Supplementary material to “Analysis of convolutional neural network image classifiers in a rotationally symmetric model”

The supplement contains additional material concerning the simulation studies from Section 5, results from the literature used in the proof of Lemma 1 and Theorem 1, the proofs of Lemma 3 and Lemma 4, as well as a bound on the covering number.

A. Additional material for Section 5

A.1. Creating the synthetic image data sets

In order to generate a random image with an appropriate label, we use the Python package *Shapely* to theoretically define a continuous image as follows: Firstly, the gray scale value of the background of the image area C_1 is set to 1 and for each of the three squares it is randomly (independently) determined whether a quarter is removed or not. The probability that a quarter is removed from a square is given by $p = 1 - 0.5^{1/3}$, which implies that the class Y of an image is discrete and uniformly distributed on $\{0, 1\}$. Secondly, the area, rotation, and gray scale value of each geometric object are determined. The area is determined for each object (independently) by a uniform distribution on the interval $[0.02, 0.08]$ for complete squares and on the interval $[0.02, 0.06]$ for squares missing a quarter (the second interval is smaller to avoid too large side lengths of these objects). The angle by which an object is rotated is determined (independently) by a uniform distribution on the interval $[0, 2\pi]$. The gray scale values of the three objects are determined by randomly permuting the list $(0, 1/3, 2/3)$ of three gray scale values. Finally, the positions of the objects are determined one after the other as follows: We choose the position of the first object according to a uniform distribution on the restricted image area so that the object is completely within the image area. We repeat the positioning of the second object in the same way until the second object covers only a maximum of five percent of the area of the first object. For the placement of the third object, we use the same method until the third object covers only a maximum of five percent of the area of the first and second object, respectively. We then use the Python package *Pillow* to discretize the continuous image on G_λ.

A.2. Rotation by nearest neighbor interpolation

In this section, we define the rotation function $f_{rot}^{(\alpha)}$, which is used in Section 5 for the network architecture $\mathcal{F}_4$. We use a nearest neighbor interpolation here to implement rotation by arbitrary angles for two reasons: Firstly, a nearest neighbor interpolation can be easily implemented using the *Keras backend* library as a layer of a CNN, so the corresponding classifier can be trained using the *Adam* optimizer. Secondly, our theory could be easily extended to such an estimator, since the nearest neighbor interpolation can be traced back to a self-mapping of G_λ (cf., equation (34) below), which swaps the image positions accordingly, and thus we can obtain a necessary bound for covering number without much effort.

Since we may rotate parts out of the image area by rotating the input image by arbitrary angles, we first introduce a zero padding function $f_z : [0,1]^{G_\lambda} \to [0,1]^{G_{\lambda+2\cdot z}}$ that symmetrically adds $z \in \mathbb{N}_0$ rows and columns of zeros on all four sides of the image. The output of the function f_z is given by

$$(f_z(\mathbf{x}))_{\left(\frac{i-1/2}{\lambda+2\cdot z}-\frac{1}{2},\frac{j-1/2}{\lambda+2\cdot z}-\frac{1}{2}\right)} = \begin{cases} x_{\left(\frac{i-z-1/2}{\lambda}-\frac{1}{2},\frac{j-z-1/2}{\lambda}-\frac{1}{2}\right)} & , \text{ if } z+1 \le i,j \le z+\lambda \\ 0 & , \text{ elsewhere} \end{cases} \tag{32}$$

for $i, j \in \{1, \dots, \lambda + 2 \cdot z\}$. We choose

$$z_\lambda = \left\lceil \frac{\sqrt{2}\cdot\lambda - \lambda}{2} \right\rceil \tag{33}$$

to ensure that a rotated version of the image entirely contains the original image. To rotate the images by a nearest neighbor interpolation, we define the function $g^{(\alpha)} : G_{\lambda'} \to G_{\lambda'}$ that rotates the image positions with a resolution $\lambda' \in \mathbb{N}$ by an angle $\alpha \in [0, 2\pi)$. The output of the function is given by

$$g^{(\alpha)}(\mathbf{v}) = \underset{\mathbf{u}\in G_{\lambda'}}{\arg\min} \|\mathbf{u} - rot^{(\alpha)}(\mathbf{v})\|_2 \quad (\mathbf{v} \in G_{\lambda'}), \tag{34}$$

where we choose the smallest index in case of ties (we use a bijection which maps $G_{\lambda'}$ to $\{1, \dots, \lambda'^2\}$ to obtain a corresponding order on the indices). The rotation function $f_{rot}^{(\alpha)} : [0,1]^{G_\lambda} \to [0,1]^{G_{\lambda+2\cdot z_\lambda}}$ which rotates an image by the angle $\alpha \in [0, 2\pi)$ is then defined by

$$\left(f_{rot}^{(\alpha)}(\mathbf{x})\right)_{\mathbf{u}} = (f_{z_\lambda}(\mathbf{x}))_{g^{(\alpha)}(\mathbf{u})} \quad (\mathbf{x} \in [0,1]^{G_\lambda})$$

for $\mathbf{u} \in G_{\lambda+2\cdot z_\lambda}$.

B. Auxiliary results

In the following section, we present some results from the literature which we have used in the proof of Lemma 1 and Theorem 1. Our first auxiliary result relates the misclassification error of our plug-in estimate to the L_2 error of the corresponding least squares estimates.

Lemma 5 *Define $(g_\lambda(\Phi), Y)$, $(g_\lambda(\Phi_1), Y_1)$, ..., $(g_\lambda(\Phi_n), Y_n)$, and $\mathcal{D}_n$, η, f^* and f_n as in Section 1.1. Then*

$$\begin{aligned} \mathbf{P}\{f_n(g_\lambda(\Phi)) \neq Y\} - \mathbf{P}\{f^*(g_\lambda(\Phi)) \neq Y\} &\le 2 \cdot \int |\eta_n(x) - \eta(x)| \, \mathbf{P}_{g_\lambda(\Phi)}(dx) \\ &\le 2 \cdot \sqrt{\int |\eta_n(x) - \eta(x)|^2 \mathbf{P}_{g_\lambda(\Phi)}(dx)} \end{aligned}$$

holds.

Proof. See Theorem 1.1 in Györfi et al. (2002). $\square$

Our next result bounds the error of the least squares estimate via empirical process theory.

Lemma 6 *Let (X, Y), (X_1, Y_1), ..., (X_n, Y_n) be independent and identically distributed $\mathbb{R}^d \times \mathbb{R}$-valued random variables. Assume that the distribution of (X, Y) satisfies*

$$\mathbf{E}\{\exp(c_{19} \cdot Y^2)\} < \infty$$

for some constant $c_{19} > 0$ and that the regression function $m(\cdot) = \mathbf{E}\{Y|X = \cdot\}$ is bounded in absolute value. Let $\tilde{m}_n$ be the least squares estimate

$$\tilde{m}_n(\cdot) = \arg \min_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{i=1}^{n} |Y_i - f(X_i)|^2$$

based on some function space $\mathcal{F}_n$ consisting of functions $f : \mathbb{R}^d \to \mathbb{R}$ and set $m_n = T_{c_{20} \cdot \log(n)} \tilde{m}_n$ for some constant $c_{20} > 0$. Then m_n satisfies

$$\begin{aligned}
&\mathbf{E} \int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \\
&\leq \frac{c_{21} \cdot (\log(n))^2 \cdot \sup_{x_1^n \in (\mathbb{R}^d)^n} \left(\log\left(\mathcal{N}_1\left(\frac{1}{n \cdot c_{20} \cdot \log(n)}, T_{c_4 \log(n)} \mathcal{F}_n, x_1^n\right)\right) + 1\right)}{n} \\
&\quad + 2 \cdot \inf_{f \in \mathcal{F}_n} \int |f(x) - m(x)|^2 \mathbf{P}_X(dx)
\end{aligned}$$

for $n > 1$ and some constant $c_{21} > 0$, which does not depend on n or the parameters of the estimate.

Proof. This result follows in a straightforward way from the proof of Theorem 1 in Bagirov, Clausen and Kohler (2009). A complete proof can be found in the supplement of Bauer and Kohler (2019). $\square$

The next result is an approximation result for (p, C)–smooth functions by very deep feedforward neural networks.

Lemma 7 *Let $d \in \mathbb{N}$, let $f : \mathbb{R}^d \to \mathbb{R}$ be (p, C)–smooth for some $p = q + s$, $q \in \mathbb{N}_0$ and $s \in (0, 1]$, and $C > 0$. Let $M \in \mathbb{N}$ with $M \geq 2$ sufficiently large, where*

$$M^{2p} \geq c_{22} \cdot \left(\max\left\{ 2, \sup_{\substack{\mathbf{x} \in [-2,2]^d \\ (l_1, \dots, l_d) \in \mathbb{N}^d \\ l_1 + \dots + l_d \leq q}} \left| \frac{\partial^{l_1 + \dots + l_d} f}{\partial^{l_1} x^{(1)} \dots \partial^{l_d} x^{(d)}} (\mathbf{x}) \right| \right\} \right)^{4(q+1)}$$

must hold for some sufficiently large constant $c_{22} \geq 1$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU activation function

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

and let $L, r \in \mathbb{N}$ such that

(i)

$$L \geq 5M^d + \left\lceil \log_4 \left(M^{2p+4\cdot d\cdot(q+1)} \cdot e^{4(q+1)\cdot(M^d-1)} \right) \right\rceil \cdot \lceil \log_2(\max\{d,q\}+2) \rceil + \lceil \log_4(M^{2p}) \rceil$$

(ii)

$$r \geq 132 \cdot 2^d \cdot \lceil e^d \rceil \cdot \binom{d+q}{d} \cdot \max\{q+1, d^2\}$$

hold. Then there exists a feedforward neural network

$$f_{net} \in \mathcal{G}_d(L, \mathbf{k})$$

with $\mathbf{k} = (k_1, \dots, k_L)$ *and* $k_1 = \dots = k_L = r$ *such that*

$$\sup_{\mathbf{x}\in[-2,2]^d} |f(\mathbf{x}) - f_{net}(\mathbf{x})| \leq c_{23} \cdot \left(\max\left\{ 2, \sup_{\substack{\mathbf{x}\in[-2,2]^d \\ (l_1,\dots,l_d)\in\mathbb{N}^d \\ l_1+\dots+l_d\leq q}} \left| \frac{\partial^{l_1+\dots+l_d} f}{\partial^{l_1} x^{(1)} \dots \partial^{l_d} x^{(d)}}(\mathbf{x}) \right| \right\} \right)^{4(q+1)} \cdot M^{-2p}.$$

Proof. See Theorem 2 b) in Kohler and Langer (2021). □

C. Proof of Lemma 3 and Lemma 4

Proof of Lemma 3. Because of inequality (18) it suffices to show that

$$\max_{i\in\{1,\dots,t\}} \max_{\mathbf{u}\in G_\lambda \,:\, \mathbf{u}+I^{(l)}\subseteq G_\lambda} \left| f_{l,1}^{(i)}(\mathbf{x}_{(i,j)+I^{(l)}}) - \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) \right| \leq (C+1)^l \cdot \max_{\substack{i\in\{1,\dots,t\}, j\in\{1,\dots,4^l\}, \\ k\in\{1,\dots,l\}, s\in\{1,\dots,4^{l-k}\}}} \left\{ \|g_{0,j}^{(i)} - \bar{g}_{0,j}^{(i)}\|_{[0,1],\infty}, \|g_{k,s}^{(i)} - \bar{g}_{k,s}^{(i)}\|_{[0,2]^4,\infty} \right\}.$$

This in turn follows from

$$\left| f_{k,s}^{(i)}(\mathbf{x}) - \bar{f}_{k,s}^{(i)}(\mathbf{x}) \right| \leq (C+1)^k \cdot \max_{m\in\{1,\dots,k\}, s\in\{1,\dots,4^{l-m}\}, j\in\{1,\dots,4^l\}} \left\{ \|g_{0,j}^{(i)} - \bar{g}_{0,j}^{(i)}\|_{[0,1],\infty}, \|g_{m,s}^{(i)} - \bar{g}_{m,s}^{(i)}\|_{[0,2]^4,\infty} \right\} \tag{35}$$

for all $\mathbf{x} \in [0,1]^{I^{(k)}}$, $i \in \{1,\dots,t\}$,$k \in \{0,\dots,l\}$ and $s \in \{1,\dots,4^{l-k}\}$, which we show by induction on k.

For $k=0$, $s \in \{1,\dots,4^l\}$ and $i \in \{1,\dots,t\}$ we have

$$\left|f_{0,s}^{(i)}(x) - \bar{f}_{0,s}^{(i)}(x)\right| = \left|g_{0,s}^{(i)}(x) - \bar{g}_{0,s}^{(i)}(x)\right| \leq \left\|g_{0,s}^{(i)} - \bar{g}_{0,s}^{(i)}\right\|_{[0,1],\infty}$$

for all $x \in [0,1]$. Assume that equation (35) holds for some $k \in \{0,\dots,l-1\}$. Because of the definition of $\bar{f}_{k,s}^{(i)}$ we have

$$0 \leq \bar{f}_{k,s}^{(i)}(\mathbf{x}) \leq 2$$

for all $\mathbf{x} \in [0,1]^{I^{(k)}}$, $i \in \{1,\dots,t\}$, $k \in \{0,\dots,l-1\}$ and $s \in \{1,\dots,4^{l-k}\}$. Then, the triangle inequality and the Lipschitz assumption on $g_{k+1,s}^{(i)}|_{[0,2]^2}$ imply

$$\begin{aligned}
&\left|f_{k+1,s}^{(i)}(\mathbf{x}) - \bar{f}_{k+1,s}^{(i)}(\mathbf{x})\right| \\
&\leq \Big|g_{k+1,s}^{(i)}\Big(f_{k,4\cdot(s-1)+1}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+1}^{(i)}+I^{(k)}}), f_{k,4\cdot(s-1)+2}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+2}^{(i)}+I^{(k)}}), \\
&\qquad\qquad f_{k,4\cdot(s-1)+3}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+3}^{(i)}+I^{(k)}}), f_{k,4\cdot s}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot s}^{(i)}+I^{(k)}})\Big) \\
&\quad - g_{k+1,s}^{(i)}\Big(\bar{f}_{k,4\cdot(s-1)+1}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+1}^{(i)}+I^{(k)}}), \bar{f}_{k,4\cdot(s-1)+2}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+2}^{(i)}+I^{(k)}}), \\
&\qquad\qquad \bar{f}_{k,4\cdot(s-1)+3}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+3}^{(i)}+I^{(k)}}), \bar{f}_{k,4\cdot s}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot s}^{(i)}+I^{(k)}})\Big)\Big| \\
&\quad + \Big|g_{k+1,s}^{(i)}\Big(\bar{f}_{k,4\cdot(s-1)+1}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+1}^{(i)}+I^{(k)}}), \bar{f}_{k,4\cdot(s-1)+2}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+2}^{(i)}+I^{(k)}}), \\
&\qquad\qquad \bar{f}_{k,4\cdot(s-1)+3}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+3}^{(i)}+I^{(k)}}), \bar{f}_{k,4\cdot s}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot s}^{(i)}+I^{(k)}})\Big) \\
&\quad - \bar{g}_{k+1,s}^{(i)}\Big(\bar{f}_{k,4\cdot(s-1)+1}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+1}^{(i)}+I^{(k)}}), \bar{f}_{k,4\cdot(s-1)+2}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+2}^{(i)}+I^{(k)}}), \\
&\qquad\qquad \bar{f}_{k,4\cdot(s-1)+3}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+3}^{(i)}+I^{(k)}}), \bar{f}_{k,4\cdot s}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot s}^{(i)}+I^{(k)}})\Big)\Big| \\
&\leq C \cdot \max_{j \in \{1,\dots,4\}} \left|f_{k,4\cdot(s-1)+m}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+j}^{(i)}+I^{(k)}}) - \bar{f}_{k,4\cdot(s-1)+j}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+m}^{(i)}+I^{(k)}})\right| \\
&\quad + \|g_{k+1,s}^{(i)} - \bar{g}_{k+1,s}^{(i)}\|_{[0,2]^4,\infty} \\
&\leq C \cdot (C+1)^k \cdot \max_{m \in \{1,\dots,k\}, s \in \{1,\dots,4^{l-m}\}, j \in \{1,\dots,4^l\}} \left\{\|g_{0,j}^{(i)} - \bar{g}_{0,j}^{(i)}\|_{[0,1],\infty}, \|g_{m,s}^{(i)} - \bar{g}_{m,s}^{(i)}\|_{[0,2]^4,\infty}\right\} \\
&\quad + \|g_{k+1,s}^{(i)} - \bar{g}_{k+1,s}^{(i)}\|_{[0,2]^4,\infty} \\
&\leq (C+1)^{k+1} \cdot \max_{m \in \{1,\dots,k+1\}, s \in \{1,\dots,4^{l-m}\}, j \in \{1,\dots,4^l\}} \left\{\|g_{0,j}^{(i)} - \bar{g}_{0,j}^{(i)}\|_{[0,1],\infty}, \|g_{m,s}^{(i)} - \bar{g}_{m,s}^{(i)}\|_{[0,2]^4,\infty}\right\}
\end{aligned}$$

for all $\mathbf{x} \in [0,1]^{I^{(k+1)}}$, $i \in \{1,\dots,t\}$ and $s \in \{1,\dots,4^{l-(k+1)}\}$. $\square$

In order to prove Lemma 4, we will use the following two auxiliary results.

Lemma 8 *Let $t \in \mathbb{N}$, set $L_{net} = \lceil \log_2 t \rceil$, set $r_{net} = 3 \cdot t$ and let $\mathcal{G}_t(L_{net}, r_{net})$ be defined as in (10). Then there exist $g_{net} \in \mathcal{G}_t(L_{net}, r_{net})$ such that*

$$g_{net}(\mathbf{x}) = \max\{x_1,\dots,x_t\}$$

for all $\mathbf{x} = (x_1, \dots, x_t) \in \mathbb{R}^t$.

Proof. W.l.o.g. assume that $t > 1$. In the proof we will use the network $g_{max} : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$g_{max}(x_1, x_2) = \sigma(x_2 - x_1) + \sigma(x_1) - \sigma(-x_1) \quad (x_1, x_2 \in \mathbb{R})$$

which satisfies

$$g_{max}(x_1, x_2) = \max\{x_2 - x_1, 0\} + \underbrace{\max\{x_1, 0\} - \max\{-x_1, 0\}}_{=x_1} = \max\{x_1, x_2\}$$

for all $x_1, x_2 \in \mathbb{R}$. For $t \in \mathbb{N} \setminus \{1\}$ we set

$$r(t) = 3 \cdot 2^{\lceil \log_2(t) \rceil - 1} \quad \text{and} \quad L(t) = \lceil \log_2 t \rceil$$

and show the assertion by showing the more powerful assertion that for all $t \in \mathbb{N} \setminus \{1\}$ there exists

$$g_{net} \in \mathcal{G}_t(L_{net}, r(t)) \overset{r(t) < r_{net}}{\subset} \mathcal{G}_t(L_{net}, r_{net})$$

such that

$$g_{net}(\mathbf{x}) = \max\{x_1, \dots, x_t\}$$

for all $\mathbf{x} \in \mathbb{R}^t$. We show this by induction on t.

For $t = 2$ the assertion follows by using the network g_{max}. Now let $t > 2$ and assume the assertion holds for all natural numbers less than t and greater than one. Then there exist $g \in \mathcal{G}_{\lceil t/2 \rceil}(L(\lceil t/2 \rceil), r(\lceil t/2 \rceil))$ such that

$$g(\mathbf{x}) = \max\{x_1, \dots, x_{\lceil t/2 \rceil}\}$$

for all $\mathbf{x} \in \mathbb{R}^{\lceil t/2 \rceil}$. We then define $g_{net} \in \mathcal{G}_t(L(\lceil t/2 \rceil) + 1, 2 \cdot r(\lceil t/2 \rceil))$ by

$$g_{net}(\mathbf{x}) = g_{max}(g(x_1, \dots, x_{\lceil t/2 \rceil}), g(x_{\lfloor t/2 \rfloor + 1}, \dots, x_t)) = \max\{x_1, \dots, x_t\}.$$

It is now sufficient to show that

$$L(t) = L(\lceil t/2 \rceil) + 1 \quad \text{and} \quad r(t) = 2 \cdot r(\lceil t/2 \rceil)).$$

Since $2^k < t \le 2^{k+1}$ for some $k \in \mathbb{N}$ we have

$$\lceil \log_2(2 \cdot \lceil t/2 \rceil) \rceil \ge \lceil \log_2(t) \rceil = k + 1 = \left\lceil \log_2 \left(2 \cdot 2^k \right) \right\rceil \ge \lceil \log_2(2 \cdot \lceil t/2 \rceil) \rceil$$

which implies

$$\lceil \log_2(2 \cdot \lceil t/2 \rceil) \rceil = \lceil \log_2(t) \rceil. \tag{36}$$

By using equation (36) we get

$$L(\lceil t/2 \rceil) + 1 = \lceil \log_2 \lceil t/2 \rceil \rceil + 1 = \lceil \log_2(2 \cdot \lceil t/2 \rceil) \rceil = \lceil \log_2 t \rceil = L(t)$$

and

$$\begin{aligned}
2 \cdot r(\lceil t/2 \rceil)) &= 2 \cdot 3 \cdot 2^{\lceil \log_2(\lceil t/2 \rceil) \rceil - 1} \\
&= 3 \cdot 2^{\lceil \log_2(\lceil t/2 \rceil) \rceil + 1 - 1} \\
&= 3 \cdot 2^{\lceil \log_2(2 \cdot \lceil t/2 \rceil) \rceil - 1} \\
&= 3 \cdot 2^{\lceil \log_2(t) \rceil - 1} \\
&= r(t).
\end{aligned}$$

$\square$

The next lemma allows us to compute the standard feedforward neural networks $\sigma \circ g_{net,k,s}^{(i)}$ from Lemma 4 within a convolutional neural network. Since the input dimension of the standard feedforward neural networks is $d = 1$ for $k = 0$ and $d = 4$ for $k \in \{1, \dots, l\}$ we consider the general case $d \in \mathbb{N}$. Lemma 9 is a modified version of Lemma 3 in Kohler, Krzyżak and Walter (2022) and Lemma 2 in Walter (2021). The idea of filter factorizations for 2-D CNNs using enough channels described in Lemma 9 appeared also in He, Li and Xu (2020). The realization of fully connected ReLU networks by 1-D CNNs can be found in Zhou (2020).

Lemma 9 *Let $d \in \mathbb{N}$ and $g_{net} \in \mathcal{G}_d(L_{net}, r_{net})$ for some $L_{net}, r_{net} \in \mathbb{N}$. Let $\sigma(x) = \max\{x, 0\}$ be the ReLU activation function. We assume that there is given a convolutional neural network $f_{CNN} \in \mathcal{F}_{L,\mathbf{k},\mathbf{M},B}^{CNN}$ with $L = r_0 + L_{net} + 1$ convolutional layers and $k_r = t + r_{net}$ channels in the convolutional layer r ($r = 1, \dots, r_0 + L_{net} + 1$) for $t \in \mathbb{N}$ and $r_0 \in \mathbb{N}_0$, and filter sizes $M_1, \dots, M_{r_0+L_{net}+1} \in \mathbb{N}$ with $M_{r_0+1} = \mathbb{1}_{\{k>0\}} \cdot 2^k + 3$ for some $k \in \mathbb{N}_0$. Let*

$$(i_1, j_1), \dots, (i_d, j_d) \in \{-\lfloor 2^{k-1} + 1 \rfloor, \dots, 0, \dots, \lfloor 2^{k-1} + 1 \rfloor\}^2,$$

$s_0 \in \{1, \dots, t\}$ and $s_1, \dots, s_d \in \{1, \dots, k_{r_0}\}$. The convolutional neural network f_{CNN} is given by its weight matrix

$$\mathbf{w} = \left(w_{i',j',s,s'}^{(r)} \right)_{1 \le i',j' \le M_r, s \in \{1,\dots,k_{r-1}\}, s' \in \{1,\dots,k_r\} r \in \{1,\dots,r_0+L_{net}+1\}}, \tag{37}$$

and its bias weights

$$\mathbf{w}_{bias} = \left(w_{s'}^{(r)} \right)_{s' \in \{1,\dots,k_r\}, r \in \{1,\dots,r_0+L_{net}+1\}}. \tag{38}$$

Then we are able to modify the weights (37) and (38)

$$w_{t_1,t_2,s,s'}^{(r)}, w_{s'}^{(r)} \quad (s \in \{1, \dots, t + r_{net}\}) \tag{39}$$

in layers $r \in \{r_0 + 1, \dots, r_0 + L_{net} + 1\}$ and in channels $s' \in \{s_0, t + 1, \dots, t + r_{net}\}$ such that

$$o_{(i',j'),s_0}^{(r_0+L_{net}+1)} = \sigma\Big(g_{net}\Big(o_{(i'+i_1,j'+j_1),s_1}^{(r_0)}, o_{(i'+i_2,j'+j_2),s_2}^{(r_0)}, \dots, o_{(i'+i_d,j'+j_d),s_d}^{(r_0)} \Big)\Big) \tag{40}$$

for all $(i', j') \in \{1, \dots, \lambda\}^2$, where we set $o_{(i',j'),s}^{(r_0)} = 0$ for $(i', j') \notin \{1, \dots, \lambda\}^2$.

Proof. We assume that the standard feedforward neural network g_{net} is given by

$$g_{net}(\mathbf{x}) = \sum_{i=1}^{r_{net}} w_{1,i}^{(L_{net})} g_i^{(L_{net})}(\mathbf{x}) + w_{1,0}^{(L_{net})},$$

where $g_i^{(L_{net})}$ is recursively defined by

$$g_i^{(r)}(\mathbf{x}) = \sigma\left(\sum_{j=1}^{r_{net}} w_{i,j}^{(r-1)} g_j^{(r-1)}(\mathbf{x}) + w_{i,0}^{(r-1)}\right)$$

for $i \in \{1, \dots, r_{net}\}$, $r \in \{2, \dots, L_{net}\}$, and

$$g_i^{(1)}(\mathbf{x}) = \sigma\left(\sum_{j=1}^{d} w_{i,j}^{(0)} x^{(j)} + w_{i,0}^{(0)}\right) \quad (i \in \{1, \dots, r_{net}\}).$$

W.l.o.g. we can assume that $(s_n, i_n, j_n) \neq (s_m, i_m, j_m)$ for distinct $n, m \in \{1, \dots, d\}$ (otherwise one can show the assertion for a accordingly defined $g'_{net} \in \mathcal{G}_{d'}(L_{net}, r_{net})$ with $d' < d$). Since $M_{r_0+1} = 2 \cdot \lfloor 2^{k-1} \rfloor + 3$ and $\lceil M_{r_0+1}/2 \rceil = \lfloor 2^{k-1} \rfloor + 2$ we have

$$\begin{aligned}
&o_{(i',j'),t+i}^{(r_0+1)} \\
&= \sigma\left(\sum_{s=1}^{k_{r_0}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_{r_0+1}\} \\ i'+t_1-\lceil M_{r_0+1}/2 \rceil \in \{1,\dots,\lambda\} \\ j'+t_2-\lceil M_{r_0+1}/2 \rceil \in \{1,\dots,\lambda\}}} w_{t_1,t_2,s,t+i}^{(r_0+1)} \cdot o_{(i'+t_1-\lceil M_{r_0+1}/2 \rceil, j'+t_2-\lceil M_{r_0+1}/2 \rceil),s}^{(r_0)} + w_{t+i}^{(r_0+1)}\right) \\
&= \sigma\left(\sum_{s=1}^{k_{r_0}} \sum_{\substack{t_1,t_2 \in \{-\lfloor 2^{k-1}+1 \rfloor,\dots,\lfloor 2^{k-1}+1 \rfloor\} \\ (i'+t_1,j'+t_2) \in \{1,\dots,\lambda\}^2}} w_{\lfloor 2^{k-1} \rfloor+2+t_1, \lfloor 2^{k-1} \rfloor+2+t_2,s,t+i}^{(r_0+1)} \cdot o_{(i'+t_1,j'+t_2),s}^{(r_0)} + w_{t+i}^{(r_0+1)}\right)
\end{aligned} \tag{41}$$

for all $i \in \{1, \dots, r_{net}\}$ and $(i', j') \in \{1, \dots, \lambda\}^2$. We aim to choose the weights in (41) such that

$$\begin{aligned}
o_{(i',j'),t+i}^{(r_0+1)} &= \sigma\left(\sum_{n=1}^{d} w_{i,n}^{(0)} \cdot o_{(i'+i_n,j'+j_n),s_n}^{(r_0)} + w_{i,0}^{(0)}\right) \\
&= g_i^{(1)}\left(o_{(i'+i_1,j'+j_1),s_1}^{(r_0)}, o_{(i'+i_2,j'+j_2),s_2}^{(r_0)}, \dots, o_{(i'+i_d,j'+j_d),s_d}^{(r_0)}\right)
\end{aligned}$$

for all $i \in \{1, \dots, r_{net}\}$ and $(i', j') \in \{1, \dots, \lambda\}^2$. Therefore we choose the only non-zero weights by

$$w_{\lfloor 2^{k-1} \rfloor+2+i_n, \lfloor 2^{k-1} \rfloor+2+j_n,s_n,t+i}^{(r_0+1)} = w_{i,n}^{(0)} \quad \text{and} \quad w_{t+i}^{(r_0+1)} = w_{i,0}^{(0)}$$

for $n \in \{1, \dots, d\}$ and $i \in \{1, \dots, r_{net}\}$ and obtain

$$\begin{aligned} o_{(i',j'),t+i}^{(r_0+1)} &= \sigma\left(\sum_{n=1}^{d} w_{i,n}^{(0)} \cdot o_{(i'+i_n,j'+j_n),s_n}^{(r_0)} + w_{i,0}^{(0)}\right) \\ &= g_i^{(1)}\left(o_{(i'+i_1,j'+j_1),s_1}^{(r_0)}, o_{(i'+i_2,j'+j_2),s_2}^{(r_0)}, \dots, o_{(i'+i_d,j'+j_d),s_d}^{(r_0)}\right) \end{aligned} \tag{42}$$

for all $i \in \{1, \dots, r_{net}\}$ and $(i', j') \in \{1, \dots, \lambda\}^2$. For the following layers we have

$$\begin{aligned} o_{(i',j'),t+i}^{(r_0+r)} &= \sigma\Bigg(\sum_{s=1}^{k_{r_0+r-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_{r_0+r}\} \\ i'+t_1-\lceil M_{r_0+r}/2\rceil \in \{1,\dots,\lambda\} \\ j'+t_2-\lceil M_{r_0+r}/2\rceil \in \{1,\dots,\lambda\}}} \\ &\qquad w_{t_1,t_2,s,t+i}^{(r_0+r)} \cdot o_{(i'+t_1-\lceil M_{r_0+r}/2\rceil, j'+t_2-\lceil M_{r_0+r}/2\rceil),s}^{(r_0+r-1)} + w_{t+i}^{(r_0+r)}\Bigg) \\ &= \sigma\Bigg(\sum_{s=1}^{k_{r_0+r-1}} \sum_{\substack{t_1,t_2 \in \{1-\lceil M_{r_0+r}/2\rceil,\dots,M_{r_0+r}-\lceil M_{r_0+r}/2\rceil\} \\ (i'+t_1,j'+t_2)\in\{1,\dots,\lambda\}^2}} \\ &\qquad w_{\lceil M_{r_0+r}/2\rceil+t_1, \lceil M_{r_0+r}/2\rceil+t_2,s,t+i}^{(r_0+r)} \cdot o_{(i'+t_1,j'+t_2),s}^{(r_0+r-1)} + w_{t+i}^{(r_0+r)}\Bigg) \end{aligned}$$

for $r \in \{2, \dots, L_{net}\}$, $i \in \{1, \dots, r_{net}\}$ and $(i', j') \in \{1, \dots, \lambda\}^2$. Here we aim to choose the weights such that

$$o_{(i',j'),t+i}^{(r_0+r)} = \sigma\left(\sum_{j=1}^{r_{net}} w_{i,j}^{(r-1)} \cdot o_{(i',j'),t+j}^{(r_0+r-1)} + w_{i,0}^{(r-1)}\right) \tag{43}$$

for all $r \in \{2, \dots, L_{net}\}$, $i \in \{1, \dots, r_{net}\}$ and $(i', j') \in \{1, \dots, \lambda\}^2$. Therefore we choose the only nonzero weights by

$$w_{\lceil M_{r_0+r}/2\rceil, \lceil M_{r_0+r}/2\rceil, t+j, t+i}^{(r_0+r)} = w_{i,j}^{(r-1)} \quad \text{and} \quad w_{t+i}^{(r_0+r)} = w_{i,0}^{(r-1)}$$

for $r \in \{2, \dots, L_{net}\}$, $i \in \{1, \dots, r_{net}\}$ and $j \in \{1, \dots, r_{net}\}$ which implies equation (43). In layer $r = r_0 + L_{net} + 1$ we have

$$\begin{aligned} o_{(i',j'),s_0}^{(r_0+L_{net}+1)} = \sigma\Bigg(&\sum_{s=1}^{k_{r-1}} \sum_{\substack{t_1,t_2 \in \{1-\lceil M_{r_0+L_{net}+1}/2\rceil,\dots,M_{r_0+L_{net}+1}-\lceil M_{r_0+L_{net}+1}/2\rceil\} \\ (i'+t_1,j'+t_2)\in\{1,\dots,\lambda\}^2}} \\ &w_{\lceil M_{r_0+L_{net}+1}/2\rceil+t_1, \lceil M_{r_0+L_{net}+1}/2\rceil+t_2,s,s_0}^{(r_0+L_{net}+1)} \cdot o_{(i'+t_1,j'+t_2),s}^{(r_0+L_{net})} + w_{s_0}^{(r_0+L_{net}+1)}\Bigg) \end{aligned}$$

for $(i', j') \in \{1, \dots, \lambda\}^2$ and want to choose the weights such that

$$o_{(i',j'),s_0}^{(r_0+L_{net}+1)} = \sigma\left(\sum_{i=1}^{r_{net}} w_{1,i}^{(L_{net})} \cdot o_{(i',j'),t+i}^{(r_0+L_{net})} + w_{1,0}^{(L_{net})}\right) \tag{44}$$

for all $(i', j') \in \{1, \dots, \lambda\}^2$. For this purpose we choose the only nonzero weights by

$$w_{\lceil M_{r_0+L_{net}+1}/2\rceil, \lceil M_{r_0+L_{net}+1}/2\rceil, t+i, s_0}^{(r_0+L_{net}+1)} = w_{1,i}^{(L_{net})} \quad \text{and} \quad w_{s_0}^{(r_0+L_{net}+1)} = w_{1,0}^{(L_{net})}$$

for $i \in \{1, \dots, r_{net}\}$ which implies equation (44). Combining equations (42), (43) and (44) then yields the assertion. $\square$

Proof of Lemma 4. In the proof we use that for $x \geq 0$ we have

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

which enables us to propagate a nonnegative value computed in a layer of a convolutional neural network in channel s' at position (i', j') to the next convolutional layer by

$$o_{(i',j'),s''}^{(r)} = \sigma\left(o_{(i',j'),s'}^{(r-1)}\right) = o_{(i',j'),s'}^{(r-1)} \tag{45}$$

with corresponding weights in the $r-$th layer in channel s'' which are choosen accordingly from the set $\{0, 1\}$.

Firstly, let $g_{max} \in \mathcal{G}_t(L_t, r_t)$ be the neural netork from Lemma 8 such that

$$\begin{aligned}
\bar{\eta}(\mathbf{x}) &= \max_{\mathbf{u} \in G_\lambda \,:\, \mathbf{u}+I^{(l)} \subseteq G_\lambda} \max_{i \in \{1,\dots,t\}} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) \\
&= \max_{i \in \{1,\dots,t\}} \max_{\mathbf{u} \in G_\lambda \,:\, \mathbf{u}+I^{(l)} \subseteq G_\lambda} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) \\
&= g_{\max}\left(\max_{\mathbf{u} \in G_\lambda \,:\, \mathbf{u}+I^{(l)} \subseteq G_\lambda} \bar{f}_{l,1}^{(1)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}), \dots, \max_{\mathbf{u} \in G_\lambda \,:\, \mathbf{u}+I^{(l)} \subseteq G_\lambda} \bar{f}_{l,1}^{(t)}(\mathbf{x}_{\mathbf{u}+I^{(l)}})\right)
\end{aligned}$$

for all $\mathbf{x} \in [0, 1]^{G_\lambda}$. Because of the definition of the function class $\mathcal{F}_{\boldsymbol{\theta}}^{CNN}$, it is thus sufficient to show that for all $i \in \{1, \dots, t\}$ there exists $f_i \in \mathcal{F}_{L,\mathbf{k},\mathbf{M},B}^{CNN}$ such that

$$f_i(\mathbf{x}) = \max_{\mathbf{u} \in G_\lambda \,:\, \mathbf{u}+I^{(l)} \subseteq G_\lambda} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) \tag{46}$$

for all $\mathbf{x} \in [0, 1]^{G_\lambda}$. Therefore, in the remaining of the proof let $i \in \{1, \dots, t\}$ be fixed. The idea is to successively compute the outputs of the functions

$$\bar{f}_{0,1}^{(i)}, \dots, \bar{f}_{0,4^l}^{(i)}, \dots, \bar{f}_{k,1}^{(i)}, \dots, \bar{f}_{k,4^{l-k}}^{(i)}, \dots, \bar{f}_{l-1,1}^{(i)}, \dots, \bar{f}_{l-1,4}^{(i)}, \bar{f}_{l,1}^{(i)}$$

of the discretized hierarchical model $\bar{f}_{l,1}^{(i)}$ by computing the functions $\{\bar{g}_{k,s}^{(i)}\}$ by repeatedly applying Lemma 9, where for $k = 0$ we apply Lemma 9 with $d = 1$ and for $k = 1, \dots, l$ we use $d = 4$. We store the outputs of the functions $\bar{f}_{k,s}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(k)}})$ by the above idea of equation (45) in corresponding channels, so that we can use the outputs severals times.

For the computation of the maximum in equation (46) we will finally use the global max-pooling layers of our CNN architecture (cf., equation (7)).

A convolutional neural network $f_i \in \mathcal{F}^{CNN}_{L,\mathbf{k},\mathbf{M},B}$ is of the form

$$f_i(\mathbf{x}) = \max \left\{ \sum_{s''=1}^{k_L} w_{s''} \cdot o^{(L)}_{(i',j'),s''} \, : \, (i',j') \in \{1+B, \ldots, \lambda - B\}^2 \right\},$$

with the weight vector

$$\mathbf{w} = \left(w^{(r)}_{i',j',s',s''} \right)_{1 \le i',j' \le M_r, s' \in \{1,\ldots,k_{r-1}\}, s'' \in \{1,\ldots,k_r\}, r \in \{1,\ldots,L\}},$$

bias weights

$$\mathbf{w}_{bias} = \left(w^{(r)}_{s''} \right)_{s'' \in \{1,\ldots,k_r\}, r \in \{1,\ldots,L\}},$$

and the output weights

$$\mathbf{w}_{out} = \left(w_s \right)_{s \in \{1,\ldots,k_L\}}.$$

In the first step we show how to choose the weight vector $\mathbf{w}$ and the bias weights $\mathbf{w}_{bias}$ such that

$$o^{(L)}_{(i',j'),1} = \bar{f}^{(i)}_{l,1}(\mathbf{x}_{\left(\frac{i'-1/2}{\lambda} - \frac{1}{2}, \frac{j'-1/2}{\lambda} - \frac{1}{2}\right) + I^{(l)}}) \tag{47}$$

for all $(i',j') \in \{2^{l-1}+l, \ldots, \lambda - 2^{l-1} - (l-1)\}^2$. For $k = 0, \ldots, l$ we set

$$r(k) = \sum_{m=0}^{k} 4^{l-m} \cdot (L_{net} + 1)$$

and show equation (47) by showing via induction on k that

$$o^{(r(k))}_{(i',j'),s} = \bar{f}^{(i)}_{k,s}\left(\mathbf{x}_{\left(\frac{i'-1/2}{\lambda} - \frac{1}{2}, \frac{j'-1/2}{\lambda} - \frac{1}{2}\right) + I^{(k)}}\right) \tag{48}$$

for all $(i',j') \in \{\lceil 2^{k-1} \rceil + k, \ldots, \lambda - \lceil 2^{k-1} \rceil - (k-1)\}^2$, $k \in \{0, \ldots, l\}$ and $s \in \{1, \ldots, 4^{l-k}\}$.

We start with $k = 0$ and show that

$$\begin{aligned} o^{(r(0))}_{(i',j'),s} &= \sigma\left(g^{(i)}_{net,0,s}\left(x_{\frac{i'-1/2}{\lambda} - \frac{1}{2}, \frac{j'-1/2}{\lambda} - \frac{1}{2}} \right) \right) \\ &= \sigma\left(g^{(i)}_{net,0,s}\left(o^{(0)}_{(i',j'),1} \right) \right) \end{aligned}$$

for all $(i',j') \in \{1, \ldots, \lambda\}^2$ and $s \in \{1, \ldots, 4^l\}$. The idea is to successively use Lemma 9 for the computation for each network

$$\left\{ \sigma\left(g^{(i)}_{net,0,s}\left(o^{(0)}_{(i',j'),1} \right) \right) \right\}_{(i',j') \in \{1,\ldots,\lambda\}^2} \tag{49}$$

for $s \in \{1, \ldots, 4^l\}$ and store the computed values in the corresponding channels

$$1, \ldots, 4^l$$

using equation (45). Before we apply Lemma 9, we choose the weights in channel

$$4^l + 1$$

as in equation (45) such that

$$o^{(r)}_{(i',j'),4^l+1} = o^{(0)}_{(i',j'),1}$$

for all $r \in \{1, \dots, r(0)\}$ and $(i', j') \in \{1, \dots, \lambda\}^2$. Next, let us specify how to use Lemma 9. We first note that

$$M_1, \dots, M_{r(0)} = 3.$$

Now, by using Lemma 9 with parameters $d = 1$,

$$s_1 = \begin{cases} 1 & , \text{ if } s = 1 \\ 4^l + 1 & , \text{ elsewhere} \end{cases}$$

$s_0 = s$, and $r_0 = (s - 1) \cdot (L_{net} + 1)$ we can calculate the values (49) in layers

$$r_0 + 1, \dots, r_0 + L_{net} + 1$$

by choosing corresponding weights in channels

$$s, 5 \cdot 4^{l-1} + 1, \dots, 5 \cdot 4^{l-1} + r_{net}$$

such that we have

$$o^{(s \cdot (L_{net}+1))}_{(i',j'),s} = \sigma\left(g^{(i)}_{net,0,s}\left(o^{(0)}_{(i',j'),1}\right)\right)$$

for all $(i', j') \in \{1, \dots, \lambda\}^2$ and $s \in \{1, \dots, 4^l\}$. Once a value has been computed in layer $s \cdot (L_{net} + 1)$ for $s \in \{1, \dots, 4^l\}$, it will be propagated to the next layer using equation (45) such that we have

$$o^{(r(0))}_{(i',j'),s} = \sigma\left(g^{(i)}_{net,0,s}\left(o^{(0)}_{(i',j'),1}\right)\right)$$

for all $(i', j') \in \{1, \dots, \lambda\}^2$ and $s \in \{1, \dots, 4^l\}$, which imply that equation (48) holds for $k = 0$.

Now assume that property (48) is true for some $k \in \{0, \dots, l - 1\}$ and show that property (48) holds for $k + 1$ by choosing the corresponding weights in layers

$$r(k) + 1, \dots, r(k + 1)$$

such that

$$\begin{aligned} o^{(r(k+1))}_{(i',j'),s} = \sigma\Big(g^{(i)}_{net,k+1,s}\Big(&\bar{f}^{(i)}_{k,4\cdot(s-1)+1}\Big(\mathbf{x}_{\left(\frac{i'-1/2}{\lambda}-\frac{1}{2},\frac{j'-1/2}{\lambda}-\frac{1}{2}\right)+\mathbf{i}_{k,4\cdot(s-1)+1}+I^{(k)}}\Big), \\ &\bar{f}^{(i)}_{k,4\cdot(s-1)+2}\Big(\mathbf{x}_{\left(\frac{i'-1/2}{\lambda}-\frac{1}{2},\frac{j'-1/2}{\lambda}-\frac{1}{2}\right)+\mathbf{i}_{k,4\cdot(s-1)+2}+I^{(k)}}\Big), \\ &\bar{f}^{(i)}_{k,4\cdot(s-1)+3}\Big(\mathbf{x}_{\left(\frac{i'-1/2}{\lambda}-\frac{1}{2},\frac{j'-1/2}{\lambda}-\frac{1}{2}\right)+\mathbf{i}_{k,4\cdot(s-1)+3}+I^{(k)}}\Big), \\ &\bar{f}^{(i)}_{k,4\cdot s}\Big(\mathbf{x}_{\left(\frac{i'-1/2}{\lambda}-\frac{1}{2},\frac{j'-1/2}{\lambda}-\frac{1}{2}\right)+\mathbf{i}_{k,4\cdot s}+I^{(k)}}\Big)\Big)\Big) \end{aligned} \tag{50}$$

for all $\mathbf{x} \in [0,1]^{G_\lambda}$, $(i',j') \in \{2^k + k + 2, \dots, \lambda - 2^k - k - 1\}^2$ and $s \in \{1, \dots, 4^{l-(k+1)}\}$. Since

$$\mathbf{i}_{k,s} \in \left\{ -\frac{\lfloor 2^{k-1} \rfloor + 1}{\lambda}, \dots, 0, \dots, \frac{\lfloor 2^{k-1} \rfloor + 1}{\lambda} \right\}^2$$

for all $s \in \{1, \dots, 4^{l-k}\}$ we have

$$(i',j') + \lambda \cdot \mathbf{i}_{k,s} \in \{\lceil 2^{k-1} \rceil + k, \dots, \lambda - \lceil 2^{k-1} \rceil - (k-1)\}^2$$

for all $(i',j') \in \{2^k + k + 1, \dots, \lambda - 2^k - k\}^2$ and $s \in \{1, \dots, 4^{l-k}\}$. Because of the induction hypothesis equation (50) then is equivalent to

$$\begin{aligned} o_{(i',j'),s}^{(r(k+1))} = \sigma\Big(g_{net,k+1,s}^{(i)}\Big(& o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+1},s}^{(r(k))}, o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+2},s}^{(r(k))}, \\ & o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+3},s}^{(r(k))}, o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot s},s}^{(r(k))}\Big)\Big). \end{aligned}$$

Analogous to the induction base case, the idea is to successively use Lemma 9 for the computation of each network

$$\begin{aligned} \sigma\Big(g_{net,k+1,s}^{(i)}\Big(& o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+1},s}^{(r(k))}, o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+2},s}^{(r(k))}, \\ & o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+3},s}^{(r(k))}, o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot s},s}^{(r(k))}\Big)\Big) \end{aligned} \tag{51}$$

for $s \in \{1, \dots, 4^{l-(k+1)}\}$ and store the computed values in the corresponding channels

$$1, \dots, 4^{l-(k+1)}$$

using equation (45). Before we apply Lemma 9, we choose the weights in channels

$$4^{l-(k+1)} + 1, \dots, 4^{l-(k+1)} + 4^{l-k}$$

such that

$$o_{(i',j'),4^{l-(k+1)}+s}^{(r)} = o_{(i',j'),s}^{(r(k))}$$

for all $r \in \{r(k)+1, \dots, r(k+1)\}$, $(i',j') \in \{1, \dots, \lambda\}^2$ and $s = 1, \dots, 4^{l-k}$ by another application of equation (45). Next, let us specify how to use Lemma 9. We first note that

$$M_{r(k)+1}, \dots, M_{r(k+1)} = 2 \cdot \lfloor 2^{k-1} \rfloor + 3.$$

Now, by using Lemma 9 for $s \in \{1, \dots, 4^{l-(k+1)}\}$ with parameters $d = 4$,

$$s_m = \begin{cases} 4 \cdot (s-1) + m & \text{, if } s = 1 \\ 4^{l-(k+1)} + 4 \cdot (s-1) + m & \text{, elsewhere} \end{cases}$$

for $m = 1, \dots, 4$, $\tilde{s} = s$, and $r_0 = r(k) + (s-1) \cdot (L_{net} + 1)$ we can calculate the values (51) in layers

$$r_0 + 1, \dots, r_0 + L_{net} + 1$$

by choosing corresponding weights in channels

$$s, 5 \cdot 4^{l-1} + 1, \dots, 5 \cdot 4^{l-1} + r_{net}$$

such that we have

$$\begin{aligned} o_{(i',j'),s}^{(r(k)+s\cdot(L_{net}+1))} &= \sigma\Big(g_{net,k+1,s}^{(i)}\Big(o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+1},s}^{(r(k))}, o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+2},s}^{(r(k))}, \\ &\qquad o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+3},s}^{(r(k))}, o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot s},s}^{(r(k))}\Big)\Big) \end{aligned}$$

for all $(i',j') \in \{2^k+k+2,\dots,\lambda-2^k-k-1\}^2$ and $s \in \{1,\dots,4^{l-(k+1)}\}$. Once a value has been saved in layer $r(k)+s\cdot(L_{net}+1)$ for $s \in \{1,\dots,4^{l-(k+1)}\}$, it will be propagated to the next layer using equation (45) such that we have

$$\begin{aligned} o_{(i',j'),s}^{(r(k+1))} &= \sigma\Big(g_{net,k+1,s}^{(i)}\Big(o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+1},s}^{(r(k))}, o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+2},s}^{(r(k))}, \\ &\qquad o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot(s-1)+3},s}^{(r(k))}, o_{(i',j')+\lambda\cdot\mathbf{i}_{k,4\cdot s},s}^{(r(k))}\Big)\Big) \end{aligned}$$

for all $(i',j') \in \{2^k+k+2,\dots,\lambda-2^k-k-1\}^2$ and $s \in \{1,\dots,4^{l-(k+1)}\}$, which concludes the first step.

In the second step we choose the output weights $\mathbf{w}_{out}$ such that (46) holds. Here we simply choose $w_1 = 1$ and $w_s = 0$ for $s \in \{2,\dots,k_L\}$ and together with equation (47) we obtain

$$\begin{aligned} f_i(\mathbf{x}) &= \max\left\{\sum_{s''=1}^{k_L} w_{s''} \cdot o_{(i',j'),s''}^{(L)} \,:\, (i',j') \in \{2^{l-1}+l,\dots,\lambda-2^{l-1}-(l-1)\}^2\right\} \\ &= \max\left\{o_{(i',j'),1}^{(L)} \,:\, (i',j') \in \{2^{l-1}+l,\dots,\lambda-2^{l-1}-(l-1)\}^2\right\} \\ &= \max\left\{\bar{f}_{l,1}^{(i)}(\mathbf{x}_{\left(\frac{i'-1/2}{\lambda}-\frac{1}{2},\frac{j'-1/2}{\lambda}-\frac{1}{2}\right)+I^{(l)}}) \,:\, (i',j') \in \{2^{l-1}+l,\dots,\lambda-2^{l-1}-(l-1)\}^2\right\} \\ &= \max_{\mathbf{u}\in G_\lambda \,:\, \mathbf{u}+I^{(l)}\subseteq G_\lambda} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}), \end{aligned}$$

where we used that

$$\begin{aligned} &\left(\frac{i'-1/2}{\lambda}-\frac{1}{2},\frac{j'-1/2}{\lambda}-\frac{1}{2}\right)+I^{(l)} \\ &= \left\{\frac{i'-2^{l-1}-l+1/2}{\lambda}-\frac{1}{2},\dots,\frac{i'+2^{l-1}+(l-1)+1/2}{\lambda}-\frac{1}{2}\right\} \\ &\quad\times\left\{\frac{j'-2^{l-1}-l+1/2}{\lambda}-\frac{1}{2},\dots,\frac{j'+2^{l-1}+(l-1)+1/2}{\lambda}-\frac{1}{2}\right\}. \end{aligned}$$

□

D. A bound on the covering number

In this Section, we present the following upper bound for the covering number of our convolutional neural network architecture $\mathcal{F}_{\boldsymbol{\theta}}^{CNN}$ defined as in Section 3.

Lemma 10 *Let $n, \lambda \in \mathbb{N}\backslash\{1\}$ and let $\sigma(x) = \max\{x, 0\}$ be the ReLU activation function, define*

$$\mathcal{F} \coloneqq \mathcal{F}_{\boldsymbol{\theta}}^{CNN}$$

with $\boldsymbol{\theta} = (t, L, \mathbf{k}, \mathbf{M}, B, L_{net}, r_{net})$ as in Section 3, and set

$$k_{max} = \max\{k_1, \dots, k_L, t, r_{net}\}, \quad M_{max} = \max\{M_1, \dots, M_L\}.$$

Assume $c_{24} \cdot \log n \geq 2$. Then we have for any $\epsilon \in (0,1)$:

$$\sup_{\mathbf{x}_1^n \in (\mathbb{R}^{G_\lambda})^n} \log\left(\mathcal{N}_1\left(\epsilon, T_{c_{24} \cdot \log n}\mathcal{F}, \mathbf{x}_1^n\right)\right)$$
$$\leq c_{25} \cdot L^2 \cdot \log(L \cdot \lambda) \cdot \log\left(\frac{c_{24} \cdot \log n}{\epsilon}\right)$$

for some constant $c_{25} > 0$ which depends only on L_{net}, k_{max} and M_{max}.

The proof of Lemma 10 is analogous to the proof of Lemma 4 in Kohler, Krzyżak and Walter (2022). For the sake of completeness, we have adapted the proof below to the slight differences in network architecture (in Kohler, Krzyżak and Walter (2022) asymmetric zero padding is used in the convolutional layers and the output bound in (7) is applied one-sided). With the aim of proving Lemma 10, we first have to study the VC dimension of our function class $\mathcal{F}_{\boldsymbol{\theta}}^{CNN}$. For a class of subsets of $\mathbb{R}^d$, the VC dimension is defined as follows:

Definition 4 *Let $\mathcal{A}$ be a class of subsets of $\mathbb{R}^d$ with $\mathcal{A} \neq \emptyset$ and $m \in \mathbb{N}$.*

1. *For $\mathbf{x}_1, ..., \mathbf{x}_m \in \mathbb{R}^d$ we define*

$$s(\mathcal{A}, \{\mathbf{x}_1, ..., \mathbf{x}_m\}) \coloneqq |\{A \cap \{\mathbf{x}_1, ..., \mathbf{x}_m\} \ : \ A \in \mathcal{A}\}|.$$

2. *Then the mth* ***shatter coefficient*** *$S(\mathcal{A}, m)$ of $\mathcal{A}$ is defined by*

$$S(\mathcal{A}, m) \coloneqq \max_{\{\mathbf{x}_1, ..., \mathbf{x}_m\} \subset \mathbb{R}^d} s(\mathcal{A}, \{\mathbf{x}_1, ..., \mathbf{x}_m\}).$$

3. *The* ***VC dimension*** *(Vapnik-Chervonenkis-Dimension) $V_\mathcal{A}$ of $\mathcal{A}$ is defined as*

$$V_\mathcal{A} \coloneqq \sup\{m \in \mathbb{N} \ : \ S(\mathcal{A}, m) = 2^m\}.$$

For a class of real-valued functions, we define the VC dimension as follows:

Definition 5 *Let $\mathcal{H}$ denote a class of functions from $\mathbb{R}^d$ to $\{0, 1\}$ and let $\mathcal{F}$ be a class of real-valued functions.*

1. *For any non-negative integer m, we define the **growth function** of H as*

$$\Pi_{\mathcal{H}}(m) \coloneqq \max_{\mathbf{x}_1,\ldots,\mathbf{x}_m \in \mathbb{R}^d} |\{(h(\mathbf{x}_1),\ldots,h(\mathbf{x}_m)) : h \in H\}|.$$

2. *The **VC dimension** (Vapnik-Chervonenkis-Dimension) of $\mathcal{H}$ we define as*

$$\mathrm{VCdim}(\mathcal{H}) \coloneqq \sup\{m \in \mathbb{N} : \Pi_{\mathcal{H}}(m) = 2^m\}.$$

3. *For $f \in \mathcal{F}$ we denote $\mathrm{sgn}(f) \coloneqq \mathbb{1}_{\{f \geq 0\}}$ and $\mathrm{sgn}(\mathcal{F}) \coloneqq \{\mathrm{sgn}(f) : f \in \mathcal{F}\}$. Then the **VC dimension** of $\mathcal{F}$ is defined as*

$$\mathrm{VCdim}(\mathcal{F}) \coloneqq \mathrm{VCdim}(\mathrm{sgn}(\mathcal{F})).$$

A connection between both definitions is given by the following lemma.

Lemma 11 *Suppose $\mathcal{F}$ is a class of real-valued functions on $\mathbb{R}^d$. Furthermore, we define*

$$\mathcal{F}^+ \coloneqq \{\{(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R} : f(\mathbf{x}) \geq y\} : f \in \mathcal{F}\}$$

and define the class $\mathcal{H}$ of real-valued functions on $\mathbb{R}^d \times \mathbb{R}$ by

$$\mathcal{H} \coloneqq \{h((\mathbf{x}, y)) = f(\mathbf{x}) - y : f \in \mathcal{F}\}.$$

Then, it holds that

$$V_{\mathcal{F}^+} = \mathrm{VCdim}(\mathcal{H}).$$

Proof. See Lemma 8 in Kohler, Krzyżak and Walter (2022). □

In order to bound the VC dimension of our function class, we need the following auxiliary result about the number of possible sign vectors attained by polynomials of bounded degree.

Lemma 12 *Suppose $W \leq m$ and let $f_1, ..., f_m$ be polynomials of degree at most D in W variables. Define*

$$K \coloneqq |\{(\mathrm{sgn}(f_1(\mathbf{a})), \ldots, \mathrm{sgn}(f_m(\mathbf{a}))) : \mathbf{a} \in \mathbb{R}^W\}|.$$

Then we have

$$K \leq 2 \cdot \left(\frac{2 \cdot e \cdot m \cdot D}{W}\right)^W.$$

Proof. See Theorem 8.3 in Anthony and Bartlett (1999). □

To get an upper bound for the VC dimension of our function class $\mathcal{F}_{\boldsymbol{\theta}}^{CNN}$ defined as in Section 3 we will use a modification of Theorem 6 in Bartlett et al. (2019).

Lemma 13 *Let $\sigma(x) = \max\{x, 0\}$ be the ReLU activation function, define*

$$\mathcal{F} := \mathcal{F}_{\boldsymbol{\theta}}^{CNN}$$

with $\boldsymbol{\theta} = (t, L, \mathbf{k}, \mathbf{M}, B, L_{net}, r_{net})$ as in Section 3, and set

$$k_{max} = \max\{k_1, \ldots, k_L, t, r_{net}\}, \quad M_{max} = \max\{M_1, \ldots, M_L\}.$$

Assume $\lambda > 1$. Then, we have

$$V_{\mathcal{F}^+} \leq c_{26} \cdot L^2 \cdot \log_2(L \cdot \lambda)$$

for some constant $c_{26} > 0$ which depends only on L_{net}, k_{max} and M_{max}.

Proof. We want to use Lemma 11 to bound $\mathcal{V}_{\mathcal{F}^+}$ by $\mathrm{VCdim}(\mathcal{H})$, where $\mathcal{H}$ is the class of real-valued functions on $[0,1]^{G_\lambda} \times \mathbb{R}$ defined by

$$\mathcal{H} := \{h((\mathbf{x}, y)) = f(\mathbf{x}) - y : f \in \mathcal{F}\}.$$

Let $h \in \mathcal{H}$. Then h depends on t convolutional neural networks

$$f_1, \ldots, f_t \in \mathcal{F}^{CNN}(L, \mathbf{k}, \mathbf{M}, B)$$

and one standard feedforward neural network $g_{net} \in \mathcal{G}_t(L_{net}, r_{net})$ such that

$$h((\mathbf{x}, y)) = g_{net} \circ (f_1, \ldots, f_t)(\mathbf{x}) - y$$

Each one of the convolutional neural networks $f_1, \ldots, f_t$ depends on a weight matrix

$$\mathbf{w}^{(b)} = \left(w_{i,j,s_1,s_2}^{(b,r)}\right)_{1 \leq i,j \leq M_r, s_1 \in \{1,\ldots,k_{r-1}\}, s_2 \in \{1,\ldots,k_r\}, r \in \{1,\ldots,L\}},$$

the weights

$$\mathbf{w}_{bias}^{(b)} = \left(w_{s_2}^{(b,r)}\right)_{s_2 \in \{1,\ldots,k_r\}, r \in \{1,\ldots,L\}}$$

for the bias in each channel and each convolutional layer, the output weights

$$\mathbf{w}_{out}^{(b)} = (w_s^{(b)})_{s \in \{1,\ldots,k_L\}}$$

for $b \in \{1, \ldots, t\}$. The standard feedforward neural network g_{net} depends on the inner weigths

$$w_{i,j}^{(r-1)}$$

for $r \in \{2, \ldots, L_{net}\}$, $j \in \{0, \ldots, r_{net}\}$ and $i \in \{1, \ldots, r_{net}\}$ and

$$w_{i,j}^{(0)}$$

for $j \in \{0, \ldots, t\}$, $i \in \{1, \ldots, r_{net}\}$ and the outer weights

$$w_i^{(L_{net})}$$

for $i \in \{0, \dots, k_{L_{net}}\}$.

We set

$$(k_0, \dots, k_{L+L_{net}+1}) = (1, k_1, \dots, k_L, t, r_{net}, \dots, r_{net})$$

and count the number of weights used up to layer $r \in \{1, \dots, L\}$ in the convolutional part by

$$W_r \coloneqq t \cdot \left(\sum_{s=1}^{r} M_s^2 \cdot k_s \cdot k_{s-1} + \sum_{s=1}^{r} k_s \right),$$

for $r \in \{1, \dots, L\}$ (where we set $W_0 \coloneqq 0$) and

$$W_{L+1} \coloneqq W_L + t \cdot k_L.$$

We continue in the part of the standard feedforward neural network by counting the weights used up to layer $r \in \{1, \dots, L_{net}\}$ by

$$W_{L+1+r} = W_{L+r} + (k_{L+r} + 1) \cdot k_{L+r+1}$$

and denote the total number of weights by

$$\begin{aligned}
W &= W_{L+L_{net}+2} \\
&= W_{L+L_{net}+1} + k_{L+L_{net}+1} + 1 \\
&\leq L \cdot t \cdot \left(M_{max}^2 \cdot k_{max}^2 + k_{max} \right) + t \cdot k_{max} \\
&\quad + L_{net} \cdot ((k_{max} + 1) \cdot k_{max}) + k_{max} + 1 \\
&\leq L \cdot t \cdot \left(M_{max}^2 \cdot (k_{max} + 1) \cdot k_{max} \right) \\
&\quad + L_{net} \cdot ((k_{max} + 1) \cdot k_{max}) \\
&\quad + 2 \cdot t \cdot (k_{max} + 1) \\
&\leq (L + L_{net} + 2) \cdot t \cdot M_{max}^2 \cdot (k_{max} + 1) \cdot k_{max} \\
&\leq 2 \cdot (L + L_{net} + 2) \cdot t \cdot M_{max}^2 \cdot k_{max}^2.
\end{aligned} \tag{52}$$

We define $I^{(0)} = \emptyset$ and for $r \in \{1, \dots, L + L_{net} + 2\}$ we define the index sets

$$I^{(r)} = \{1, \dots, W_r\}.$$

Furthermore, we define a sequence of vectors containing the weights used up to layer $r \in \{1, \dots, L\}$ in the convolutional part by

$$\begin{aligned}
\mathbf{a}_{I^{(r)}} \coloneqq \Big(&\mathbf{a}_{I^{(r-1)}}, w_{1,1,1,1}^{(1,r)}, \dots, w_{M_r, M_r, k_{r-1}, k_r}^{(1,r)}, w_1^{(1,r)}, \dots, w_{k_r}^{(1,r)}, \\
&\dots, w_{1,1,1,1}^{(t,r)}, \dots, w_{M_r, M_r, k_{r-1}, k_r}^{(t,r)}, w_1^{(t,r)}, \dots, w_{k_r}^{(t,r)} \Big) \in \mathbb{R}^{W_r}
\end{aligned}$$

(where $\mathbf{a}_\emptyset$ denotes the empty vector),

$$\mathbf{a}_{I^{(L+1)}} \coloneqq (\mathbf{a}_{I^{(L)}}, w_1^{(1)}, \dots, w_{k_L}^{(1)}, \dots, w_1^{(t)}, \dots, w_{k_L}^{(t)}) \in \mathbb{R}^{W_{L+1}},$$

and by continuing with the part of the standard feedforward neural network we get for $r \in \{1, \dots, L_{net}\}$

$$\mathbf{a}_{I(r+L+1)} \coloneqq \left(\mathbf{a}_{I(r+L)}, w_{1,0}^{(r-1)}, \dots, w_{k_{r+L+1},k_{r+L}}^{(r-1)}\right) \in \mathbb{R}^{W_{r+L+1}}$$

and

$$\mathbf{a} \coloneqq \left(\mathbf{a}_{I(L+L_{net}+1)}, w_0^{(L_{net})}, \dots, w_{L_{net}}^{(L_{net})}\right) \in \mathbb{R}^W.$$

With this notation we can write

$$\mathcal{H} = \{(\mathbf{x}, y) \mapsto h((\mathbf{x}, y), \mathbf{a}) : \mathbf{a} \in \mathbb{R}^W\}$$

and for $b \in \{1, \dots, t\}$

$$\mathcal{F}^{CNN}(L, \mathbf{k}, \mathbf{M}, B) = \{\mathbf{x} \mapsto f_b(\mathbf{x}, \mathbf{a}) : \mathbf{a} \in \mathbb{R}^W\},$$

where the convolutional networks $f_1, \dots, f_t \in \mathcal{F}^{CNN}(L, \mathbf{k}, \mathbf{M}, B)$, as described above, each depends only on W_{L+1}/t variables of $\mathbf{a}$. To get an upper bound for the VC-dimension of $\mathcal{H}$, we will bound the growth function $\Pi_{\mathrm{sgn}(\mathcal{H})}(m)$. In the following we consider first the case where

$$m \geq W \tag{53}$$

since this will allow us several uses of Lemma 12. To bound the growth function $\Pi_{\mathrm{sgn}(\mathcal{H})}(m)$, we fix the input values

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m) \in [0,1]^{G_\lambda} \times \mathbb{R}$$

and consider $h \in \mathcal{H}$ as a function of the weight vector $\mathbf{a} \in \mathbb{R}^W$ of h

$$\mathbf{a} \mapsto h((\mathbf{x}_k, y_k), \mathbf{a}) = g \circ (f_1, \dots, f_t)(\mathbf{x}_k, \mathbf{a}) - y_k = h_k(\mathbf{a})$$

for any $k \in \{1, \dots, m\}$. Then, an upper bound for

$$K \coloneqq |\{(\mathrm{sgn}(h_1(\mathbf{a})), \dots, \mathrm{sgn}(h_m(\mathbf{a}))) : \mathbf{a} \in \mathbb{R}^W\}|$$

implies an upper bound for the growth function $\Pi_{\mathrm{sgn}(\mathcal{H})}(m)$. For any partition

$$\mathcal{S} = \{S_1, \dots, S_M\}$$

of $\mathbb{R}^W$ it holds that

$$K \leq \sum_{i=1}^{M} |\{(\mathrm{sgn}(h_1(\mathbf{a})), \dots, \mathrm{sgn}(h_m(\mathbf{a})) : \mathbf{a} \in S_i\}|. \tag{54}$$

We will construct a partition $\mathcal{S}$ of $\mathbb{R}^W$ such that within each region $S \in \mathcal{S}$, the functions $h_k(\cdot)$ are all fixed polynomials of bounded degree for $k \in \{1, \dots, m\}$, so that each summand of equation (54) can be bounded via Lemma 12. We do this in two steps.

In the first step we construct a partition $\mathcal{S}^{(1)}$ of $\mathbb{R}^W$ such that within each $S \in \mathcal{S}^{(1)}$ the t convolutional neural networks $f_{1,k}(\mathbf{a}), \dots, f_{t,k}(\mathbf{a})$ are all fixed polynomials with dergee of at most $L+1$ for all $k \in \{1, \dots, m\}$, where we denote

$$f_{b,k}(\mathbf{a}) = f_b(\mathbf{x}_k, \mathbf{a})$$

for $b \in \{1, \dots, t\}$. For $b \in \{1, \dots, t\}$ we have

$$f_{b,k}(\mathbf{a}) = \max\left\{ \sum_{s=1}^{k_L} w_s^{(b)} \cdot o_{(i,j),b,s,\mathbf{x}_k}^{(L)}(\mathbf{a}_{I^{(L)}}) : (i,j) \in \{1+B, \dots, \lambda - B\}^2 \right\},$$

where $o_{(i,j),b,s_2,\mathbf{x}}^{(L)}(\mathbf{a}_{I^{(L)}})$ is recursively defined by

$$o_{(i,j),b,s_2,\mathbf{x}}^{(r)}(\mathbf{a}_{I^{(r)}})$$
$$= \sigma\left(\sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_r\} \\ i+t_1-\lceil M_r/2\rceil \in \{1,\dots,\lambda\} \\ j+t_2-\lceil M_r/2\rceil \in \{1,\dots,\lambda\}}} w_{t_1,t_2,s_1,s_2}^{(b,r)} \cdot o_{(i+t_1-\lceil M_r/2\rceil, j+t_2-\lceil M_r/2\rceil),b,s_1,\mathbf{x}}^{(r-1)}(\mathbf{a}_{I^{(r-1)}}) + w_{s_2}^{(b,r)} \right)$$

for $(i,j) \in \{1, \dots, \lambda\}^2$ and $r \in \{1, \dots, L\}$, and by

$$o_{(i,j),b,1,\mathbf{x}}^{(0)}(\mathbf{a}_{I^{(0)}}) = x_{\left(\frac{i-1/2}{\lambda} - \frac{1}{2}, \frac{j-1/2}{\lambda} - \frac{1}{2}\right)} \quad \text{for } (i,j) \in \{1, \dots, \lambda\}^2.$$

Firstly, we construct a partition $\mathcal{S}_L = \{S_1, \dots, S_M\}$ of $\mathbb{R}^W$ such that within each $S \in \mathcal{S}_L$

$$o_{(i,j),b,s,\mathbf{x}_k}^{(L)}(\mathbf{a}_{I^{(L)}})$$

is a fixed polynomial for all $k \in \{1, \dots, m\}$, $s \in \{1, \dots, k_L\}$, $b \in \{1, \dots, t\}$ and $(i,j) \in D$ with degree of at most L in the W_L variables $\mathbf{a}_{I^{(L)}}$ of $\mathbf{a} \in S$. We construct the partition $\mathcal{S}_L$ iteratively layer by layer, by creating a sequence $\mathcal{S}_0, \dots, \mathcal{S}_L$, where each $\mathcal{S}_r$ is a partition of $\mathbb{R}^W$ with the following properties:

1. We have $|\mathcal{S}_0| = 1$ and, for each $r \in \{1, \dots, L\}$,

$$\frac{|\mathcal{S}_r|}{|\mathcal{S}_{r-1}|} \leq 2\left(\frac{2 \cdot e \cdot t \cdot k_r \cdot \lambda^2 \cdot m \cdot r}{W_r}\right)^{W_r}, \tag{55}$$

2. For each $r \in \{0, \dots, L\}$, and each element $S \in \mathcal{S}_r$, each $(i,j) \in \{1, \dots, \lambda\}^2$, each $s \in \{1, \dots, k_r\}$, each $k \in \{1, \dots, m\}$, and each $b \in \{1, \dots, t\}$ when $\mathbf{a}$ varies in S,

$$o_{(i,j),b,s,\mathbf{x}_k}^{(r)}(\mathbf{a}_{I^{(r)}})$$

is a fixed polynomial function in the W_r variables $\mathbf{a}_{I^{(r)}}$ of $\mathbf{a}$, of total degree no more than r.

We define $\mathcal{S}_0 := \{\mathbb{R}^W\}$. Since

$$o^{(0)}_{(i,j),b,s,\mathbf{x}_k}(\mathbf{a}_{I^{(0)}}) = (x_k)_{\left(\frac{i-1/2}{\lambda}-\frac{1}{2},\frac{j-1/2}{\lambda}-\frac{1}{2}\right)}$$

is a constant polynomial, property 2 above is satisfied for $r = 0$. Now suppose that $\mathcal{S}_0, \dots, \mathcal{S}_{r-1}$ have been defined, and we want to define $\mathcal{S}_r$. For $S \in \mathcal{S}_{r-1}$ let

$$p_{(i,j),b,s_1,\mathbf{x}_k,S}(\mathbf{a}_{I^{(r-1)}})$$

denote the function $o^{(r-1)}_{(i,j),b,s_1,\mathbf{x}_k}(\mathbf{a}_{I^{(r-1)}})$, when $\mathbf{a} \in S$. By induction hypothesis

$$p_{(i,j),b,s_1,\mathbf{x}_k,S}(\mathbf{a}_{I^{(r-1)}})$$

is a polynomial with total degree no more than $r-1$, and depends on the W_{r-1} variables $\mathbf{a}_{I^{(r-1)}}$ of $\mathbf{a}$ for any $b \in \{1,\dots,t\}$, $k \in \{1,\dots,m\}$, $(i,j) \in \{1,\dots,\lambda\}^2$ and $s_1 \in \{1,\dots,k_{r-1}\}$. Hence for any $b \in \{1,\dots,t\}$ $k \in \{1,\dots,m\}$, $(i,j) \in \{1,\dots,\lambda\}^2$ and $s_2 \in \{1,\dots,k_r\}$

$$\sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2\in\{1,\dots,M_r\}\\ i+t_1-\lceil M_r/2\rceil\in\{1,\dots,\lambda\}\\ j+t_2-\lceil M_r/2\rceil\in\{1,\dots,\lambda\}}} w^{(b,r)}_{t_1,t_2,s_1,s_2} \cdot p_{(i+t_1-\lceil M_r/2\rceil,j+t_2-\lceil M_r/2\rceil),b,s_1,\mathbf{x}_k,S}(\mathbf{a}_{I^{(r-1)}}) + w^{(b,r)}_{s_2}$$

is a polynomial in the W_r variables $\mathbf{a}_{I^{(r)}}$ of $\mathbf{a}$ with total degree no more than r. Because of condition (53) we have $t \cdot k_r \cdot m \cdot \lambda^2 \geq W_r$. Hence, by Lemma 12, the collection of polynomials

$$\left\{\sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2\in\{1,\dots,M_r\}\\ i+t_1-\lceil M_r/2\rceil\in\{1,\dots,\lambda\}\\ j+t_2-\lceil M_r/2\rceil\in\{1,\dots,\lambda\}}} w^{(b,r)}_{t_1,t_2,s_1,s_2} \cdot p_{(i+t_1-\lceil M_r/2\rceil,j+t_2-\lceil M_r/2\rceil),b,s_1,\mathbf{x}_k,S}(\mathbf{a}_{I^{(r-1)}}) + w^{(b,r)}_{s_2} : \right.$$
$$\left. b \in \{1,\dots,t\}, k \in \{1,\dots,m\}, (i,j) \in \{1,\dots,\lambda\}^2, s_2 \in \{1,\dots,k_r\}\right\} \tag{56}$$

attains at most

$$\Pi := 2\left(\frac{2 \cdot e \cdot t \cdot k_r \cdot m \cdot \lambda^2 \cdot r}{W_r}\right)^{W_r}$$

distinct sign patterns when $\mathbf{a} \in S$. Therefore, we can partition $S \subset \mathbb{R}^W$ into Π subregions, such that all the polynomials don't change their signs within each subregion. Doing this for all regions $S \in \mathcal{S}_{r-1}$ we get our required partition $\mathcal{S}_r$ by assembling all of these subregions. In particular, property 1 (inequality (55)) is then satisfied.

Fix some $S' \in \mathcal{S}_r$. Notice that, when $\mathbf{a}$ varies in S', all the polynomials in (56) don't change their signs, hence

$$
\begin{aligned}
&o^{(r)}_{(i,j),b,s_2,\mathbf{x}_k}(\mathbf{a}_{I^{(r)}}) \\
&= \sigma\left(\sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2\in\{1,\ldots,M_r\} \\ i+t_1-\lceil M_r/2\rceil\in\{1,\ldots,\lambda\} \\ j+t_2-\lceil M_r/2\rceil\in\{1,\ldots,\lambda\}}} w^{(b,r)}_{t_1,t_2,s_1,s_2}\cdot o^{(r-1)}_{(i+t_1-\lceil M_r/2\rceil,j+t_2-\lceil M_r/2\rceil),b,s_1,\mathbf{x}}(\mathbf{a}_{I^{(r-1)}}) + w^{(b,r)}_{s_2}\right) \\
&= \max\left\{\sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2\in\{1,\ldots,M_r\} \\ i+t_1-\lceil M_r/2\rceil\in\{1,\ldots,\lambda\} \\ j+t_2-\lceil M_r/2\rceil\in\{1,\ldots,\lambda\}}} w^{(b,r)}_{t_1,t_2,s_1,s_2}\cdot o^{(r-1)}_{(i+t_1-\lceil M_r/2\rceil,j+t_2-\lceil M_r/2\rceil),b,s_1,\mathbf{x}}(\mathbf{a}_{I^{(r-1)}})\right. \\
&\qquad\qquad \left. + w^{(b,r)}_{s_2}, 0\right\}
\end{aligned}
$$

is either a polynomial of degree no more than r in the W_r variables $\mathbf{a}_{I^{(r)}}$ of $\mathbf{a}$ or a constant polynomial with value 0 for all $(i,j) \in \{1,\ldots,\lambda\}^2$, $b \in \{1,\ldots,t\}$, $s_2 \in \{1,\ldots,k_r\}$ and $k \in \{1,\ldots,m\}$. Hence, property 2 is also satisfied and we are able to construct our desired partition $\mathcal{S}_L$. Because of inequality (55) of property 1 it holds that

$$
|\mathcal{S}_L| \le \prod_{r=1}^{L} 2\left(\frac{2\cdot e\cdot t\cdot k_r\cdot\lambda^2\cdot m\cdot r}{W_r}\right)^{W_r}.
$$

For any $(i,j) \in \{1,\ldots,\lambda\}^2$, $b \in \{1,\ldots,t\}$ and $k \in \{1,\ldots,m\}$, we define

$$
f_{(i,j),b,\mathbf{x}_k}(\mathbf{a}_{I^{(L+1)}}) \coloneqq \sum_{s_2=1}^{k_L} w^{(b)}_{s_2}\cdot o^{(L)}_{(i,j),b,s_2,\mathbf{x}_k}(\mathbf{a}_{I^{(L)}}).
$$

For any fixed $S \in \mathcal{S}_L$, let $p_{(i,j),b,S,\mathbf{x}_k}(\mathbf{a}_{I^{(L+1)}})$ denote the function $f_{(i,j),b,\mathbf{x}_k}(\mathbf{a}_{I^{(L+1)}})$, when $\mathbf{a} \in S$. By construction of $\mathcal{S}_L$ this is a polynomial of degree no more than $L+1$ in the W_{L+1} variables $\mathbf{a}_{I^{(L+1)}}$ of $\mathbf{a}$. Because of condition (53) we have $t\cdot\lambda^4\cdot m \ge W_{L+1}$. Hence, by Lemma 12, the collection of polynomials

$$
\begin{aligned}
&\Big\{p_{(i_1,j_1),b,S,\mathbf{x}_k}(\mathbf{a}_{I^{(L+1)}}) - p_{(i_2,j_2),b,S,\mathbf{x}_k}(\mathbf{a}_{I^{(L+1)}}) : \\
&\quad (i_1,j_1),(i_2,j_2) \in \{1,\ldots,\lambda\}^2, (i_1,j_1) \ne (i_2,j_2), b \in \{1,\ldots,t\}, k \in \{1,\ldots,m\}\Big\}
\end{aligned}
$$

attains at most

$$
\Delta \coloneqq 2\left(\frac{2\cdot e\cdot t\cdot\lambda^4\cdot m\cdot(L+1)}{W_{L+1}}\right)^{W_{L+1}}
$$

distinct sign patterns when $\mathbf{a} \in S$. Therefore, we can partition $S \subset \mathbb{R}^W$ into Δ subregions, such that all the polynomials don't change their signs within each subregion.

Doing this for all regions $S \in \mathcal{S}_L$ we get our required partition $\mathcal{S}^{(1)}$ by assembling all of these subregions. For the size of our partition $\mathcal{S}^{(1)}$ we get

$$|\mathcal{S}^{(1)}| \leq \prod_{r=1}^{L} 2 \cdot \left(\frac{2 \cdot t \cdot e \cdot k_r \cdot \lambda^2 \cdot m \cdot r}{W_r} \right)^{W_r} \cdot 2 \cdot \left(\frac{2 \cdot e \cdot t \cdot \lambda^4 \cdot m \cdot (L+1)}{W_{L+1}} \right)^{W_{L+1}}.$$

Fix some $S' \in \mathcal{S}^{(1)}$. Notice that, when $\mathbf{a}$ varies in S', all the polynomials

$$\Big\{ p_{(i_1,j_1),b,S,\mathbf{x}_k}(\mathbf{a}_{I(L+1)}) - p_{(i_2,j_2),b,S,\mathbf{x}_k}(\mathbf{a}_{I(L+1)}) : \\ (i_1,j_1),(i_2,j_2) \in \{1,\dots,\lambda\}^2, (i_1,j_1) \neq (i_2,j_2), b \in \{1,\dots,t\}, k \in \{1,\dots,m\} \Big\}$$

don't change their signs. Hence, there is a permutation $\pi^{(b,k)}$ of the set

$$\{1+B,\dots,\lambda-B\}^2$$

for any $b \in \{1,\dots,t\}$ and $k \in \{1,\dots,m\}$ such that

$$f_{\pi^{(b,k)}((1+B,1+B)),b,\mathbf{x}_k}(\mathbf{a}_{I(L+1)}) \geq \dots \geq f_{\pi^{(b,k)}((\lambda-B,\lambda-B)),b,\mathbf{x}_k}(\mathbf{a}_{I(L+1)})$$

for $\mathbf{a} \in S'$ and any $k \in \{1,\dots,m\}$ and $b \in \{1,\dots,t\}$. Therefore, it holds that

$$f_{b,k}(\mathbf{a}) = \max\Big\{ f_{(1+B,1+B),b,\mathbf{x}_k}(\mathbf{a}_{I(L+1)}), \dots, f_{(\lambda-B,\lambda-B),b,\mathbf{x}_k}(\mathbf{a}_{I(L+1)}) \Big\} \\ = f_{\pi^{(b,k)}((1+B,1+B)),b,\mathbf{x}_k}(\mathbf{a}_{I(L+1)}),$$

for $\mathbf{a} \in S'$. Since $f_{\pi^{(b,k)}((1+B,1+B)),b,\mathbf{x}_k}(\mathbf{a}_{I(L+1)})$ is a polynomial within S', also $f_{b,k}(\mathbf{a})$ is a polynomial within S' with degree no more than $L+1$ and in the W_{L+1} variables $\mathbf{a}_{I(L+1)}$ of $\mathbf{a} \in \mathbb{R}^W$.

In the second step we construct the partition $\mathcal{S}$ starting from partition $\mathcal{S}^{(1)}$ such that within each region $S \in \mathcal{S}$ the functions $h_k(\cdot)$ are all fixed polynomials of degree of at most $L + L_{net} + 2$ for $k \in \{1,\dots,m\}$. We have

$$h_k(\mathbf{a}) = \sum_{i=1}^{k_{L+L_{net}+1}} w_i^{(L_{net})} \cdot g_{i,k}^{(L_{net})}(\mathbf{a}_{I(L+L_{net}+1)}) + w_0^{(L_{net})} - y_k$$

where the $g_{i,k}^{(L_{net})}$ are recursively defined by

$$g_{i,k}^{(r)}(\mathbf{a}_{I(L+r+1)}) = \sigma\left(\sum_{j=1}^{k_{L+r}} w_{i,j}^{(r-1)} g_{j,k}^{(r-1)}(\mathbf{a}_{I(L+r)}) \right)$$

for $r \in \{1,\dots,L_{net}\}$ and

$$g_{i,k}^{(0)}(\mathbf{a}_{I(L+1)}) = f_{i,k}(\mathbf{a})$$

for $i \in \{1,\dots,k_{L+1}\}$ ($k_{L+1} = t$). As above we construct the partition $\mathcal{S}$ iteratively layer by layer, by creating a sequence $\mathcal{S}_0,\dots,\mathcal{S}_{L_{net}}$, where each $\mathcal{S}_r$ is a partition of $\mathbb{R}^W$ with the following porperties:

1. We set $\mathcal{S}_0 = \mathcal{S}^{(1)}$ and, for each $r \in \{1, \dots, L_{net}\}$,

$$\frac{|\mathcal{S}_r|}{|\mathcal{S}_{r-1}|} \leq 2\left(\frac{2 \cdot e \cdot k_{L+r+1} \cdot m \cdot (L+r+1)}{W_{L+r+1}}\right)^{W_{L+r+1}}, \tag{57}$$

2. For each $r \in \{0, \dots, L_{net}\}$, and each element $S \in \mathcal{S}_r$, each $i \in \{1, \dots, k_{L+r+1}\}$, and each $k \in \{1, \dots, m\}$ when $\mathbf{a}$ varies in S,

$$g_{i,k}^{(r)}(\mathbf{a}_{I(L+r+1)})$$

is a fixed polynomial function in the W_{L+r+1} variables $\mathbf{a}_{I(L+r+1)}$ of $\mathbf{a}$, of total degree no more than $L+r+1$.

As we have already shown in step 1, property 2 above is satisfied for $r = 0$. Now suppose that $\mathcal{S}_0, \dots, \mathcal{S}_{r-1}$ have been defined, and we want to define $\mathcal{S}_r$. For $S \in \mathcal{S}_{r-1}$ and $j \in \{1, \dots, k_{L+r}\}$ let $p_{j,k,S}(\mathbf{a}_{I(L+r)})$ denote the function $g_{j,k}^{(r-1)}(\mathbf{a}_{I(L+r)})$, when $\mathbf{a} \in S$. By induction hypothesis $p_{j,k,S}(\mathbf{a}_{I(L+r)})$ is a polynomial with total degree no more than $L+r$, and depends on the W_{L+r} variables $\mathbf{a}_{I(L+r)}$ of $\mathbf{a}$. Hence for any $k \in \{1, \dots, m\}$ and $i \in \{1, \dots, k_{L+r+1}\}$

$$\sum_{j=1}^{k_{L+r}} w_{(i,j)}^{(r-1)} \cdot p_{j,k,S}(\mathbf{a}_{I(L+r)}) + w_{i,0}^{(r-1)}$$

is a polynomial in the W_{L+r+1} variables $\mathbf{a}_{I(L+r+1)}$ variables of $\mathbf{a}$ with total degree no more than $L+r+1$. Because of condition (53) we have $k_{L+r+1} \cdot m \geq W_{L+r+1}$. Hence, by Lemma 12, the collection of polynomials

$$\left\{\sum_{j=1}^{k_{L+r}} w_{(i,j)}^{(r-1)} \cdot p_{j,k,S}(\mathbf{a}_{I(L+r)}) + w_{i,0}^{(r-1)} : k \in \{1, \dots, m\}, i \in \{1, \dots, k_{L+r+1}\}\right\}$$

attains at most

$$\Pi \coloneqq 2\left(\frac{2 \cdot e \cdot k_{L+r+1} \cdot m \cdot (L+r+1)}{W_{L+r+1}}\right)^{W_{L+r+1}}$$

distinct sign patterns when $\mathbf{a} \in S$. Therefore, we can partition $S \subset \mathbb{R}^W$ into Π subregions, such that all the polynomials don't change their signs within each subregion. Doing this for all regions $S \in \mathcal{S}_{r-1}$ we get our required partition $\mathcal{S}_r$ by assembling all of these subregions. In particular property 1 is then satisfied. In order to see that condition 2 is also satisfied, we can proceed analogously to step 1. Hence, when $\mathbf{a}$ varies in $S \in \mathcal{S}$ the function

$$h_k(\mathbf{a}) = \sum_{i=1}^{k_{L+L_{net}+1}} w_i^{(L)} \cdot g_{i,k}^{(L_{net})}(\mathbf{a}_{I(L+L_{net}+1)}) + w_0^{(L)} - y_k$$

is a polynomial of degree no more than $L + L_{net} + 2$ in the W variables of $\mathbf{a} \in \mathbb{R}^W$ for any $k \in \{1, \dots, m\}$. For the size of our partition $\mathcal{S}$ we get

$$
\begin{aligned}
|\mathcal{S}| \leq \prod_{r=1}^{L} 2 \cdot \left(\frac{2 \cdot e \cdot t \cdot k_r \cdot \lambda^2 \cdot m \cdot r}{W_r}\right)^{W_r} \cdot 2 \cdot \left(\frac{2 \cdot e \cdot \lambda^4 \cdot m \cdot (L+1)}{W_{L+1}}\right)^{W_{L+1}} \\
\cdot \prod_{r=1}^{L_{net}} 2 \cdot \left(\frac{2 \cdot e \cdot k_{L+r+1} \cdot m \cdot (L+r+1)}{W_{L+r+1}}\right)^{W_{L+r+1}} \\
\leq \prod_{r=1}^{L+L_{net}+1} 2 \cdot \left(\frac{2 \cdot e \cdot t \cdot k_r \cdot \lambda^4 \cdot m \cdot r}{W_r}\right)^{W_r}
\end{aligned}
$$

By condition (53) and another application of Lemma 12 it holds for any $S' \in \mathcal{S}$ that

$$
\begin{aligned}
&|\{(\mathrm{sgn}(h_1(\mathbf{a})), \dots, \mathrm{sgn}(h_m(\mathbf{a}))) : \mathbf{a} \in S'\}| \\
&\leq 2 \cdot \left(\frac{2 \cdot e \cdot m \cdot (L + L_{net} + 2)}{W}\right)^{W}.
\end{aligned}
$$

Now we are able to bound K via equation (54) and because K is an upper bound for the growth function we set $k_{L+L_{net}+2} = 1$ and get

$$
\begin{aligned}
\Pi_{\mathrm{sgn}(\mathcal{H})}(m) \leq & \prod_{r=1}^{L+L_{net}+2} 2 \cdot \left(\frac{2 \cdot e \cdot t \cdot k_r \cdot \lambda^4 \cdot r \cdot m}{W_r}\right)^{W_r} \\
\leq & 2^{L+L_{net}+2} \cdot \left(\frac{\sum_{r=1}^{L+L_{net}+2} 2 \cdot e \cdot t \cdot k_r \cdot \lambda^4 \cdot r \cdot m}{\sum_{r=1}^{L+L_{net}+2} W_r}\right)^{\sum_{r=1}^{L+L_{net}+2} W_r} \\
= & 2^{L+L_{net}+2} \cdot \left(\frac{R \cdot m}{\sum_{r=1}^{L+L_{net}+2} W_r}\right)^{\sum_{r=1}^{L+L_{net}+2} W_r}, \qquad (58)
\end{aligned}
$$

with $R \coloneqq 2 \cdot e \cdot t \cdot \lambda^4 \cdot \sum_{r=1}^{L+L_{net}+2} k_r \cdot r$. In the second row we used the weighted AM-GM inequality (see, e.g., Cvetkovski (2012), pp. 74-75). Without loss of generality, we can assume that $\mathrm{VCdim}(\mathcal{H}) \geq \sum_{r=1}^{L+L_{net}+2} W_r$ because in the case $\mathrm{VCdim}(\mathcal{H}) < \sum_{r=1}^{L+L_{net}+2} W_r$ we have

$$
\begin{aligned}
\mathrm{VCdim}(\mathcal{H}) &< (L + L_{net} + 2) \cdot W \\
&\overset{(52)}{\leq} 2 \cdot (L + L_{net} + 2)^2 \cdot t \cdot M_{max}^2 \cdot k_{max}^2 \\
&\leq c_{26} \cdot L^2
\end{aligned}
$$

for some constant $c_{26} > 0$ which only depends on L_{net}, M_{max} and k_{max} and get the assertion by Lemma 11. Hence we get by the definition of the VC–dimension and inequality (58) (which only holds for $m \geq W$)

$$
2^{\mathrm{VCdim}(\mathcal{H})} = \Pi_{\mathrm{sgn}(\mathcal{H})}(\mathrm{VCdim}(\mathcal{H})) \leq 2^{L+L_{net}+2} \cdot \left(\frac{R \cdot \mathrm{VCdim}(\mathcal{H})}{\sum_{r=1}^{L+L_{net}+2} W_r}\right)^{\sum_{r=1}^{L+L_{net}+2} W_r}.
$$

Since

$$R \geq 2 \cdot e \cdot t \cdot \lambda^4 \cdot \sum_{r=1}^{1+1+2} r \geq 2 \cdot e \cdot t \cdot \lambda^4 \cdot 10 \geq 16$$

Lemma 14 below (with parameters R, $m = \mathrm{VCdim}(\mathcal{H})$, $w = \sum_{r=1}^{L+L_{net}+2} W_r$ and $L' = L + L_{net} + 2$) implies that

$$\begin{aligned}
\mathrm{VCdim}(\mathcal{H}) &\leq (L + L_{net} + 2) + \left(\sum_{r=1}^{L+L_{net}+2} W_r \right) \cdot \log_2(2 \cdot R \cdot \log_2(R)) \\
&\leq (L + L_{net} + 2) + (L + L_{net} + 2) \cdot W \\
&\qquad \cdot \log_2(2 \cdot (2 \cdot e \cdot t \cdot \lambda^4 \cdot (L + L_{net} + 2) \cdot k_{max})^2) \\
&\leq 2 \cdot (L + L_{net} + 2) \cdot W \cdot \log_2 \left((2 \cdot e \cdot t \cdot (L + L_{net} + 2) \cdot k_{max} \cdot \lambda)^8 \right) \\
&\overset{(52)}{\leq} 32 \cdot t \cdot (L + L_{net} + 2)^2 \cdot k_{\max}^2 \cdot M_{\max}^2 \\
&\qquad \cdot \log_2 \left(2 \cdot e \cdot t \cdot (L + L_{net} + 2) \cdot k_{max} \cdot \lambda\right) \\
&\leq c_{26} \cdot L^2 \cdot \log_2(L \cdot \lambda),
\end{aligned}$$

for some constant $c_{26} > 0$ which only depends on L_{net}, $k_{\max}$ and $M_{\max}$. In the third row we used equation (52) for the total number of weights W. Now we make use of Lemma 11 and finally get

$$V_{\mathcal{F}^+} \leq c_{26} \cdot L^2 \cdot \log_2(L \cdot \lambda).$$

□

Lemma 14 *Suppose that* $2^m \leq 2^{L'} \cdot (m \cdot R/w)^w$ *for some* $R \geq 16$ *and* $m \geq w \geq L' \geq 0$. *Then,*

$$m \leq L' + w \cdot \log_2(2 \cdot R \cdot \log_2(R)).$$

Proof. See Lemma 16 in Bartlett et al. (2019). □

Proof of Lemma 10. Using Lemma 13 and

$$V_{T_{c_4 \cdot \log n}\mathcal{F}^+} \leq V_{\mathcal{F}^+},$$

we can conclude from this together with Lemma 9.2 and Theorem 9.4 in Györfi et al. (2002)

$$\begin{aligned}
&\mathcal{N}_1\left(\epsilon, T_{c_{24} \cdot \log n}\mathcal{F}, \mathbf{x}_1^n\right) \\
&\leq 3 \cdot \left(\frac{4e \cdot c_{24} \cdot \log n}{\epsilon} \cdot \log \frac{6e \cdot c_{24} \cdot \log n}{\epsilon} \right)^{V_{T_{c_{24} \cdot \log n}\mathcal{F}^+}} \\
&\leq 3 \cdot \left(\frac{6e \cdot c_{24} \cdot \log n}{\epsilon} \right)^{2 \cdot c_{25} \cdot L^2 \cdot \log(L \cdot \lambda^2)}.
\end{aligned}$$

This completes the proof of Lemma 10.

E. Proof of inequality (12)

The aim of this section is to prove equation (12) of Example 1 (see Section 4). For this we first define the bilinear interpolation $\phi_{\mathbf{x}} : C_1 \to [0,1]$ for some $\mathbf{x} \in [0,1]^{H_{\lambda_{max}}}$:

Let $\mathbf{x} \in [0,1]^{H_{\lambda_{max}}}$ be fixed. For $\mathbf{v} := (v_1, v_2) \in C_1$ let

$$(a_1^{(\mathbf{v})}, b_1^{(\mathbf{v})}), (a_2^{(\mathbf{v})}, b_1^{(\mathbf{v})}), (a_1^{(\mathbf{v})}, b_2^{(\mathbf{v})}), (a_2^{(\mathbf{v})}, b_2^{(\mathbf{v})}) \in H_{\lambda_{max}} \tag{59}$$

be grid points such that $\mathbf{v} \in [a_1^{(\mathbf{v})}, b_1^{(\mathbf{v})}] \times [a_2^{(\mathbf{v})}, b_2^{(\mathbf{v})}]$ and define the coefficients

$$k_0^{(\mathbf{v})}, k_1^{(\mathbf{v})}, k_2^{(\mathbf{v})}, k_3^{(\mathbf{v})} \in \mathbb{R}$$

by

$$\begin{pmatrix} k_0^{(\mathbf{v})} \\ k_1^{(\mathbf{v})} \\ k_2^{(\mathbf{v})} \\ k_3^{(\mathbf{v})} \end{pmatrix} = (\lambda_{max} - 1)^2 \cdot \begin{pmatrix} a_2^{(\mathbf{v})} \cdot b_2^{(\mathbf{v})} & -a_2^{(\mathbf{v})} \cdot b_1^{(\mathbf{v})} & -a_1^{(\mathbf{v})} \cdot b_2^{(\mathbf{v})} & a_1^{(\mathbf{v})} \cdot b_1^{(\mathbf{v})} \\ -b_2^{(\mathbf{v})} & b_1^{(\mathbf{v})} & b_2^{(\mathbf{v})} & -b_1^{(\mathbf{v})} \\ -a_2^{(\mathbf{v})} & a_2^{(\mathbf{v})} & a_1^{(\mathbf{v})} & -a_1^{(\mathbf{v})} \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_{(a_1^{(\mathbf{v})}, b_1^{(\mathbf{v})})} \\ x_{(a_1^{(\mathbf{v})}, b_2^{(\mathbf{v})})} \\ x_{(a_2^{(\mathbf{v})}, b_1^{(\mathbf{v})})} \\ x_{(a_2^{(\mathbf{v})}, b_2^{(\mathbf{v})})} \end{pmatrix},$$

where we note that

$$\max\left\{|k_1^{(\mathbf{v})}|, |k_2^{(\mathbf{v})}|, |k_3^{(\mathbf{v})}|\right\} \leq 2 \cdot (\lambda_{max} - 1)^2. \tag{60}$$

The bilinear interpolation $\phi_{\mathbf{x}}$ is then defined as

$$\phi_{\mathbf{x}}(\mathbf{v}) = k_0^{(\mathbf{v})} + k_1^{(\mathbf{v})} \cdot v_1 + k_2^{(\mathbf{v})} \cdot v_2 + k_3^{(\mathbf{v})} \cdot v_1 \cdot v_2 \quad (\mathbf{v} \in C_1)$$

(see Figure 10 for an illustration of a bilinear interpolation and see, e.g., Kirkland (2010) for a derivation of the above formula). Next, we show that

$$\sup_{\mathbf{z}, \mathbf{v} \in C_1 \,:\, \|\mathbf{v} - \mathbf{z}\|_\infty \leq \delta} |\phi_{\mathbf{x}}(\mathbf{v}) - \phi_{\mathbf{x}}(\mathbf{z})| \leq 16 \cdot \lambda_{max}^2 \cdot \delta \tag{61}$$

holds for arbitrary $0 \leq \delta \leq \frac{1}{\lambda_{\max} - 1}$. For $\mathbf{v}, \mathbf{z} \in C_1$ with $\|\mathbf{v} - \mathbf{z}\|_\infty \leq \delta$ we can choose the grid points (59) for the computation of $\phi_{\mathbf{x}}(\mathbf{v})$ and $\phi_{\mathbf{x}}(\mathbf{z})$ such that there exists some $\mathbf{u} \in C_1$ satisfying

$$\mathbf{u} \in ([a_1^{(\mathbf{v})}, b_1^{(\mathbf{v})}] \times [a_2^{(\mathbf{v})}, b_2^{(\mathbf{v})}]) \cap ([a_1^{(\mathbf{z})}, b_1^{(\mathbf{z})}] \times [a_2^{(\mathbf{z})}, b_2^{(\mathbf{z})}])$$

and

$$\max\{\|\mathbf{v} - \mathbf{u}\|_\infty, \|\mathbf{z} - \mathbf{u}\|_\infty\} \leq \delta.$$

To compute $\phi_{\mathbf{x}}(\mathbf{u})$ we can therefore use the same grid points (59) as for the computation of $\phi_{\mathbf{x}}(\mathbf{v})$, such that together with inequality (60) we get

$$|\phi_{\mathbf{x}}(\mathbf{v}) - \phi_{\mathbf{x}}(\mathbf{u})| = |k_1^{(\mathbf{v})} \cdot (v_1 - u_1) + k_2^{(\mathbf{v})} \cdot (v_2 - u_2) + k_3^{(\mathbf{v})} \cdot (v_1 \cdot v_2 - u_1 \cdot u_2)|$$

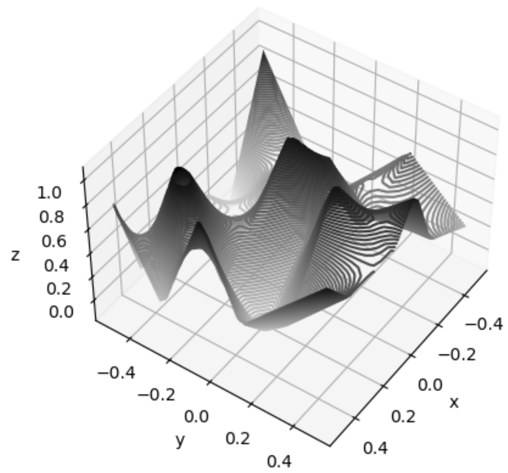

Figure 10: Illustration of a bilinear interpolation for $\lambda_{max} = 5$.

$$\begin{aligned}
&\leq |k_1^{(\mathbf{v})}| \cdot |v_1 - u_1| + |k_2^{(\mathbf{v})}| \cdot |v_2 - u_2| + |k_3^{(\mathbf{v})}| \cdot |v_1 \cdot v_2 - u_1 \cdot u_2| \\
&\leq 4 \cdot (\lambda_{max} - 1)^2 \cdot \delta + |k_3^{(\mathbf{v})}| \cdot (|v_1| \cdot |v_2 - u_2| + |u_2| \cdot |v_1 - u_1|) \\
&\leq 8 \cdot (\lambda_{max} - 1)^2 \cdot \delta,
\end{aligned}$$

and analogously

$$|\phi_{\mathbf{x}}(\mathbf{u}) - \phi_{\mathbf{x}}(\mathbf{z})| \leq 8 \cdot (\lambda_{max} - 1)^2 \cdot \delta,$$

which implies

$$|\phi_{\mathbf{x}}(\mathbf{v}) - \phi_{\mathbf{x}}(\mathbf{z})| \leq |\phi_{\mathbf{x}}(\mathbf{v}) - \phi_{\mathbf{x}}(\mathbf{u})| + |\phi_{\mathbf{x}}(\mathbf{u}) - \phi_{\mathbf{x}}(\mathbf{z})| \leq 16 \cdot \lambda_{max}^2 \cdot \delta.$$

Using inequality (61), and under the assumptions of Assumption 2 and Example 1, we then obtain

$$\begin{aligned}
&\sup_{\mathbf{z} \in C_1 \,:\, \|\mathbf{v}-\mathbf{z}\|_\infty \leq \frac{c}{\lambda}} \left| f_{0,s}\Big(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_{h_0}}\Big) - f_{0,s}\Big(\phi(\mathbf{z}) \cdot 1|_{C_{h_0}}\Big) \right| \\
&\leq \sup_{\mathbf{z} \in C_1 \,:\, \|\mathbf{v}-\mathbf{z}\|_\infty \leq \frac{c}{\lambda}} \frac{1}{h_0^2} \int_{C_{h_0}} \left| \phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_{h_0}}(\mathbf{y}) - \phi(\mathbf{z}) \cdot 1|_{C_{h_0}}(\mathbf{y}) \right| d\mathbf{y} \\
&\leq \sup_{\mathbf{z} \in C_1 \,:\, \|\mathbf{v}-\mathbf{z}\|_\infty \leq \frac{c}{\lambda}} \max_{\mathbf{y} \in C_{h_0}} \left| \phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_{h_0}}(\mathbf{y}) - \phi(\mathbf{z}) \cdot 1|_{C_{h_0}}(\mathbf{y}) \right| \\
&\leq \sup_{\mathbf{y}, \mathbf{z} \in C_1 \,:\, \|\mathbf{y}-\mathbf{z}\|_\infty \leq 2 \cdot \frac{c}{\lambda}} |\phi(\mathbf{y}) - \phi(\mathbf{z})| \\
&\leq \frac{32 \cdot c \cdot \lambda_{max}^2}{\lambda}.
\end{aligned}$$

References

[1] Anthony, M., and Bartlett, P. L. (1999). *Neural Network Learning: Theoretical Foundations.* Cambridge University Press, Cambridge.

[2] Bagirov, A. M., Clausen, C., and Kohler, M. (2009). Estimation of a Regression Function by Maxima of Minima of Linear Functions. *IEEE Transactions on Information Theory*, **55**, pp. 833–845.

[3] Bartlett, P. L., Harvey, N., Liaw, C., and Mehrabian, A. (2019). Nearly-tight VC-dimension and Pseudodimension Bounds for Piecewise Linear Neural Networks. *Journal of Machine Learning Research*, **20**, pp. 1–17.

[4] Cvetkovski, Z. (2012). *Inequalities: Theorems, Techniques and Selected Problems.* Springer, Berlin, Heidelberg.

[5] Györfi, L., Kohler, M., Krzyzak, A., and Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression.* Springer, New York.

[6] He, J. C., Li, L., and Xu, J. C. (2020). Approximation properties of deep ReLU CNNs. arXiv: 109.00190.

[7] Kirkland, E. J. (2010). *Advanced Computing in Electron Microscopy.* Springer New York, NY, 2. edition.

[8] Kohler, M., Krzyżak, A., and Walter, B. (2022). On the rate of convergence of image classifiers based on convolutional neural networks. *Annals of the Institute of Statistical Mathematics*, pp. 1–24.

[9] Kohler, M., and Langer, S. (2021). On the rate of convergence of fully connected very deep neural network regression estimates. *Annals of Statistics*, **49**, pp. 2231–2249.

[10] Walter, B. (2021). Analysis of convolutional neural network image classifiers in a hierarchical max-pooling model with additional local pooling. arXiv: 2106.05233.

[11] Zhou, D.-X. (2020). Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, **124**, pp. 319–327.