lfloor \log_2 r \rfloor\},$$

958 and outputs zero for any input function $f : G^r \rightarrow \mathbb{R} \setminus [A, B]$.

959 *Proof.* To simplify the notation, let $m := \lfloor \log_2 r \rfloor$. It will be enough to show that a subclass of
 960 GCNNs $\mathcal{H} \subset \mathcal{H}_{4,L,r}$ shatters the set of m input functions as this immediately implies that

$$\text{VC}(\mathcal{H}_{4,L,r}) \geq \lfloor \log_2 r \rfloor.$$

961 The proof involves selecting $d := 2^m$ distinct points in the interval $[A, B]$ and defining "indicator"
 962 neural networks of the form (38) that output 1 at exactly one of these points. By adjusting the
 963 parameters of these networks, we can control the intervals of our indicator networks and ensure that
 964 each network outputs 1 at the desired point.

965 Specifically, define $\delta := \frac{B-A}{2(d+2)}$ and select the d points

$$\mathcal{Y} := \{y_i := A + i\delta \mid i = 1, \dots, d\}.$$

966 The input functions F_m are now chosen from $\{f : G^r \rightarrow \mathcal{Y} \cup \{B - \delta\}\}$.

967 There are $d = 2^m$ different binary classifiers for the set of m elements. Each binary classifier is
 968 defined by the elements for which it outputs 1, and we can index these classifiers by the subsets of
 969 $\{1, 2, \dots, m\}$, denoted by S_1, \dots, S_d . In our construction, each $y_i \in \mathcal{Y}$ corresponds to the binary
 970 classifier determined by S_i . More formally, the set of m input functions $F_m := \{f_1, \dots, f_m\}$ is
 971 defined by

$$f_j(g_i) := \begin{cases} y_i, & \text{if } j \in S_i, \\ B - \delta, & \text{otherwise.} \end{cases}$$

972 Next, we define the finite subclass in $\mathcal{H}_{4,L,r}$ that shatters F_m and outputs zero for any function
 973 $f : G^r \rightarrow \mathbb{R} \setminus [A, B]$.

974 By the definition of shattering (definition is in the main article), for any binary classifier $\mathcal{C} : F_m \rightarrow$
 975 $\{-1, 1\}$, we need to find a function in $\text{sign}(\mathcal{H}_{4,L,r})$ matching \mathcal{C} on F_m .

976 For any classifier $\mathcal{C} : F_m \rightarrow \{-1, 1\}$ we can find a subset $S \subseteq \{1, \dots, m\}$ such that $\mathcal{C}(f_j) = 1$ if
 977 $j \in S$ and $\mathcal{C}(f_j) = -1$ if $j \in S^c$. There exists an index i^* such that $S = S_{i^*}$. Using Lemma 8, one
 978 can construct a GCNN $h_{i^*} \in \mathcal{H}_{4,L,r}$ that matches \mathcal{C} on F_m . Indeed, for any $f_j \in F_m$,

$$h_{i^*}(f_j) := \sum_{s=1}^r \mathbf{1}_{(y_{i^*} - \frac{\delta}{2}, y_{i^*} + \frac{\delta}{2}, \frac{\delta}{2})}(f_j(g_s)) = \begin{cases} 1, & \text{if } j \in S_{i^*}, \\ 0, & \text{otherwise.} \end{cases} \quad (41)$$

979 Thus, $\text{sign}(h_{i^*}(f) - 0.5) = \mathcal{C}(f)$ for all $f \in F_m$. As an 'indicator' neural network $\mathbf{1}_{(y_{i^*} - \frac{\delta}{2}, y_{i^*} + \frac{\delta}{2}, \frac{\delta}{2})}$
 980 has only 4 parameters and 2 layers, it is in $\mathcal{H}_{4,L,r}$.

981 Moreover, for any $i = 1, \dots, d$ and any $x \in \mathbb{R} \setminus [A, B]$, $\mathbf{1}_{(y_i - \frac{\delta}{2}, y_i + \frac{\delta}{2}, \frac{\delta}{2})}(x) = 0$. Arguing as for (41),
 982 $h_{i^*}(f) = 0$ for any $f : G^r \rightarrow \mathbb{R} \setminus [A, B]$.

983 That means that the class $\mathcal{H} := \{h_1, \dots, h_d\}$ shatters input functions F_m and outputs 0 on the subset
 984 $\{f : G^r \rightarrow \mathbb{R} \setminus [A, B]\}$. This completes the proof. \square

985 **Corollary 5.** [Corollary 3] The VC dimension of the class $\mathcal{H}_{4W,L,r}$, consisting of GCNNs with $4W$
 986 weights, L layers, and resolution r satisfies the inequality

$$\text{VC}(\mathcal{H}_{4W,L,r}) \geq W \lfloor \log_2 r \rfloor.$$

987 *Proof.* To simplify notation, let $m := \lfloor \log_2 r \rfloor$.

988 We prove this corollary by defining W disjoint intervals $[A_1, B_1], \dots, [A_W, B_W]$, where $A_i :=$
 989 $(m+3)i$ and $B_i := (m+2)i$. For different $i \in \{1, \dots, W\}$ the set of input functions $\mathcal{F}_i := \{f :$
 990 $G^r \rightarrow [A_i, B_i]\}$ is disjoint since the values of the intervals do not overlap.

991 By Lemma 10, for each $i = 1, \dots, W$, we can find a class of GCNNs $\mathcal{H}_i \subset \mathcal{H}_{4,L,r}$ that shatters a set
 992 of m input functions $F_{m,i} \subset \mathcal{F}_i$ and outputs 0 on any other set $F_{m,j}$, where $j \neq i$.

993 Next we show that the class of GCNNs $\mathcal{H} := \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus \dots \oplus \mathcal{H}_W \subset \mathcal{H}_{4W,L,r}$ shatters the set
 994 $F_{Wm} := \bigcup_{i=1}^W F_{m,i}$. This will prove the corollary.

995 By the definition of shattering, we need to find for any binary classifier $\mathcal{C} : F_{Wm} \rightarrow \{0, 1\}$, a function
 996 in $\text{sign}(\mathcal{H})$ that matches \mathcal{C} on F_{Wm} .

997 For $i = 1, \dots, W$, let $\mathcal{C}_i := \mathcal{C}|_{F_{m,i}}$ be the restriction of \mathcal{C} to $F_{m,i}$. As \mathcal{H}_i shatters $F_{m,i}$, we can
 998 choose a GCNN $h_{\mathcal{C}_i} \in \mathcal{H}_i$ such that its values match those of \mathcal{C}_i on $F_{m,i}$.

999 Next, we show that the values of the GCNN $h_{\mathcal{C}} := \sum_{i=1}^W h_{\mathcal{C}_i}$ match \mathcal{C} on F_{Wm} . Let f be any input
 1000 function from F_{Wm} , say $f \in F_q$. For any $i \neq q$, it holds that $h_{\mathcal{C}_i}(f) = 0$ since $h_{\mathcal{C}_i} \in \mathcal{H}_i$. Thus,

$$\sum_{i=1}^W h_{\mathcal{C}_i}(f) = h_{\mathcal{C}_q}(f).$$

1001 Since $h_{\mathcal{C}_q}(f) = \mathcal{C}(f)$ by the choice of $h_{\mathcal{C}_q}$, it follows that $h_{\mathcal{C}}(f) = \mathcal{C}(f)$ for any $f \in F_{Wm}$.

1002 This shows that the class \mathcal{H} of GCNNs shatters F_{Wm} , proving the corollary. \square